Verifiable random function from the Deuring correspondence and higher dimensional isogenies

Antonin Leroux

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

Abstract. In this paper, we introduce the family $\mathsf{DeuringVRF}_{y,z}$ of Verifiable Random Function (VRF) protocols. Based on isogenies between supersingular curves, the random function at the heart of our scheme is the one that computes the codomain of an isogeny of big prime degree from its kernel.
In $\mathsf{DeuringVRF}_{y,z}$, the evaluation is done with algorithms for the Deuring correspondence that make use of isogenies in dimension z, and the verification is based on the isogeny representation obtained from isogenies in dimension y.
The main advantage of the $\mathsf{DeuringVRF}_{y,z}$ family is its compactness, with proof sizes of a few hundred bytes, which is orders of magnitude smaller than other generic purpose post-quantum VRF constructions.
We describe four variants of our scheme with $(y,z) \in \{(2,1),(2,2),(4,1),(4,2)\}$ each offering different tradeoffs between compactness, evaluation efficiency and verification efficiency.
In the process, we introduce several new algorithms that might be of independent interest. In particular, for the variants with $z = 2$, we introduce the first algorithm to translate an ideal into the corresponding isogeny of dimension 1 using isogenies between abelian variety of dimension 2 as a tool.
The main advantage of this new algorithm compared to existing solution is the relaxation of the constraints on the prime characteristic: our new algorithm can run efficiently with "SIDH primes" that are very easy to generate unlike "SQIsign primes" that are currently required by the state of the art appoach. We believe that this algorithm opens a promising research direction to speed-up other schemes based on the Deuring correspondence such as 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 [38] and have found several practical applications in the DNSSEC protocol [25] or in blockchain consensus [7,24,13].

The most widely-used VRF constructions are based on pairings and elliptic curves such as [3] 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.

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. 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 [5], Maino, Martindale, Panny, Pope and Wesolowski [37], and Robert [41] 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 [12,1,16].

In this article, we follow the example set in [12] 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 [23,14]. 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. These algorithms will also be crucial for our new VRF construction. In this work, we also tackle the important problem of improving their efficiency.

Related Works. There exists several other proposals of quantum-resistant VRF. Lattice-based constructions were the first to appear with [26,44]. These first constructions suffered from huge proof sizes and has been subsequently improved [45,21]. Among those, the recent proposal from [21] appears to be quite practical with reasonable key and proof sizes of around 10KB. 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 [4] that introduces several construction based on hash functions.

Finally, two proposals based on isogenies have been recently introduced in [33]. The VRF protocol presented in [33] 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 $\mathsf{DeuringVRF}_{y,z}$ schemes is much closer to the recent weak VDF proposal of [16] or the new SQIsignHD [12] variant of the SQIsign signature scheme.

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

- A new family $\mathsf{DeuringVRF}_{y,z}$ of VRF protocols 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 in the random oracle model that is related to well-studied algorithmic problem of isogeny-based cryptographic. Every variant of $\mathsf{DeuringVRF}_{y,z}$ is much more compact than all existing post-quantum constructions as exhibited in Table 1.
- 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 the example parameters that we will introduce in Section 5.2 with other existing constructions. We see that all our new protocols are much more compact than all existing solutions.

	Public Key	Proof	No restrictions	Assumption	Security
	(bytes)	(bytes)			level
LB - VRF [20]	3.3K	4.9K	✗	MSIS/MLWE (Lattice)	128
X - VRF [4]	64	2.6K	✗	XMSS (Hash)	128
SL - VRF [4]	48	40K	✓	LowMC(Hash)	128
LaV [21]	8.81K	10.27K	✓	MSIS/MLWR (Lattices)	128
CAPYBARA [33]	8.3K	39K	✓	DDH (Isogenies)	128
TSUBAKI [33]	5.3K	34K	✓	sDDH (Isogenies)	128
$\mathsf{DeuringVRF}_{2,1}$	944	224	✓	$\mathrm{OMIP}_{2\mathsf{dim}}$ (Isogenies)	128
$\mathsf{DeuringVRF}_{2,2}$	304	224	✓	$\mathrm{OMIP}_{2\mathsf{dim}}$ (Isogenies)	128
$\mathsf{DeuringVRF}_{4,1}$	832	112	✓	$\mathrm{OMIP}_{4\mathsf{dim}}$ (Isogenies)	128
$\mathsf{DeuringVRF}_{4,2}$	192	112	✓	$\mathrm{OMIP}_{4\mathsf{dim}}$ (Isogenies)	128

Table 1. Comparison of the sizes of several post-quantum VRF schemes with our $\mathsf{DeuringVRF}_{y,z}$ family for $(y,z) \in \{(2,1),(2,2),(4,1),(4,2)\}$ and the parameters of Section 5.2.

,

1.1 Technical overview

The high-level idea of our $\mathsf{DeuringVRF}_{y,z}$ construction is the following: given a supersingular curve E (the public key), the $\mathsf{DeuringVRF}_{y,z}$ 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 parameter z defines which dimension is going to be used to perform this computation.

The main feature of a VRF is the verifiability of the output. The correctness of the result can be proven by embedding the isogeny $E \to E/G$ in a y-dimensional isogeny using the techniques recently introduced to attack SIDH [5,37,41].

At its heart, our construction exploits the difference 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 characteristics 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 *y-dimensional isogeny representation* (noted $y\mathsf{dim}$ hereafter) introduced recently by Robert [40]. It allows us to evaluate efficiently the isogeny with the help of dimension y 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 $\mathsf{DeuringVRF}_{y,z}$ 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 $\mathbb{F}_{p^2}$, k is the smallest exponent such that $E[N]$ is defined over $\mathbb{F}_{p^k}$. Let f be the biggest exponent such that $E[2^f]$ is defined over $\mathbb{F}_{p^2}$.

Keys. The public keys are made of:

1. a supersingular curve E,
2. a basis $\langle P, Q \rangle$ of $E[N]$,
3. some additional information $c^N_{y\mathsf{dim}}(E)$ (defined in Section 2.3).

Secret keys are constituted by:

1. an ideal I connecting a fixed maximal order $\mathcal{O}_0$ to $\mathcal{O} \cong \mathrm{End}(E)$.
2. the ideal I_P corresponding to the kernel ideal associated to $\langle P \rangle$
3. an endomorphism $\theta \in \mathrm{End}(E)$ such that $\theta(P) = Q$
4. a basis U_0, V_0 of $E_0[2^f]$.

Evaluation Mechanism. On input x, the VRF evaluation is as follows: hash x into one element $(a:b) \in \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$, compute $R_x = [a]P + [b]Q$ and compute $E_x = E/\langle R_x \rangle$. The output is then $y = H(x \mid\mid j(E_x))$. The knowledge of I_P and θ enables the efficient computation of the kernel ideal I_x of $\langle R_x \rangle$. Then, the Deuring correspondence can be used to find the curve whose endomorphism ring is isomorphic to the right order of I_x.

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 [40] obtained by embedding φ_x inside a 2^h-isogeny in dimension 2 or 4 (with $h < 2f$).

In both cases, 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 the Deuring correspondence (with the $\mathsf{IsogEval}^{z}_{\mathsf{id}}$ algorithm that we describe in Section 4.2). 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 $\mathsf{IsogEval}_{y\mathsf{dim}}$ algorithm to evaluate the isogeny from its $y\mathsf{dim}$ representation.

New algorithm to evaluate an isogeny of big prime degree from its ideal. The algorithm $\mathsf{IsogEval}_{\mathsf{id}}$ that allows us to evaluate any isogeny from the Deuring correspondence plays an important role in our new construction. The principle of this algorithm is now pretty standard in isogeny-based cryptography (see [22, Algorithm 1] or [34, Algorithm 4] for instance): first, compute with the Deuring correspondence an alternate isogeny of same domain and codomain and having smooth degree coprime to the order of the points to be evaluated with the $\mathsf{IdealToIsogeny}$ algorithm. Then, use this alternate isogeny to evaluate the first one. While this algorithm works pretty well in theory, in practice it requires a lot of available torsion (partly due to the $\mathsf{IdealToIsogeny}$ algorithm but also to the requirement that the degree of the alternate isogeny is coprime to the order of the points to be evaluated). When the order of the points to be evaluated is pretty big, the torsion requirement might simply be too big (this will be the case for our $\mathsf{DeuringVRF}_{y,1}$ constructions). In that case, it might be useful to remove the coprimality requirements. This is what we do with our new $\mathsf{IsogEvalNonCoprime}^{1}_{\mathsf{id}}$ algorithm by reusing the ideas introduced for the $\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}$ algorithm in [15] to "push" ℓ^e torsion points through an isogeny of degree $\ell^\bullet$.

New algorithms for the effective Deuring correspondence. As we explained above, the $\mathsf{IsogEval}^{z}_{\mathsf{id}}$ algorithm that constitutes the main algorithmic building block of $\mathsf{DeuringVRF}_{y,z}$ is based on a $\mathsf{IdealToIsogeny}^{z}$ algorithm to translate ideals into their corresponding isogenies (of dimension 1) with the help of isogenies in dimension z. A relatively efficient algorithm to do so for $z = 1$ was introduced in [14] and later improved in [15]. We introduce a new algorithm to do the same thing with $z = 2$.

The motivation for this new algorithm is to overcome the obstacles of $\mathsf{IdealToIsogeny}^1$ with the new possibilities offered by high-dimensional isogenies. More specifically, the bottleneck in $\mathsf{IdealToIsogeny}^1$ 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 is roughly the same idea that led to the SQIsignHD variant [12] of the SQIsign signature scheme. 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 could have interesting consequences on the efficiency of the ideal-to-isogeny algorithm. We discuss in more details the comparison between $z = 1$ and $z = 2$ at the end of this section.

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 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 $\mathrm{OMIP}_{y\mathsf{dim}}$, a variant of this problem, where the adversary has access to an oracle that computes the codomain and a $y\mathsf{dim}$ 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 $\mathsf{DeuringVRF}_{y,z}$ scheme can be found in Section 3.1. The concrete protocols include several additional steps to meet the requirements of a cryptographic VRF.

Comparison between the different variants. One might wonder why we introduce four different variants for $(y, z) \in \{(2, 1), (2, 2), (4, 1), (4, 2)\}$ instead of choosing the best one and stick with it. The main reason is that there is currently no clear answer to the question: *what is the best version of* $\mathsf{DeuringVRF}_{y,z}$? Our parametrization is made of two variables y and z, and each of them have a different role that we try to explain below.

On $y = 2$ vs. $y = 4$. The $2\mathsf{dim}$ and $4\mathsf{dim}$ representations have quite different performance profiles. We refer the reader to Section 2.3 for a detailed description of these two representation. In both cases, the algorithm $\mathsf{IsogEval}_{y\mathsf{dim}}$ (used during the verification of our protocol) essentially consists in the evaluation of a 2^h-isogeny between abelian variety of dimension y. Thus, $\mathsf{IsogEval}_{2\mathsf{dim}}$ will be clearly faster than $\mathsf{IsogEval}_{\mathit{4}\mathsf{dim}}$ as any operation in dimension 2 will be faster than the corresponding operation in dimension 4 (the complexity is in fact exponential in the dimension y). However, the *2*dim representation in itself requires to compute strictly more things than the *4*dim representation. Thus, the computation of the isogeny representation (done during the evaluation algorithm of the VRF protocol) will be faster for the *4*dim and its size is essentially half the size of the *2*dim isogeny representation.

Hence, the choice between $y = 2$ and $y = 4$ will amount to a tradeoff between efficiency of the verification for $y = 2$, and compactness and efficiency of evaluation for $y = 4$.

On $z = 1$ vs. $z = 2$. The main advantage of $z = 2$ compared to $z = 1$ is the simplification of the parameter selection (and in particular the computation of the prime characteristic) that we already mention. There are two clear positive impacts:

1. A better asymptotic scaling for $z = 2$ due to the expected sub-exponential growth of the smoothness bound of the parameter T of the $\mathsf{IdealToIsogeny}^1$ algorithm.
2. A better compactness of $\mathsf{DeuringVRF}_{y,2}$ due to the field of definition of the N-torsion points (more details on that in Section 5.1).

The only remaining open question is: *at the lower levels of security, what value of z is more efficient*? Unfortunately, it is still too early to give a definite answer to this question and it is clearly out of the scope of this paper. It was demonstrated in [14,15] that implementing the algorithm $\mathsf{IdealToIsogeny}^1$ efficiently is a daunting task and finding the best way of implementing this algorithm remains an active research question.

The state of the algorithms to compute the dimension 2 isogenies required by our $\mathsf{IdealToIsogeny}^2$ algorithm is even less stable. Some relatively efficient algorithms have been known for quite some time with the so-called Richelot correspondence [39] and new algorithms have been the focus of a lot of recent works [6,32,37,16,1,12,9].

Despite this uncertainty, the cost estimates for the computation of dimension 2 isogenies in the theta model provided in [12] allow us to be quite optimistic that a careful implementation of our new $\mathsf{IdealToIsogeny}^2$ algorithm might be competitive with the state of the art implementation of $\mathsf{IdealToIsogeny}^1$, even for the lower levels of security. This is particularly true in the context of our $\mathsf{DeuringVRF}_{y,z}$ protocol where the value of p is even more constrained than for the SQIsign signature scheme (see Section 5).

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 4, we present all the algorithms required to instantiate the protocols. In Section 5.1, 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 $a \leq \mathrm{nelg}(x)$.

2.1 Verifiable Random Function

A Verifiable Random Function (VRF) is a way to generate authenticated randomness that can be verified. It constist of the following protocols:

- $\mathsf{SetUp}(1^\lambda)$, 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.
- $\mathsf{VRFEval}(sk, x) = (v, \pi)$, 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.
- $\mathsf{Verif}(pk, \pi, x, v)$ takes the VRF public key, proof, input and output and returns 0 or 1.

In this article, we construct a VRF satisfying the following properties:

- **Provability**: The verification always returns 1 on correctly generated proof and output from a given input (see Definition 2).
- **Pseudo-randomness**: With access to an oracle computing $\mathsf{VRFEval}(sk, x)$ for $x \neq x_0$, an adversary cannot distinguish between $\mathsf{VRFEval}(sk, x_0)$ and a random value (see Definition 3).
- **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).

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 [27,30,35,42].

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 [43]. 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.

Supersingular j-invariants over $\mathbb{F}_{p^2}$ $j(E)$ (up to Galois conjugacy)	Maximal orders in $B_{p,\infty}$ $\mathcal{O} \cong \mathrm{End}(E)$ (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
$\theta \in \mathrm{End}(E_0)$	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].

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 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 [35, 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 [34,35,12] 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.

To avoid redundant computation we are going to divide the representation in two parts : one that is unique to the isogeny φ that we will write $s^{\varphi}_{\mathsf{xx}}$ (and that sometimes might be called the representation of φ), and one that is common to all isogenies of same domain E and degree N. This second part will be denoted by $c^{N}_{\mathsf{xx}}(E)$ and will be called the "common information". The distinction between $s^{\varphi}_{\mathsf{xx}}$ and $c^{N}_{\mathsf{xx}}(E)$ will be useful for an efficient instantiation (to avoid recomputing $c^{N}_{\mathsf{xx}}(E)$ for each new isogeny).

Definition 1. *An efficient isogeny representation* xx *for an isogeny* $\varphi : E \to E'$ *of degree* N *defined over* $\mathbb{F}_q$ *is in two parts:* $c^{N}_{\mathsf{xx}}(E)$ *(the same for all isogeny of degree* N *having domain* E*), and* $s^{\varphi}_{\mathsf{xx}}$*. Both have size* $O\left(\mathrm{polylog}(qN)\right)$*, and there exists the following algorithm:* $\mathsf{IsogEval}_{\mathsf{xx}}$ *that takes* $E, s^{\varphi}_{\mathsf{xx}}, c^{N}_{\mathsf{xx}}(E)$ *and a point* P *in* $E[\mathbb{F}_{q^k}]$ *in input, and computes* $\varphi(P) \in E'[\mathbb{F}_{q^k}]$ *in time* $O\big(\mathrm{polylog}(q^k N)\big)$*.*

On existing isogeny representations. There exists several isogeny representation in the literature. A non-exhaustive list of them can be found in [35, 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 $\ker\varphi$ and $c^{N}_{\mathsf{ker}}(E)$ is trivial. This representation can be quite compact $(O\left(\mathrm{polylog}(p)\right))$ 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. As for the kernel representation, the common information $c^{N}_{\mathsf{id}}(E)$ is trivial. The ideal representation matches our definition of *efficient*, however it requires to know the endomorphism ring of the domain. When $\mathrm{End}(E)$ is known, we can also efficiently sample ideal representations of uniformly random N-isogenies.

The recent attacks against the scheme SIDH [5,37,41] have introduced a new way to build an isogeny representation, as was noted by Robert in [40], 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 $y = 2$ and $y = 4$. We will call dimension-y representation (with the tag $y\mathsf{dim}$, 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 [5,37,41,40,12] 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 coprime integers a, b. The isogeny $F : A \times B' \longrightarrow B \times A'$ given in matrix notation by

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

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

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

Similarly,

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

Remark 1. This lemma was first proven in [29, Theorem 2.3]. We are going to use it for $g = 1$ and $g = 2$ to obtain the $2\mathsf{dim}$ and $4\mathsf{dim}$ representations respectively. 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.

Moreover, we are going to make good use of the trick presented in [12, Section 5.4] to cut the isogeny F in half to greatly lower the amount of torsion information required.

Indeed, when the isogeny F from Lemma 1 has degree $d_1 d_2$, then it can be factored as $F = F_2 \circ F_1$ where each F_i has degree d_i. In that case, we get that

$$\begin{aligned} \ker(F_1) &= \{(\widetilde{\varphi}(x), \psi'(x)) \mid x \in B[d_1]\} \qquad (1) \\ \ker(\widetilde{F_2}) &= \{(\varphi(x), \psi(x)) \mid x \in A[d_2]\}. \end{aligned}$$

This means that we can use the d_1-torsion and d_2-torsion instead of the $d_1 d_2$-torsion (the two situations are really different only if d_1 and d_2 are not coprime to eachother). This trick will prove to be necessary to obtain an efficient scheme in our case as we will take d_1 and d_2 as big powers of 2, which will essentially divide by 2 the torsion requirement by 2 (in bitsize).

More details on Kani's Lemma and the ways to compute isogenies in dimension $g > 1$ can be found in the appendices of [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 $\mathit{2}\mathsf{dim}$ 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^h for the smallest exponent h such that $2^h > N$. In that case, the isogeny ψ is any isogeny of degree $2^h - N$ coprime to N of domain E_1. This will define an isogeny diagram of the form

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

The isogeny ψ can be the same for every isogeny φ of the same degree and codomain. Let f be an exponent such that $2f > h$. The isogenies F_1 and F_2 are chosen to be of degree 2^{h_1} and 2^{h_2} for $h = h_1 + h_2$ and $h_i < f$ for $i = 1, 2$. Then, we can define the $\mathit{2}\mathsf{dim}$ isogeny representation as follows. Let P, Q be a basis of $E_1[2^f]$.

The common information $c^N_{\mathit{2}\mathsf{dim}}(E)$ is made of the curve E_3 and the points $P, Q, \psi(P), \psi(Q)$. Then, we define $s^{\varphi}_{\mathit{2}\mathsf{dim}}$ as E_2, E_4 and the points $\varphi(P), \varphi(Q), \psi' \circ \varphi(P), \psi' \circ \varphi(Q)$.

After the isogeny $F_1, \widetilde{F_2}$ have been computed, the full isogeny F can be recovered. Then, φ can be evaluated on any point as $\pi \circ F \circ \iota$. This is how we get the algorithm $\mathsf{IsogEval}^{\cdot}_{\mathit{2}\mathsf{dim}}$

The $\mathit{4}\mathsf{dim}$ isogeny representation. The $\mathit{4}\mathsf{dim}$ isogeny is a bit trickier to obtain. In fact, we will not provide a representation for φ directly, but for $[\alpha]\varphi$ for some integer α coprime to N. This scalar is unfortunately necessary because $2^h - \alpha^2 N$ must be represantable as the sum of two squares (which might not always be true for $\alpha = 1$). Let us now assume that we have chosen a scalar α such that $2^h - \alpha^2 N = a_1^2 + a_2^2$ for h the smallest exponent such that $2^h > \alpha^2 N$ and two integers a_1, a_2. We will later explain in Section 3.2 a bit more concretely how this integer can be chosen as it will play a role in the efficiency of the verification of our $\mathsf{DeuringVRF}_{4,z}$ scheme.

In dimension $g = 2$, Kani's lemma is applied to the isogeny $\mathrm{Diag}([\alpha]\varphi) : E_1^2 \times E_2^2$. We abuse notation by denoting this diagonal isogeny as $[\alpha]\varphi$ as well. Then, we take $A = E_1^2$ and $B = E_2^2$. The degree d is 2^h where h is the smallest exponent such that the quadratic equation $2^h - [\alpha]^2 N = a_1^2 + a_2^2$ has a solution. Then, the isogeny $\psi : E_1^2 \to E_1^2$ is defined as the matrix

$$\psi := \begin{pmatrix} [a_1]_{E_1} & [a_2]_{E_2} \\ -[a_2]_{E_2} & [a_1]_{E_1} \end{pmatrix}$$

In that case, the parameters a_1, a_2, α can be considered as known constants (since they are common to all N-isogenies). Thus, the common data $c^N_{4\mathsf{dim}}(E)$ is made of P, Q, a basis of $E_1[2^f]$ (with $2f > h$), and the data $s^{\varphi}_{4\mathsf{dim}}$ is constituted by the curve E_2 and the points $\varphi(P), \varphi(Q)$.

Similarly to the $2\mathsf{dim}$ representation, the isogeny F can be recovered from $s^{\varphi}_{4\mathsf{dim}}$ and the constants f, h, a_1, a_2, α and this allow us to instantiate a $\mathsf{IsogEval}_{4\mathsf{dim}}$ algorithm.

3 New post-quantum VRF from isogenies

In this section, we provide a generic description of our $\mathsf{DeuringVRF}_{y,z}$ protocol. The protocoles are presented 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 protocols that composes our $\mathsf{DeuringVRF}_{x,y}$ protocol for two integers $y, z \in \{(2,1),(2,2),(4,1),(4,2)\}$. We provide a common framework for all values of y, z 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 4. We also omit the parameter generation; it will be discussed later in Section 5.1.

Henceforth, let us assume that there are three distinct primes ℓ, p, N, and exponents e, f, k such that all the 2^f and ℓ^e torsion points of supersingular curves can be defined over $\mathbb{F}_{p^2}$, and the N-torsion can be defined over $\mathbb{F}_{p^k}$.

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 $\iota \in \mathcal{O}_0$ such that $\iota(P_0) = Q_0$. We write $pp = (p, N, \ell, E_0, P_0, Q_0, I_{P_0}, \iota)$.

We write $f_{\mathsf{in}} : \{0,1\}^{n_1(\lambda)} \to \mathbb{P}^1(\mathbb{Z}/N\mathbb{Z})$ an injective function, and $H_{\mathsf{out}} : \{0,1\}^* \to \{0,1\}^{n_2(\lambda)}$ a hash function, where $n_1(\lambda), n_2(\lambda)$ are functions of the security parameter λ.

Algorithm 1 $\mathsf{KeyGen}_{y,z}(pp)$

```
Input: Public parameters $pp$.
Output: A pair of $\mathsf{DeuringVRF}_{y,z}$ keys $sk, pk$.
Take some $e_0 = 2\lambda$ and generate a random $\mathcal{O}_0$-ideal $I$ of norm $\ell^{e_0}$.
Compute $\varphi_I = \mathsf{IdealToIsogeny}^{z}_{\ell^\bullet}(I)$.
Set $\theta = \varphi_I \circ \iota_0 \circ \hat{\varphi}_I$, $R = \varphi_I(P_0)$ and $S = [\ell^{e_0}]\varphi_I(Q_0)$.
Generate a random $M = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{GL}_2(\mathbb{Z}/N\mathbb{Z})$.
Set $(P, Q) \leftarrow ([a]R + [b]S, [c]R + [d]S)$.
Compute $I_P = [I \cdot (a + b\theta)]_* I_{P_0}$.
Compute the scalars $\lambda, \mu$ such that $(\lambda + \mu\theta)(P) = Q$ and set $\theta$ as $\lambda + \mu\theta$.
Compute $U, V$, a basis of $E[2^f]$, and $U_0, V_0 = \hat{\varphi}_I(P, Q)$.
Compute $c^N_{y\mathsf{dim}}(E) = \mathsf{CommonIsogenyRepresentation}_{\mathsf{y,z}}(I, U, V, U_0, V_0)$.
return $(sk, pk) = \big((I, I_P, \theta, U_0, V_0), (E, P, Q, c^N_{y\mathsf{dim}}(E))\big)$.
```

Algorithm 2 $\mathsf{VRFEval}_{y,z}(sk, x)$

```
Input: A $\mathsf{DeuringVRF}_{y,z}$ secret key, and an input $x \in \{0,1\}^{n_1(\lambda)}$.
Output: A proof $\pi$ and the evaluation $v$ of the $\mathsf{DeuringVRF}_{y,z}$ function on input $x$.
Parse $sk$ as $I, I_P, \theta, U_0, V_0$.
Compute $(r : s) = f_{\mathsf{in}}(x)$.
Compute $I_x = [\mathcal{O}(r + s\theta)]_* I_P$.
Compute $E_x$ and $s^{\varphi}_{y\mathsf{dim}} = \mathsf{IsogenyRepresentation}_{\mathsf{y,z}}(I, I_x, U_0, V_0)$.
Set $\pi$ as $j(E_x), s^{\varphi}_{y\mathsf{dim}}$.
Compute $v = H_{\mathsf{out}}(x \mid\mid j(E_x))$.
return $(\pi, v)$.
```

The algorithms $\mathsf{IsogEval}_{y\mathsf{dim}}$ are used as blackbox and we do not give much details about them in this paper. There are, however, several tasks that needs more details in the algorithms we have outlined above. We list them below. These missing algorithms will be treated in Sections 3.2 and 4.

- The algorithm $\mathsf{CommonIsogenyRepresentation}_{\mathsf{2,z}}$ to compute the common information $c^N_{2\mathsf{dim}}(E)$. We do not describe $\mathsf{CommonIsogenyRepresentation}_{\mathsf{4,z}}$ which simply takes a basis of $E[2^f]$ and returns this basis.
- The algorithm $\mathsf{IsogenyRepresentation}_{\mathsf{y,z}}$ to compute the isogeny representation and the codomain.
- The $\mathsf{IsogVerif}_{y\mathsf{dim}}$ algorithm to check that the high dimensional representation is correct.

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

```
Input: A $\mathsf{DeuringVRF}_{y,z}$ 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 $E, P, Q, c^N_{\mathsf{ydim}}(E)$.
Parse $\pi$ as $j, s^{\varphi}_{\mathsf{ydim}}$.
Compute $(r : s) = f_{\mathsf{in}}(x)$ and $R_x = [r]P + [s]Q$.
If $\mathsf{IsogVerif}_{\mathsf{ydim}}(s^{\varphi}_{\mathsf{ydim}}, c^N_{\mathsf{ydim}}(E), j) = 0$, output 0.
If $\mathsf{IsogEval}_{\mathsf{ydim}}(\pi, s^{\varphi}_{\mathsf{ydim}}, R_x) \neq O_{E_x}$ or $H_{\mathsf{out}}(x||j) \neq v$, then output 0.
return 1.
```

3.2 Security Analysis

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

Provability. The first and most basic notions a VRF 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 VRF scheme is said to be provable if, for any $x \in \{0,1\}^{n_1(\lambda)}$, $(sk, pk) \leftarrow \mathsf{KeyGen}_{y,z}(pp)$ and $(\pi, v) \leftarrow \mathsf{VRFEval}_{y,z}(sk, x)$, the following equality is satisfied:*

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

Using results proven in Section 4, we are able to show that our $\mathsf{DeuringVRF}_{y,z}$ scheme has *provability.*

Proposition 1. *The $\mathsf{DeuringVRF}_{y,z}$ scheme has provability for $(y,z) \in \{(2,1), (2,2), (4,1), (4,2)\}$.*

Proof. In $\mathsf{KeyGen}_{y,z}$, the correctness of $\mathsf{IdealToIsogeny}^{z}_{\ell^{\bullet}}$, ensures that the isogeny φ_I is the one corresponding to the ideal I. Then, with $\theta = \varphi_I \circ \iota \circ \hat{\varphi}_I$, we see that if $Q_0 = \iota(P_0)$ the definition of $S = [\ell^e]\varphi_I(Q_0)$ ensures that $\theta(R) = S$. After the change of basis is applied on R, S to get the basis P, Q, the value of θ is adjusted to ensure that $\theta(P) = Q$.

Next, we show that the ideal I_P is the kernel ideal corresponding to the point P. The point P is equal to $(a + b\theta)\varphi_I(P_0)$. By property of the Deuring correspoding, ideal multiplication corresponds to isogeny composition. Thus, the ideal $I \cdot (a + b\theta)$ corresponds to the isogeny $(a + b\theta)\varphi_I$. Then, by definition of the push-forward ideals, the ideal $[I(a+b\theta)]_* I_{P_0}$ has kernel equal to $(a+b\theta)\varphi(\ker I_{P_0})$ and this is indeed the group generated by P.

The correctness of the computation of the $\mathsf{DeuringVRF}_{y,z}$ keypair follows from the correctness of $\mathsf{CommonIsogenyRepresentation}_{\mathsf{y,z}}$.

In $\mathsf{VRFEval}_{y,z}$, we can use the same reasoning as for I_P to prove that I_x is indeed the kernel ideal corresponding to the subgroup generated by $[r]P + [s]\theta(P)$.

The correctness of the computation of $s^{\varphi}_{\mathsf{ydim}}$ follows from the correctness of $\mathsf{IsogenyRepresentation}_{\mathsf{y,z}}$.

Since we have the equality $[r]P + [s]\theta(P) = [r]P + [s]Q$, the evaluation of the point R_x in $\mathsf{Verify}_{y,z}$ will be O_{E_x}, and since $c^N_{\mathsf{ydim}}(E)$ and $s^{\varphi}_{\mathsf{ydim}}$ are honestly computed as a valid representation of an isogeny of degree N, the output of $\mathsf{IsogVerif}$ will be 1 by Proposition 3. □

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

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

where the query of $\mathsf{VRFEval}_{y,z}$ on $x^\star$ are implicitly forbidden. The pseudo-randomness advantage of $\mathcal{A}$ is defined as

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

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

The VRF is pseudo-random if

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

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

The pseudo-randomness property of our VRF is based on the hardness of Problem 1 that we introduce below. This problem is defined with respect of an isogeny representation with the tag xx. This problem uses an isogeny oracle in the fashion of the RADIO and RUGDIO introduced in [12]. 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 $E/\mathbb{F}_{p^2}$, some common information $c^N_{\mathsf{xx}}(E)$, 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 supersingular elliptic curve and $c^N_{\mathsf{xx}}(E)$ the associated common information. Given access to the N-FIXDIO$_{\mathsf{xx}}$ on input $E, c^N_{\mathsf{xx}}(E)$, 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 $\mathrm{Adv}_{\mathrm{OMIP}_{\mathsf{xx}}}(t) = \max_{\mathcal{A}} \Pr\{\mathcal{A}^{N\text{-FIXDIO}_{\mathsf{xx}}(\cdot)}(E, c^N_{\mathsf{xx}}(E))$ solves Pb. 1$\}$ for $\mathcal{A}$ ranging over all algorithms running in time t.

The hardness of Problem 1 underlies the pseudo-randomness of our VRF as we prove in Proposition 2.

Proposition 2. *In the random oracle model, the scheme* $\mathsf{DeuringVRF}_{y,z}$ *satisfies*

$$\mathrm{Adv}_{\mathrm{PR}}(t) \leq \frac{1}{2} + q\mathrm{Adv}_{\mathrm{OMIP}_{y\mathsf{dim}}}(t')$$

against any adversary $\mathcal{A} = \mathcal{A}_1, \mathcal{A}_2$ allowed less than q query to the random oracle H_{out} for some time t' polynomial in t in the random oracle model.

Proof. Let $\mathbb{G}_0$ be the pseudo-randomness game as given in Definition 3.

Let us define the game $\mathbb{G}_1$ where the random oracle H_{out} answers $\perp$ on input $x_\star, j(E_{x^\star})$ (after the value $x^\star$ has been defined). Let us call $E^\star$, the event "H_{out} is queried on $x_\star, j(E_{x^\star})$". It is clear that the two games are identical when $\neg E^\star$. Hence, $\Pr\{\mathcal{A}$ wins $\mathbb{G}_0 | \neg E^\star\} = \Pr\{\mathcal{A}$ wins $\mathbb{G}_1 | \neg E^\star\}$ and the conditional probability formula leads to

$$|\Pr\{\mathcal{A} \text{ wins } \mathbb{G}_0\} - \Pr\{\mathcal{A} \text{ wins } \mathbb{G}_1\}| \leq \Pr\{E^\star\}.$$

With the modifications defining $\mathbb{G}_1$, the output of H_{out} is always independent of v_0. In that case, given that v_0 and v_1 are distributed as uniformly random values, $\mathcal{A}$ has no way to distinguish between the two without making a forbidden evaluation query and so $\Pr\{\mathcal{A}$ wins $\mathbb{G}_1\} = 1/2$.

It now remains to bound $\Pr\{E_\star\}$. For that, we will build an adversary $\mathcal{C}$ against the OMIP from $\mathcal{A}$. This adversary proceed as follows :

1. $\mathcal{C}$ receives $E, c^N_{y\mathsf{dim}}(E)$.
2. $\mathcal{C}$ generates a random basis of $E[N]$ and transmits $pk = (E, P, Q, c^N_{y\mathsf{dim}}(E))$ to $\mathcal{A}_1$.
3. $\mathcal{C}$ answers to any query to $\mathsf{VRFEval}_{y,z}(x)$ by using the N-FIXDIO$_{y\mathsf{dim}}$ on input $R_x = [r]P + [s]Q$ where $(r : s) = f_{\mathsf{in}}(x)$ to compute E_x and the proof π. Then, $\mathcal{C}$ computes $v_x = H_{\mathsf{out}}(x, j(E_x))$ and returns $j(E_x), \pi, v_x$.
4. $\mathcal{C}$ answers to all queries to H_{out} as a random oracle would.
5. $\mathcal{C}$ receives $x^\star$ from $\mathcal{A}_1$, and aborts if $\mathsf{VRFEval}_{y,z}$ has been queried on $x^\star$.
6. $\mathcal{C}$ generates the bit b and the values v_0, v_1 as in the PR-experiment.
7. $\mathcal{C}$ transmits v_b to $\mathcal{A}_2$ and continues to simulate the evaluation oracle.
8. When $\mathcal{A}$ is done, $\mathcal{C}$ picks a random query to H_{out} of the form $x^\star, j$ and output the value j.

By definition of the N-FIXDIO$_{y\mathsf{dim}}$, $\mathcal{C}$ is able to simulate honestly all the requests to $\mathsf{VRFEval}_{y,z}$. Since the basis P, Q in $\mathsf{KeyGen}_{y,z}$ is rerandomized by a random invertible matrix M, it behaves as a random basis of $E[N]$. Moreover, until $E^\star$ happens, all the queries to H_{out} are also answered honestly. Thus, until $E^\star$ happens $\mathcal{C}$ simulates perfectly the game $\mathbb{G}_1$ for $\mathcal{A}$. This means that the probability that $E_\star$ happens during $\mathbb{G}_1$ is exactly the probability that $E^\star$ happens during the simulation by $\mathcal{C}$.

Moreover, it is clear that $\Pr\{\mathcal{C}$ solves Problem 1 $|E^\star\} \geq 1/q$. Thus, since $\Pr\{\mathcal{C}$ solves Problem 1$\} \geq \Pr\{E^\star\}\Pr\{\mathcal{C}$ solves Problem 1 $|E^\star\}$, we get

$$\Pr\{E^\star\} \leq q\mathrm{Adv}_{\mathrm{OMIP}_{y\mathsf{dim}}}(t')$$

where t' is the running time of $\mathcal{C}$ which is polynomial in t. This proves the desired result. □

Below, we analyze the complexity of the $\mathrm{OMIP}_{\mathsf{xx}}$ in the two cases relevant to our construction : $\mathsf{xx} \in \{2\mathsf{dim}, 4\mathsf{dim}\}$. We start with the $\mathrm{OMIP}_{4\mathsf{dim}}$ which is simpler to analyze.

Analysis of the $\mathrm{OMIP}_{4\mathsf{dim}}$. The most obvious way to attack the $\mathrm{OMIP}_{4\mathsf{dim}}$ is to try to compute directly any isogeny of domain E and degree N from its kernel. The best known method is the $\sqrt{\text{élu}}$ algorithm from [2]. 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 $\mathsf{Deuring}$-$\mathsf{VRF}_{y,z}$ protocols). As our protocols can run in polynomial-time, the knowledge of the endomorphism ring is obviously enough to break the $\mathrm{OMIP}_{4\mathsf{dim}}$. 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-$\mathrm{FIXDIO}_{4\mathsf{dim}}$ is provided in the $\mathrm{OMIP}_{4\mathsf{dim}}$. 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-$\mathrm{FIXDIO}_{4\mathsf{dim}}$ to obtain a new isogeny that would lead to a suitable solution to the $\mathrm{OMIP}_{4\mathsf{dim}}$. 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 [12, 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.

Analysis of the $\mathrm{OMIP}_{\mathsf{2dim}}$ In dimension 2, both $c^N_{\mathsf{2dim}}(E)$ and $s^{\varphi}_{\mathsf{2dim}}$ contain more informations than their counterpart in dimension 4. However, there does not seem to be any relevant way to exploit this additional information. Indeed, the main difference between the dimension 2 and dimension 4 is that the dimension 2 embedding is constructed from two non-trivial isogenies (and the commutative diagram they generate) instead of one in dimension 4 (since one side of the commutative diagram in $\mathsf{4dim}$ is obtained from scalar multiplications isogenies that anyone can compute efficiently). However, since the additional isogenies revealed in dimension 2 do not satisfy any specific property, there does not seem to be a reason that they would make the $\mathrm{OMIP}_{\mathsf{2dim}}$ more vulnerable. We will see later with the $\mathsf{CommonIsogenyRepresentation}_{2,z}$ algorithm, that the isogeny ψ used in our $\mathsf{2dim}$ representation is distributed uniformly among all isogenies of the same degree and domain. This should prevent any weird behaviour that might leak some information on the endomorphism ring of E.

Comparison between the two problems. Note that in general the $\mathrm{OMIP}_{\mathsf{2dim}}$ and the $\mathrm{OMIP}_{\mathsf{4dim}}$ do not appear to be equivalent. Indeed, if the degree $2^h - N$ of the extra isogeny required in $\mathsf{2dim}$ is not smooth, then computing any isogeny of degree $N - 2^h$ from its domain E is believed to be hard (for the same reasons that we assume that the OMIP is hard) and so translating $\mathsf{4dim}$ representations to $\mathsf{2dim}$ representations should be hard. However, note that if $N - 2^h$ is powersmooth, then any isogeny of degree $N - 2^h$ can be computed from E in polynomial time and so the $\mathrm{OMIP}_{\mathsf{2dim}}$ and $\mathrm{OMIP}_{\mathsf{4dim}}$ are in fact equivalent. This fact might appear surprising, but in our opinion it is only one more reason to believe that the additional information revealed in the $\mathrm{OMIP}_{\mathsf{2dim}}$ should not make it easier than the $\mathrm{OMIP}_{\mathsf{4dim}}$ for any N.

Uniqueness. A VRF 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 VRF is said to satisfy unconditional full uniqueness when no values* pk, v, v', x, π, π' *can satisfy* $\mathsf{Verify}_{y,z}(pk, \pi, x, v) = 1$ *and* $\mathsf{Verify}_{y,z}(pk, \pi', x, v') = 1$ *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 $\mathsf{IsogVerif}_{y\mathsf{dim}}$ algorithms for $y = 2, 4$.

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 $\mathsf{IsogVerif}_{\mathsf{2dim}}$ algorithm. For uniqueness, we also need to verify that the degree and kernel are correct. These two properties

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

```
Input: A curve $E_1$, a $2\mathsf{dim}$ isogeny representation $c^N_{2\mathsf{dim}}(E), s^{\varphi}_{2\mathsf{dim}}$, and a $j$-invariant $j$.
Output: A bit $b$.
Parse $c^N_{2\mathsf{dim}}(E)$ as $E_3, P_1, Q_1, P_3, Q_3$.
Parse $s^{\varphi}_{2\mathsf{dim}}$ as $E_2, E_4, P_2, Q_2, P_4, Q_4$.
if $j(E_2) \neq j$ then
    Return 0.
end if
Compute $G_1 = [2^{f-h_1}]\langle([N]P_1, P_4), ([N]Q_1, Q_4)\rangle$ and $G_2 = [2^{f-h_2}]\langle(P_2, P_3), (Q_2, Q_3)\rangle$.
if $G_i$ is not a kernel of a $2^{h_i}$-isogeny of dimension 2 for $i = 1, 2$ then
    Return 0.
end if
Compute $F_1$ of kernel $G_1$ and $\widetilde{F_2}$ of kernel $G_2$.
if the codomain of $F_1, \widetilde{F_2}$ do not agree. then
    Return 0.
end if
return 1.
```

will be verified during the check that $\varphi(R_x) = 0$ performed in $\mathsf{Verify}_{y,z}$ (see the proof of Proposition 4 for the full reasonning that this is enough).

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.

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

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

Proof. When $\mathsf{IsogVerif}_{2\mathsf{dim}}(E_1, c^N_{2\mathsf{dim}}(E), s^{\varphi}_{2\mathsf{dim}}, j) = 1$, then, since the codomain of $\widetilde{F_2}$ and F_1 agree, there exists a 2^h-isogeny $F = F_2 \circ F_1 : E_1 \times E_4 \to E_2 \times E_3$ with $2^{h_1+h_2} = 2^h$ and $j(E_2) = j$. Kani's Lemma imply that we have a valid $2\mathsf{dim}$ 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 1, and that the degree of this isogeny must be smaller than 2^h.

Conversely, when $c^N_{2\mathsf{dim}}(E), s^{\varphi}_{2\mathsf{dim}}$ are a valid representation for an isogeny φ of degree N, then by definition, we must have $P_3, Q_3 = \psi(P_1, Q_1)$, $P_2, Q_2 = \varphi(P_1, Q_1)$ and $P_4, Q_4 = \psi' \circ \varphi(P_1, Q_1)$. In the case, it can be verified that the two subgroup G_1, G_2 agree exactly with the subgroups defined in Equation 1. Thus, G_1 and G_2 are correct kernels of dimension 2 isogenies and the codomains of F_1 and $\widetilde{F_2}$ agree. Thus, the output of $\mathsf{IsogVerif}_{2\mathsf{dim}}$ is 1. □

Proposition 4. *The scheme* $\mathsf{DeuringVRF}_{2,z}$ *satisfies unconditional full uniqueness.*

Proof. Let us assume that we have a value $v = H_{\mathsf{out}}(x||j(E'))$ 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 $\varphi : E \to E'$ can be extracted from pk and π. Since $\varphi(R_x) = 0$, we know that $\langle R_x \rangle \subset \ker \varphi$. This implies that N divides the degree of φ. But since φ has degree smaller than 2^h and we assumed that h was the smallest exponent such that $2^h > N$ then $\deg \varphi$ must be N. So $\ker \varphi = \langle R_x \rangle$ and so E' must be isomorphic to $E/\langle R_x \rangle$ which proves the result. □

Verification in dimension 4. The verification in dimension 2 is quite simple because the *2*dim isogeny representation embeds the isogeny φ directly. However, the situation is a bit more complicated in dimension 4 because the concrete isogeny that is represented is $[\alpha]\varphi$ as we explained in Section 2.3. Thus, checking the degree is slightly harder. In particular, checking that $\rho_2 \circ F \circ \rho_1(R_x) = 0$ is not enough anymore as this only proves that $\rho_2 \circ F \circ \rho_1$ can be factored by φ. To ensure uniqueness, the verifier must be able to ensure that $\rho_2 \circ F \circ \rho_1 = [\alpha]\varphi$. This is why $\mathsf{IsogVerif}_{\mathit{4}\mathsf{dim}}$ will include another step compared to $\mathsf{IsogVerif}_{\mathit{2}\mathsf{dim}}$: check that $E[\alpha] \subset \ker \rho_2 \circ F \circ \rho_1$. This is why the exact choice of α is important. To enable an efficient verification, we need that the α-torsion points are defined over a small field extension. That way, it can be easily checked that $\rho_2 \circ F \circ \rho_1(E[\alpha]) = 0$ with $\mathsf{IsogEval}_{\mathit{4}\mathsf{dim}}$. We will specify in Section 5.2 an example of parameters for which the α-torsion is defined over $\mathbb{F}_{p^2}$.

Proposition 5. *If* $\mathsf{IsogVerif}_{\mathit{4}\mathsf{dim}}(E_1, c^N_{\mathit{4}\mathsf{dim}}(E), s^{\varphi}_{\mathit{4}\mathsf{dim}}, j) = 1$*, then* $c^N_{\mathit{4}\mathsf{dim}}(E), s^{\varphi}_{\mathit{4}\mathsf{dim}}$ *constitute a valid 4*dim *isogeny representation for an isogeny* $\varphi : E_1 \to E_2$ *of degree smaller than* $2^h/\alpha^2$ *where* $j(E_2) = j$.

Conversely, if $c^N_{\mathit{4}\mathsf{dim}}(E), s^{\varphi}_{\mathit{4}\mathsf{dim}}$ *is a valid 4*dim *isogeny representation for an isogeny* $\varphi : E_1 \to E_2$ *of degree* N *, then* $\mathsf{IsogVerif}_{\mathit{4}\mathsf{dim}}(E, c^N_{\mathit{4}\mathsf{dim}}(E), s^{\varphi}_{\mathit{4}\mathsf{dim}}, j(E_2)) = 1$.

Proof. When $\mathsf{IsogVerif}_{\mathit{4}\mathsf{dim}}(E_1, c^N_{\mathit{4}\mathsf{dim}}(E), s^{\varphi}_{\mathit{4}\mathsf{dim}}, j) = 1$, then, since the codomain of $\widetilde{F_2}$ and F_1 agree, there exists a 2^h-isogeny $F = F_2 \circ F_1 : E_1^2 \to E_2^2$ with $2^{h_1+h_2} = 2^h$ and $j(E_2) = j$. Kani's Lemma, imply that we have a valid *4*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 1, and that the degree of this isogeny must be smaller than 2^h. Moreover, since $\rho_2 \circ F \circ \rho_1(E_1[\alpha]) = 0$, then $\rho_2 \circ F \circ \rho_1$ can be factored by $[\alpha]$ and so the degree of the isogeny from E_1 to E_2 is smaller than $2^h/\alpha^2$.

Conversely, when $c^N_{\mathit{4}\mathsf{dim}}(E), s^{\varphi}_{\mathit{4}\mathsf{dim}}$ is a valid representation for an isogeny φ of degree N, then by definition, we must have $P_2, Q_2 = \varphi(P_1, Q_1)$. In that case, it can be verified that the two subgroup G_1, G_2 agree exactly with the subgroups defined in Equation 1. Thus, G_1 and G_2 are correct kernels of dimension 4 2^{h_1}-isogenies and 2^{h_2}-isogenies and the codomains of F_1 and $\widetilde{F_2}$ agree. Finally, since the points have been multiplied by α in the definition of G_1,G_2. The isogeny $\rho_2 \circ F \circ \rho_1$ can be factored by $[\alpha]$ and so the evaluation on the points of order α will be 0. Thus, the output of $\mathsf{IsogVerif}_{\mathit{4}\mathsf{dim}}$ is 1. □

Algorithm 5 $\mathsf{IsogVerif}_{4\,\mathsf{dim}}(E_1, c^N_{4\,\mathsf{dim}}(E), s^{\varphi}_{4\,\mathsf{dim}}, j)$

```
Input: A 4dim isogeny representation $c^N_{4\,\mathsf{dim}}(E), s^{\varphi}_{4\,\mathsf{dim}}$, a $j$-invariant $j$.
Output: A bit $b$.
Parse $c^N_{4\,\mathsf{dim}}(E)$ as $P_1, Q_1$.
Parse $s^{\varphi}_{4\,\mathsf{dim}}$ as $E_2, P_2, Q_2$.
if $j(E_2) \neq j$ then
    Return 0.
end if
Compute
    $$\begin{aligned} G_1 = [2^{f-h_1}]\langle &(([a_1]P_1, [a_2]P_1, [\alpha]P_2, 0), ([a_1]Q_1, [a_2]Q_1, [\alpha]Q_2, 0), \\ &(-[a_2]P_1, [a_1]P_1, 0, [\alpha]P_2), (-[a_2]Q_1, [a_1]Q_1, 0, [\alpha]Q_2, )\rangle \\ G_2 = [2^{f-h_2}]\langle &(([a_1]P_1, -[a_2]P_1, -[\alpha]P_2, 0), ([a_1]Q_1, -[a_2]Q_1, -[\alpha]Q_2, 0), \\ &([a_2]P_1, [a_1]P_1, 0, -[\alpha]P_2), (-[a_2]Q_1, [a_1]Q_1, 0, -[\alpha]Q_2, )\rangle. \end{aligned}$$
if $G_i$ is not a kernel of a $2^{h_i}$-isogeny of dimension 4 for $i = 1, 2$ then
    Return 0.
end if
Compute $F_1$ of kernel $G_1$ and $\widetilde{F}_2$ of kernel $G_2$.
if the codomain of $F_1, \widetilde{F}_2$ do not agree. then
    Return 0.
end if
Compute a basis $P_\alpha, Q_\alpha$ of $E_1[\alpha]$.
Set $\rho_1 : E_1 \to E_1^2 \times E_2^2$ any embedding that is the identity on the first coordinate,
    and $\rho_2 : E_1^2 \times E_2^2 \to E_2$ as the canonical projection on the third coordinate.
if $\rho_2 \circ F_2 \circ F_1 \circ \rho_1(P_\alpha) \neq 0$ or $\rho_2 \circ F_2 \circ F_1 \circ \rho_1(Q_\alpha) \neq 0$ then
    Return 0.
end if
return 1.
```

Proposition 6. *The scheme* $\mathsf{DeuringVRF}_{4,z}$ *satisfies unconditional full uniqueness.*

Proof. Let us assume that we have a value $v = H_{\mathsf{out}}(x||j(E'))$ 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 5, we know that a valid isogeny representation for an isogeny $\varphi : E \to E'$ can be extracted from pk and π. Since $\varphi(R_x) = 0$, we know that $\langle R_x \rangle \subset \ker \varphi$. This implies that N divides the degree of φ. But since φ has degree smaller than $2^h/\alpha^2$ and we assumed that h was the smallest exponent such that $2^h > N\alpha^2$ then $\deg \varphi$ must be N. So $\ker \varphi = \langle R_x \rangle$ and so E' must be isomorphic to $E/\langle R_x \rangle$ which proves the result. □

4 The algorithmic instantiation of $\mathsf{DeuringVRF}_{y,z}$.

In this section, we fill the blanks left in Section 3.1, and dive into the more complicated sub-algorithms of our VRF construction. Our goal is to be able to instantiate the $\mathsf{DeuringVRF}_{y,z}$ family for $(y, z) \in \{(2, 1), (2, 2), (4, 1), (4, 2)\}$. Thus, this section will introduce the three following algorithms: $\mathsf{IsogenyRepresentation}_{4,z}$, $\mathsf{CommonIsogenyRepresentation}_{2,z}$ and $\mathsf{IsogenyRepresentation}_{2,z}$ for $z \in \{1, 2\}$. The description of those algorithms can be found in Section 4.3, but several building blocks are required before that.

The most crucial sub-algorithm is an algorithm to evaluate an isogeny from its ideal representation. This will be the focus of Section 4.2. This evaluation algorithm itself is built on top of another algorithm, which is our most basic build block : $\mathsf{IdealToIsogeny}^z$, an algorithm to realize the effective the Deuring correspondence by translating an ideal of smooth norm given in input to its corresponding isogeny. Here, the label z plays the same role as in $\mathsf{DeuringVRF}_{y,z}$, it indicates that the algorithm will make use of isogenies in dimension z. The algorithm $\mathsf{IdealToIsogeny}^z$ is introduced in Section 4.1 with all the necessary building blocks to instantiate it with $z = 1, 2$.

We will provide in Section 5.3 a complete discussion on the various parameter constraints and choices to instantiate our $\mathsf{DeuringVRF}_{y,z}$ family. In the rest of this section, we omit most efficiency considerations even when they underlie some of the design choices and focus on obtaining correct algorithms.

4.1 Algorithms for the effective Deuring correspondence

The goal of this section is to instantiate the algorithm $\mathsf{IdealToIsogeny}^z_{\ell^\bullet}$ to translate an ideal of norm a power of ℓ into their corresponding isogeny for some small prime ℓ and $z = 1, 2$. The isogeny to be computed is always of dimension 1 (even when $z > 1$), but isogenies of dimension z will be used during the execution of $\mathsf{IdealToIsogeny}^z_{\ell^\bullet}$.

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 $\mathsf{IdealToIsogeny}^{z}_{\ell^\bullet}$ is exactly ℓ^{ne} for some integer n. The generic case can be derived trivially from there.

In fact, our algorithm $\mathsf{IdealToIsogeny}^{z}_{\ell^\bullet}$ is not new. It has been introduced as $\mathsf{IdealToIsogenyEichler}_{\ell^\bullet}$ for $z = 1$ in [15, Algorithm 5] in the context of the SQIsign signature scheme. When the input has norm ℓ^{ne}, the algorithm $\mathsf{IdealToIsogeny}$ $\mathsf{Eichler}_{\ell^\bullet}$ consists in n sequential executions of a sub-algorithm $\mathsf{IdealToIsogeny}$ $\mathsf{Eichler}_{\ell^e}$ ([15, Algorithm 4]) that performs the translation for inputs of norm ℓ^e exactly.

For $z = 2$, we will keep the same structure as $z = 1$ for the high-level algorithm $\mathsf{IdealToIsogeny}^{2}_{\ell^\bullet}$. Thus, we give a common description (with z as a non-specified parameter) at the end of this section as Algorithm 8.

We introduce an algorithm $\mathsf{IdealToIsogeny}^{2}_{\ell^e}$ to replace $\mathsf{IdealToIsogenyEichler}_{\ell^e}$ when $z = 2$. A detailed description of $\mathsf{IdealToIsogenyEichler}_{\ell^e}$ (that we relabel as $\mathsf{IdealToIsogeny}^{1}_{\ell^e}$) can be found in Appendix A.

We start with a brief summary of the ideas underlying $\mathsf{IdealToIsogeny}^{1}_{\ell^e}$ 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 $\mathsf{IdealToIsogeny}^{1}_{\ell^\bullet}$ 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 $\mathsf{IdealToIsogeny}^{1}_{\ell^e}$ performs a "refresh" operation, necessary to all its subsequent executions inside $\mathsf{IdealToIsogeny}^{1}_{\ell^\bullet}$. In $\mathsf{IdealToIsogeny}^{1}_{\ell^e}$, 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 the faster the computation will be). Moreover, we will see in Section 5.3 that, in the context of our $\mathsf{DeuringVRF}_{y,1}$ protocol, there are some additional constraints to take into account that complicate even more the search for a smooth T.

This limitation is the main motivation to introduce a variant with $z = 2$. The goal of this algorithm is to overcome the obstacle of the case $z = 1$ by exploiting the $\mathit{2}\mathsf{dim}$ isogeny representation.

Translating ideal to isogenies with isogenies in dimension 2, an overview. Our goal with the case $z = 2$ 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 [12]: by embedding θ in an 2^h-isogeny of higher dimension (for some exponent h), 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 RandomGoodEndomorphism to find suitable endomorphisms.

Remark 2. Unlike SQIsignHD, where the dimension 4 is required, we will see that the dimension 2 is enough for our purpose because we can use endomorphisms to apply Kani's Lemma. This idea could not be applied to obtain a dimension 2 equivalent of SQIsignHD protocol because that would imply to reveal endomorphisms of the public key curve which would destroy the security of the scheme (since the endomorphism ring of the public key must remain secret). Also, note that it should probably be possible to devise $\mathsf{IdealToIsogeny}^z_{\ell^e}$ algorithms for $z = 4$ (or even $z = 8$) following the same ideas. However, there do not really seem to be a gain that would compensate for the efficiency loss induced by the cost of the higher dimension computations.

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 *2*dim representation for any isogeny φ of degree a requires a second isogeny β of degree b such that $2^h = a + b$. In our case, the isogeny we want to represent is an endomorphism θ. Following an idea introduced in [8], we propose to choose β as an endomorphism of E as well. With the method described in [8], it is possible to find efficiently two endomorphisms θ, β in the same quadratic order satisfying the norm equation $2^h = n(\theta) + n(\beta)$. This can be done in the following way: let us take $\omega \in \mathrm{End}(E)$ an endomorphism of E of trace 0 and norm n. Then, for any x, c we have $n(x + c\omega) = x^2 + c^2 n$. Thus, $n(x_1 + c_1\omega) + n(x_2 + c_2\omega) = x_1^2 + x_2^2 + (c_1^2 + c_2^2)n$ and it suffices to find x_1, x_2, c_1, c_2 such that $x_1^2 + x_2^2 + (c_1^2 + c_2^2)n = 2^h$ to obtain an endomorphism $\theta = x_1 + c_1\omega$ that we will be able to represent efficiently with the *2*dim representation. This equation can be solved quite easily using Cornacchia's algorithm when 2^h is big enough compared to n.

As in SpecialEichlerNorm, the endomorphism θ computed by 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 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 also provide some more insights on these potential failures later in this section.

For RandomGoodEndomorphism, we define as $\mathcal{O}^\perp$ the sub-lattice of $\mathcal{O}$ made of elements of trace 0.

RandomGoodEndomorphism makes use of the Cornacchia algorithm from [10] that takes two integers d, n and finds (when possible) x, y such that $x^2 + dy^2 = n$.

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

```
Input: A maximal order $\mathcal{O}$, an $\mathcal{O}$-ideal $K$ of norm $\ell$, and an integer $h$.
Output: $\theta, \beta \in \mathcal{O}$ with $\beta \notin \mathbb{Z} + K$ and $2^h = n(\theta) + n(\beta)$.
Compute $\alpha_1, \alpha_2, \alpha_3$, the three elements of $\mathcal{O}^\perp$ realizing the three successive minimas
  of $\mathcal{O}^\perp$ and set $M$ as the set of $\alpha_i$ with norm smaller than $2^h$ that are not contained
  in $\mathbb{Z} + K$.
if $M = \emptyset$ then
    Return $\perp$
end if
Set found as false
for $\omega \in M$ do
    Set $C$ as the set of coefficients $c_1, c_2$ such that $2^h - (c_1^2 + c_2^2)n(\omega) > 0$.
    for $c_1, c_2 \in C$ do
        Set $n = 2^h - (c_1^2 + c_2^2)n(\omega)$.
        if If Cornacchia$(n, 1) \neq \perp$ then
            $x_1, x_2 =$ Cornacchia$(n, 1)$.
            Set $\theta_i = x_i + c_i\omega$ for $i = 1, 2$.
            Set found = true
        end if
    end for
end for
Set $\theta = \theta_1$, $\beta = \theta_2$.
return Return $\theta, \beta$
```

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 $8p^{2/3}$ and trace 0. We have also included a result by Boneh and Love regarding the number of curves having an endomorphism of norm smaller than some bound [36, Proposition A.3].

Lemma 2. *For any supersingular curve E, there exists an endomorphism $\omega : E \to E$ of degree smaller than $8p^{2/3}$ and $\mathrm{tr}(\omega) = 0$.*

Moreover, given any $B < 8p^{2/3}$, the number of curves having a non-trivial endomorphism of trace 0 and 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 $2p^{2/3}$. Either $\mathrm{tr}(\theta) = 0$ or we can assume w.l.o.g that $\mathrm{tr}(\theta) = 1$. In that case $\mathrm{tr}(2\theta - 1) = 0$ and $n(2\theta - 1) < 8p^{2/3}$.

The second result regaring the number of curves having a non-trivial endomorphism smaller than B was proven as [36, Proposition A.3]. □

Proposition 7. *(Heuristic) For any $\kappa > 0$, there exists $\eta = \Theta(\log\log(p) + \kappa)$ such that, if $h > 2/3\log(p) + \eta$, then* RandomGoodEndomorphism *will succeed with probability bigger than $1 - 2^{-\kappa}$ on input $\mathcal{O}, K, h$ 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 $M = \emptyset$ or no suitable pair c_1, c_2 was found. First, note that if $n(\alpha_3) < 2^h$, then M cannot be empty. Indeed, 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 $\mathcal{O}^\perp/\ell\mathcal{O}^\perp \cong M_2(\mathbb{Z}/\ell\mathbb{Z})/\mathbb{Z}$ (since $\ell \neq 2$).

Let us take some value $A < 2^h$ and assume that the input $\mathcal{O}$ is such that $n(\alpha_3) < A$. By choice of A, M is not empty. Then, for a given $\omega \in M$ there is $\#C \geq \lambda_0 2^h/n(\omega)$ possible pairs c_1, c_2 where λ_0 is some constant. Under the heuristic that $N = 2^h - (c_1^2 + c_2^2)n(\omega)$ behave as a random integer of the same size, the success probability of Cornacchia$(N, 1)$ is bigger than $\lambda_1/\log(p)$ for some constant λ_1. Thus, for any $\mathcal{O}$ such that $n(\alpha_3) < A$ we can upper-bound the failure probability of RandomGoodEndomorphism on input $\mathcal{O}$ by $(1 - \lambda_1/\log(p))^{\lambda_0 2^h/A}$.

If we write $p(A)$, the probability that a random $\mathcal{O}$ in $B_{p,\infty}$ is such that $n(\alpha_3) > A$, 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

$$p(A) + (1 - \lambda_1/\log(p))^{\lambda_0 2^h/A}$$

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, and so we obtain that $p(A) \leq \mu_3\sqrt{p}/A^{3/4}$ for some constant μ_3.

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

$$\mu_3 \frac{\sqrt{p}}{A^{3/4}} + \left(1 - \frac{\lambda_1}{\log(p)}\right)^{\lambda_0 2^h/A} \tag{3}$$

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

Potential failures of RandomGoodEndomorphism*.* We can extract from the proof of Proposition 7 the cases where RandomGoodEndomorphism will potentially fail: when the smallest non-trivial endomorphism of $\mathcal{O}$ is smaller than usual (so the third successive minima is bigger than, or very close to, 2^{2f}) and is contained in

$\mathbb{Z}+K$. 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 7 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 $f > 1/2 \log(p)$ the proportion of failing maximal order will decrease very quickly to 0. In all of our applications of 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 $1/3 \log(p)$. In particular, in our application to $\mathsf{DeuringVRF}_{y,2}$, we need to take $f \approx 1/2 \log(p)$ and so we expect those failures to happen with negligible probability.

Remark 3. Note that θ and β commute because they are in the same quadratic order. Thus, we do not need to bother with any coprimality conditions on the norm of θ and β in RandomGoodEndomorphism because Lemma 1 applies as soon as there is a commutative diagram. However, the formula to compute $\ker F$ and $\ker \widetilde{F}$ needs to be adjusted (this formula is only correct if $\ker \beta \cap \ker \theta = \{0\}$, which is always true for coprime degrees but might not be otherwise). We explain in Appendix B that this should only happen when θ and β have a very specific form. We expect these bad cases to occur with very small probability which is why we do not treat them directly in the description of $\mathsf{IdealToIsogeny}^2_{\ell^e}$ below. In Appendix B, we explain how to solve the issue when it happens. Our proposed solution requires a bit more available torsion than is necessary for the method detailed in $\mathsf{IdealToIsogeny}^2_{\ell^e}$ (we need $f = h$ to be able to make it work in every case whereas $2f \geq h$ is usually enough).

The full ideal-to-isogeny subroutine in dimension 2. As we explained above, we obtain $\mathsf{IdealToIsogeny}^2$ by adapting $\mathsf{IdealToIsogeny}^1$ to use RandomGoodEndomorphism instead of SpecialEichlerNorm and compute θ from a 2dim 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 $\mathsf{IsogEval}_{2\mathsf{dim}}$ to evaluate isogenies from their 2dim-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 $\theta_0 \in \mathrm{End}(E_0)$.

Proposition 8. *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 $\mathsf{IsogEval}_{2\mathsf{dim}}$, we need to show that the isogeny representation $c^{n(\theta)}_{2\mathsf{dim}}(E)$, $s^{\theta}_{2\mathsf{dim}}$ 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,

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

```
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 $s^{\theta}_{\mathsf{2dim}} = E, E, U, V, Y, Z$, and $c^{n(\theta)}_{\mathsf{2dim}}(E) = E, R, S, W, X$.
Compute $Q = \mathsf{IsogEval}_{\mathsf{2dim}}(s^{\theta}_{\mathsf{2dim}}, c^{n(\theta)}_{\mathsf{2dim}}(E), P)$
Compute $\varphi_I$ of kernel $\langle [C]P + [D]Q \rangle$.
return $\varphi_I$.
```

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 $\mathsf{2dim}$ 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 $\mathsf{IdealToIsogeny}^{z}$ 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.

4.2 Evaluating isogenies from the ideal representation

In this section, we introduce two algorithms to evaluate isogenies from their ideal representation. The first one is called $\mathsf{IsogEval}^{z}_{\mathsf{id}}$ (see Algorithm 9 below) and works only when order of the point P is coprime to ℓ the small prime used in $\mathsf{IdealToIsogeny}^{z}_{\ell^\bullet}$. This algorithm is greatly inspired from [34, Algorithm 2] (or equivalently [22, Algorithm 1]). In our application of$\mathsf{IsogEval}^{z}_{\mathsf{id}}$, we will need to evaluate points of order 2^f. This will not be problematic when $z = 2$ because we take $\ell \neq 2$. However, with our choices of parameters when $z = 1$, only the translation of isogenies of degree $2^\bullet$ will be efficient (more on that in Section 5.1). Thus, $\mathsf{IsogEval}^{1}_{\mathsf{id}}$ will not be applicable efficiently and this is why

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

```
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 $\varphi_{I_i} = \mathsf{IdealToIsogeny}^{z}_{\ell^e}(\mathcal{O}_i, I_i, J_i, \varphi_I \circ \varphi_J, P_i)$.
    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 $\varphi_I$.
```

we need to introduce a second algorithm called $\mathsf{IsogEvalNonCoprime}^{1}_{\mathsf{id}}$ to handle the case where the order of the points is not coprime to ℓ.

When the order is coprime to ℓ. The high level idea is the following: find an ideal of norm $\ell^\bullet$ equivalent to the ideal in input with the KLPT algorithm from [31], compute the corresponding isogeny with $\mathsf{IdealToIsogeny}^{z}_{\ell^\bullet}$ and use this isogeny to compute the final output. As we said above, this algorithm is now pretty standard. We refer the reader to [35, Section 4.2.4] for a more detailed description, correctness proof and complexity analysis, including the presentation of all the necessary building blocks.

Algorithm 9 $\mathsf{IsogEval}^{z}_{\mathsf{id}}(I, E, P)$

```
Input: $I$ an ideal of $B_{p,\infty}$, $E$ an elliptic curve with $\mathrm{End}(E) \cong \mathcal{O}_L(I)$ and $P \in E[2^f]$ where $\ell$ is coprime to $N = n(I)$.
Output: $\varphi_I(P)$.
Compute $J = \mathsf{KLPT}_{\ell_2^\bullet}(I)$ and set $K = I \cdot \overline{J}$. Set $\alpha \in \mathrm{End}(E)$ as the endomorphism $\varphi_K$.
Compute $\alpha(P)$.
Compute $\varphi_J = \mathsf{IdealToIsogeny}^{z}_{\ell_2^\bullet}(J)$ and compute $Q = \varphi_J(\alpha(P))$.
Compute $\mu = n(J)^{-1} \mod 2^f$.
return $[\mu]Q$.
```

When the order is not coprime to ℓ. When we want to use $\mathsf{IsogEval}^{z}_{\mathsf{id}}$ with $\ell = 2$, the main problem we encounter is that $n(J)$ is not invertible $\mod 2^f$. Fortunately, there is a way to circumvent this issue by applying the following idea that underlies $\mathsf{IdealToIsogenyEichler}_{\ell^\mathsf{e}}$: 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 $O(\mathrm{poly}(\log(p) + f))$ 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 $O(\mathrm{poly}(\log(p) + f))$. □

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 [35, section 4.2.1]). The integer T is an implicit parameter of Algorithm 10.

Proposition 9. *(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 [35] 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

```
Algorithm 10 $\mathsf{IsogEvalNonCoprime}^1_{\mathsf{id}}(I, P)$
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 $[a]S_1 + [b]S_2$.
```

same that are performed in $\mathsf{IdealToIsogeny}^1_{\ell^\bullet}$ (that we describe in Appendix A). We refer the reader to the proof of [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 $[a]S_1 + [b]S_2 = \pm\varphi_I(P)$.

□

4.3 Computing high dimensional isogeny representation from the ideal representation.

The goal of this section is to introduce the algorithms $\mathsf{CommonIsogenyRepresentation}_{\mathsf{y,z}}$ and $\mathsf{IsogenyRepresentation}_{\mathsf{y,z}}$ that are respectively building blocks of the $\mathsf{KeyGen}_{y,z}$ and $\mathsf{VRFEval}_{y,z}$ algorithms of our $\mathsf{DeuringVRF}_{y,z}$ scheme. Given the pairs y, z that we target, we only need this function for $y = 2, 4$. These algorithms can be derived in a straightforward manner from the definition of the *2*dim and *4*dim representation. They mainly consists in applications of IsogEval$_{\mathsf{z}}^{id}$ and IsogEvalNonCoprime$_{1}^{id}$. We start with the simpler case $y = 4$.

*Constructing the 4*dim *isogeny representation.* The CommonIsogenyRepresentation$_{\mathsf{4,z}}$ algorithm is trivial, it simply outputs the points P, Q. We do not give a more detailed description than that. The IsogenyRepresentation$_{\mathsf{4,z}}$ algorithm simply consists in two executions of the isogeny evaluation algorithm. In our case, we use IsogEval$_{\mathsf{id}}^{z}$ when $z \neq 1$ and IsogEvalNonCoprime$_{\mathsf{id}}^{1}$ when $z = 1$.

Algorithm 11 IsogenyRepresentation$_{\mathsf{4,z}}(I, J, \hat{\varphi}_I(P), \hat{\varphi}_I(Q))$

```
Input: $I$ an $\mathcal{O}_0$-ideal, $J$ an $\mathcal{O}_R(I)$-ideal of norm $N$, the image $\hat{\varphi}_I(P, Q)$ of a basis $P, Q$ of $E[2^f]$.
Output: $s^{\varphi}_{4\mathsf{dim}}$ for $\varphi$ the isogeny corresponding to $J$
Compute the codomain $E_2$ of $J$.
if $z \neq 1$ then
    $R = \mathsf{IsogEval}^{z}_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(P))$ and $S = \mathsf{IsogEval}^{z}_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(Q))$.
else
    $R = \mathsf{IsogEvalNonCoprime}^{z}_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(P))$ and $S = \mathsf{IsogEvalNonCoprime}^{z}_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(Q))$.
end if
return $E_2, R, S$.
```

*Constructing the 2*dim *isogeny representation.* Similarly to the case $y = 4$, the two algorithms CommonIsogenyRepresentation$_{\mathsf{2,z}}$ and IsogenyRepresentation$_{\mathsf{2,z}}$ are mainly constitued of isogeny evaluations.

Algorithm 12 CommonIsogenyRepresentation$_{\mathsf{2,z}}(I, P, Q, \hat{\varphi}_I(P), \hat{\varphi}_I(Q))$

```
Input: $I$ an $\mathcal{O}_0$-ideal, the points $P, Q, \hat{\varphi}_I(P), \hat{\varphi}(Q)$ where $P, Q$ is a basis of $E[2^f]$ for the codomain $E$ of $\varphi_I : E_0 \to E$.
Output: $c^{N}_{2\mathsf{dim}}(E)$
Compute deterministically a random $\mathcal{O}_R(I)$-ideal $J$ of norm $2^h - N$.
if z =4 then
    $R = \mathsf{IsogEval}^{z}_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(P))$ and $S = \mathsf{IsogEval}^{z}_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(Q))$.
else if z=1 then
    $R = \mathsf{IsogEvalNonCoprime}^{z}_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(P))$ and $S = \mathsf{IsogEvalNonCoprime}^{z}_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(Q))$.
end if
Set $E_3$ the codomain of the isogeny corresponding to $I \cdot J$.
return $E_3, P, Q, R, S$.
```

Algorithm 13 $\mathsf{IsogenyRepresentation}_{2,z}(I, J, \hat{\varphi}_I(P), \hat{\varphi}_I(Q))$

```
Input: $I$ an $\mathcal{O}_0$-ideal, $J$ an $\mathcal{O}_R(I)$-ideal of norm $N$, the image $\hat{\varphi}_I(P, Q)$ of a basis $P, Q$ of $E[2^f]$.
Output: $s^{\varphi}_{2\mathsf{dim}}$
if z =4 then
    $R_2 = \mathsf{IsogEval}^z_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(P))$ and $S_2 = \mathsf{IsogEval}^z_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(Q))$.
else if z=1 then
    $R_2 = \mathsf{IsogEvalNonCoprime}^z_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(P))$ and $S_2 = \mathsf{IsogEvalNonCoprime}^z_{\mathsf{id}}(I \cdot J, \hat{\varphi}_I(Q))$.
end if
Set $E_2$ the codomain of the isogeny corresponding to $I \cdot J$.
Compute deterministically a random $\mathcal{O}_R(I)$-ideal $J'$ of norm $2^h - N$.
Set $K = I \cdot J \cdot [J]_* J'$.
if z =4 then
    $R_4 = \mathsf{IsogEval}^z_{\mathsf{id}}(K, \hat{\varphi}_I(P))$ and $S_4 = \mathsf{IsogEval}^z_{\mathsf{id}}(K, \hat{\varphi}_I(Q))$.
else if z=1 then
    $R_4 = \mathsf{IsogEvalNonCoprime}^z_{\mathsf{id}}(K, \hat{\varphi}_I(P))$ and $S_4 = \mathsf{IsogEvalNonCoprime}^z_{\mathsf{id}}(K, \hat{\varphi}_I(Q))$.
end if
Set $E_4$ the codomain of the isogeny corresponding to $K$.
return $E_2, R_2, S_2, E_4, R_4, S_4$.
```

5 Parameters and Performances

In this section we discuss the choice of parameters to instantiate our $\mathsf{Deuring}$-$\mathsf{VRF}_{y,z}$ family as efficiently as possible at a given level of security λ. We propose concrete sets of parameters for $\lambda = 128$ presumably corresponding to the NIST-I level of security. Then, we assess the features of the different variants of our VRF family.

We remind that we target pairs $y, z \in \{(2,1), (2,2), (4,1), (4,2)\}$. We will see below that the value of $z \in \{1,2\}$ mostly determines our choice of prime p.

5.1 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 diving into how to choose 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 $O(\sqrt{p})$. Thus, we need to take $\log(p) = 2\lambda$. Similarly, the best known algorithm to compute N-isogenies has complexity $O(\sqrt{N})$. Thus, we need to target $\log(N) = 2\lambda$.

A summary of the constraints for efficiency. We need to take a prime N such that $p^k = 1 \mod N$ for the smallest possible k to ensure that the N-torsion points are defined over $\mathbb{F}_{p^k}$. We need that the 2^f torsion is defined over $\mathbb{F}_{p^2}$ for a value of f such that $2^{2f} > N$. We also need to have other points of smooth order defined over $\mathbb{F}_{p^2}$, but the exact requirement depends on the value of z.

When $z = 1$. In that case, we need to have the T-torsion defined over $\mathbb{F}_{p^2}$ where T is a smooth and odd number bigger than $p^{5/4}$ (this is a requirement of the $\mathsf{IdealToIsogeny}^1_{2^\bullet}$) algorithm). In that case, the prime p that we need is a so-called "SQIsign prime" as in [15]. The main difference between our setting and SQIsign's is that we have the additional constraints that the power of two exponent f must be of size $\approx \log(p)/2$ and that there must be a big prime divisor of $p^k - 1$ for a small value of k. The second is not too hard to satisfy for any given p if we allow the value of k to grow slightly (even though it might be hard to compute the value of N due to the cost of factorization). However, the first constraint on the size of f is quite restrictive and will be the main obstacle for an efficient instantiation of our scheme. Indeed, with such a big value of f, the smoothness bound of T is necessarily quite big.

We will give later an example of primes satisyfing almost all of our constraints. No extensive computation was involved to find this prime that we just use as a proof of concept. With more work, more efficient primes can certainly be found.

When $z = 2$. For our algorithms in the setting $z = 2$, the additional torsion requirement comes from $\mathsf{IdealToIsogeny}^2_{\ell^\bullet}$, that we will use for some small prime $\ell \neq 2$. This algorithm makes calls to the sub-algorithm $\mathsf{IdealToIsogeny}^2_{\ell^e}$ for the value of e such that the ℓ^e-torsion is defined over $\mathbb{F}_{p^2}$. To reduce as much as possible the number of calls to this algorithm it is important to maximize the value of e. The best possible primes we can hope are the so-called "SIDH-primes" of the form $p = c2^f\ell^e - 1$ with a value of c as small as possible. In our case, we also need that $p + 1$ has a big prime factor N. Fortunately this additional constraint is quite easy to satisfy. We can simply iterate over small values of c until a suitable candidate is found. Asymptotically, we can expect to find it after $O(\log(p)^2)$ attempts. With this choice of prime we will be able to define the 2^f, ℓ^e-torsion and N-torsion on $\mathbb{F}_{p^2}$ (although not on the same curve) which is essentially the best we can hope for.

There is a bit of freedom regarding the choice of the exact value of f. Indeed, the minimal requirement is that $2^{2f} > N$. However, the probability of success of the executions of the algorithm $\mathsf{RandomGoodEndomorphism}$ inside $\mathsf{IdealToIsogeny}^2_{\ell^e}$ is directly impacted by the value of f. Given Proposition 7, the requirement $2^{2f} > N$, which implies that f is at least equal to $1/2\log(p)$, should be enough to ensure a negligible failure probability.

Remark 4. It might be worth taking a value of f significantly bigger than this lower bound. Indeed, even though a bigger value of f implies a smaller value of e (which we want to maximize as we explained), if f go as far as $2/3\log(p)$, then we might be able to compute the *2*dim representation of the endomorphism θ

directly (without having to cut it in half as we do now), which might allow us to use a faster $\mathsf{IsogEval}_{\mathsf{2dim}}$ algorithm. In the toy example that we provide, we take $\ell = 3$ and $2^f \approx 3^e \approx \sqrt{p}$, but a careful implementation might prove that taking $f \approx 2/3 \log(p)$ is better for effiency. We leave to future work the problem of answering to this interrogation.

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 ι such that $\iota(P_0) = Q_0$ 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). Our solution uses the fact that we can evaluate efficiently any endomorphisms of E_0 on points of E_0. 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 ι of norm coprime to N and we compute $Q_0 = \iota(P_0)$.

5.2 Example parameters for $\boldsymbol{\lambda = 128}$

We now describe concrete parameters for $\lambda = 128$ that are estimated to reach the NIST-I security level.

$\boldsymbol{z = 1}$**.** We found the following prime p using the $2x^n - 1$ method introduced in [11]. Given the requirement to have a power of 2 of size $\approx \sqrt{p}$ dividing $p^2 - 1$, we found that method more efficient than the XGCD method introduced in [14].

$$p + 1 = 2 \cdot (2^{32} \cdot 3029820973)^4$$

The biggest prime factor of $T|p^2 - 1$ is 106273. We found the following candidate value for N:

393726873534132384928444413735281749360848116391706711676559356809147

among the divisors of $p^2 + p + 1$. In that case, the points of order N can be defined over $\mathbb{F}_{p^{12}}$ since $\#E(\mathbb{F}_{p^{12}}) = (p^6 - 1)^2 = \big((p^3 + 1)(p^2 + p + 1)(p - 1)\big)^2$. This N is only of 228 bits which is a bit short of the desired 256 bits. However, given that the algorithm from [2] requires also $O(\sqrt{N})$ memory and that we have omitted logarithmic factors in the complexity estimate, this value of N should already be enough the make the OMIP very hard to break. If needed, more secure parameters can be found.

For $y = 4$, we can take the parameter α to be equal to $87 | p^2 - 1$. We have the equality

$$\begin{aligned} 2^{258} = \; & 87^2 \cdot N + (36411760473075724296387938437332815162 5)^2 \\ & + (5749641264945970805144830518307338850 74)^2 \end{aligned}$$

For $z = 2$ and $y = 2$. We found the prime p

$$p + 1 = 2^{131} \cdot 3^{80} \cdot 53$$

for which we have a value of $N = (p - 1)/2$ equal to

106629085248844946116312584635000023127652851438714460169853847954739308496 15871.

For $z = 2$, $y = 4$. The case $y = 4$ is slightly harder to find, because we need to find the parameter α. To facilitate the search, we looked for primes p of the form $2^f \cdot 3^e \cdot c - 1$ for which $2^h - (\alpha)^2 N$ is a prime equal to 1 mod 4 for some $h < 2f$ and $\alpha | c$. Using this idea, we found the prime p

$$p + 1 = 2^{131} \cdot 3^{71} \cdot 41 \cdot 293$$

for which we have a value of $N = (p - 1)/2$ equal to

122789151551561527349552969205324705816674834267793279136622473322950205374463

and we have $\alpha = 1$ and the following equality holds

$$\begin{aligned} 2^{258} = \; & N + (5383532256986440060630677217229367092 09)^2 \\ & + (2248444123779110532928579889607545578 60)^2 \end{aligned}$$

5.3 Sizes

In this section, we explain how to compute the sizes given in Table 1 for the parameters given in Section 5.2. We provide abstract formulas that are true for any security level λ.

On compression. To reduce the size of the public key and proof of our VRF family, we can use standard compression techniques for elliptic curves and their points (described for instance in [12, section 7.1]).

- Curves can be represented by their j-invariants which are always defined over $\mathbb{F}_{p^2}$ for supersingular curves.
- Any point of smooth order T can be represented by two scalars mod T (as coefficients of this point in a prescribed basis). The image of a full basis of order T under an isogeny can be represented as 3 scalars mod T (the fourth one can be recovered with the Weil pairing).

– A point of arbitrary order N defined over $\mathbb{F}_{p^k}$ can always be represented as one element over $\mathbb{F}_{p^k}$ and a bit.

Moreover, the points P, Q of order 2^f provided in $c^N_{y\mathsf{dim}}(E)$ for $y = 2, 4$ can be any basis of $E[2^f]$. Thus they can be computed in a deterministic manner from the curve E and we can omit them from $c^N_{y\mathsf{dim}}(E)$.

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 $\mathsf{DeuringVRF}_{y,z}$ family for all values of y, z from the security parameter λ. To comply with the security requirements, we take $\log(p) \approx 2\lambda$ and $f \approx \lambda$. We remind the reader that the integer k is such that $\mathbb{F}_{p^k}$ is the smallest field extension on which the points of order N are defined. The public key sizes will depend on this parameter. This parameter k is the only place where the value of z will impact the sizes which depend mostly on the value of y.

When $y = 2$, the public key is made of 2 curves, 2 points of order N and a basis of order 2^f. This can be represented in $8\lambda + 4k\lambda + 3\lambda$ bits. The proof is made of two curves, and two images of a basis of order 2^f, so this is $8\lambda + 6\lambda$.

When $y = 4$, the public key is simply made of one curve and two points of order N. This is $4\lambda + 4k\lambda$ bits. The proof is made of one curve and the image of one basis of order 2^f so this is $4\lambda + 3\lambda$.

In both cases, since we need the function f_{in} to be injective to avoid collisions, the input space cannot be bigger than N. Thus, we can take the input space has size $n_1(\lambda) = 2\lambda$ (of course this is an upper-bound, smaller values of $n_1(\lambda)$ can be considered). A value of $n_2(\lambda) = 2\lambda$ seems also to be reasonable.

These results are summarized in Table 3.

y	Input (bits)	Output (bits)	Public Key (bits)	Proof (bits)
2	2λ	2λ	$(4k+11)\lambda$	14λ
4	2λ	2λ	$(4k+4)\lambda$	7λ

Table 3. Size of the inputs, outputs, keys and proofs of the $\mathsf{DeuringVRF}_{y,z}$ schemes.

6 Prospects, open questions and future work

We have introduced a new family $\mathsf{DeuringVRF}_{y,z}$ of VRF protocols based on isogenies between supersingular curves by making use of two important subfields 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 in the random oracle model. Despite its novelty, this new assumption is related to various well-studied problem, and its hardness appears quite plausible. Interestingly, progress in the resolution of this progress could have very positive impacts on the efficiency of other areas of isogeny-based cryptography.

Our new scheme relies on various algorithmic block related to the Deuring correspondence that might be of independent interest. In particular, we have proposed an alternate method to perform the translation of ideal to their corresponding isogenies with the help of isogenies in dimension 2, that we believe could very well improve upon the state of the art algorithms.

Each variant of our new construction is more compact (by a good margin) than every other post-quantum VRF protocol and this is the main advantage of our new protocol.

The major remaining open question about our new protocol is its efficiency. While several related work have demonstrated that our construction will not be completely impractical, it is not easy to guess how fast it can concretely be. A part of the isogeny-community is currently dedicating significant efforts to implementing efficiently isogenies between abelian varieties of dimension 2 and 4 and we plan to be able to provide an efficient implementation of our protocol in the near future. Other isogeny-based protocols such as the SQIsign signature scheme could benefit from this implementation if our new $\mathsf{IdealToIsogeny}^2$ algorithm turns out to be as fast as we hope.

References

1. Basso, A., Maino, L., Pope, G.: Festa: Fast encryption from supersingular torsion attacks. Cryptology ePrint Archive (2023)
2. Bernstein, D.J., Feo, L.D., Leroux, A., Smith, B.: Faster computation of isogenies of large prime degree. ANTS (2020)
3. 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)
4. 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)
5. Castryck, W., Decru, T.: An efficient key recovery attack on sidh. EUROCRYPT (2023)
6. Castryck, W., Decru, T., Smith, B.: Hash functions from superspecial genus-2 curves using richelot isogenies. Journal of Mathematical Cryptology **14**(1), 268–292 (2020)
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.: On going work, soon to appear. (2023)
9. Chi-Domínguez, J.J., Pizarro-Madariaga, A., Riquelme, E.: Computing quotient groups of smooth order with applications to isogenies over higher-dimensional abelian varieties. Cryptology ePrint Archive (2023)
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. Costello, C.: B-sidh: supersingular isogeny diffie-hellman using twisted torsion. In: Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part II 26. pp. 440–463. Springer (2020)
12. Dartois, P., Leroux, A., Robert, D., Wesolowski, B.: Sqisignhd: New dimensions in cryptography. Cryptology ePrint Archive (2023)
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 (2023), https://eprint.iacr.org/2023/1197
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. Cryptology ePrint Archive, Report 2020/1222 (2020), https://eprint.iacr.org/2020/1222
21. Esgin, M.F., Steinfeld, R., Liu, D., Ruj, S.: Efficient hybrid exact/relaxed lattice proofs and applications to rounding and vrfs. Cryptology ePrint Archive (2022)
22. Fouotsa, T.B., Kutas, P., Merz, S.P., Ti, Y.B.: On the isogeny problem with torsion point information. In: PKC. pp. 142–161. Springer (2022)
23. Galbraith, S.D., Petit, C., Silva, J.: Identification protocols and signature schemes based on supersingular isogeny problems. In: ASIACRYPT (2017)
24. 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)
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. Kunzweiler, S.: Efficient computation of (2ˆ n, 2ˆ n)-isogenies. Cryptology ePrint Archive (2022)
33. Lai, Y.F.: Capybara and tsubaki: Verifiable random functions from group actions and isogenies. Cryptology ePrint Archive (2023)
34. 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)
35. Leroux, A.: Quaternion Algebra and isogeny-based cryptography. Ph.D. thesis, Ecole doctorale de l'Institut Polytechnique de Paris (2022)
36. Love, J., Boneh, D.: Supersingular curves with small noninteger endomorphisms. Open Book Series **4**(1), 7–22 (2020)
37. Maino, L., Martindale, C., Panny, L., Pope, G., Wesolowski, B.: A direct key recovery attack on sidh. In: EUROCRYPT (2023)
38. 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)
39. Richelot, F.: Ueber die integration eines merkwürdigen systems differentialgleichungen. (1842)
40. Robert, D.: Evaluating isogenies in polylogarithmic time. Cryptology ePrint Archive (2022)
41. Robert, D.: Breaking SIDH in polynomial time. EUROCRYPT (2023)
42. Voight, J.: Quaternion Algebras. Springer Graduate Texts in Mathematics series (2018)
43. Waterhouse, W.C.: Abelian varieties over finite fields. Annales Scientifiques de l'E.N.S (1969)
44. 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)
45. 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 Ideal to isogeny in dimension 1

We describe here the algorithm $\mathsf{IdealToIsogeny}^1_{\ell^e}$ ($\mathsf{IdealToIsogenyEichler}_{\ell^e}$ in [15, Algorithm 4]). We refer the reader to the explanations in [15] to obtain more details on the design, correctness and termination of this algorithm.

Algorithm 14 $\mathsf{IdealToIsogeny}^1_{\ell^{\mathsf{e}}}(\mathcal{O}, I, J, \varphi_J, P)$

```
Input: $I$ a left $\mathcal{O}$-ideal of norm $\ell^f$, an $(\mathcal{O}_0, \mathcal{O})$-ideal $J$ of norm $\ell^\bullet$ and $\varphi_J : E_0 \to E$
    the corresponding isogeny, a generator $P$ of $E[\ell^f] \cap \ker(\hat{\varphi}_J)$.
Output: $\varphi_I$ of degree $\ell^f$
Set $K = \overline{J} + \mathcal{O}\ell^f$.
Compute $\theta = \mathsf{SpecialEichlerNorm}_T(\mathcal{O}, K + \mathcal{O}\ell)$ of norm dividing $T^2$.
Select $\alpha \in I$ s.t $I = \mathcal{O}\langle \alpha, \ell^f \rangle$.
Compute $C, D$ s.t. $\alpha \cdot (C + D\theta) \in K$ and $\gcd(C, D, \ell) = 1$ using linear algebra.
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 $H_2 =$
    $\mathcal{O}\langle \overline{\theta}, n_2 \rangle$.
Compute $L_i = [J]^* H_i$, and $\varphi_i = [\varphi_J]_* \mathsf{IdealToIsogeny}_{n_i}(L_i)$ for $i \in \{1, 2\}$.
Compute $Q = \hat{\varphi}_2 \circ \varphi_1(P)$.
Compute $\varphi_I$ of kernel $\langle [C]P + [D]Q \rangle$.
return $\varphi_I$.
```

B Adapting Kani's Lemma in the bad cases.

As we explained in Remark 3, the description we gave of $\mathsf{IdealToIsogeny}^2_{\ell^{\mathsf{e}}}$ in Algorithm 7 is only correct when the output θ, β of $\mathsf{RandomGoodEndomorphism}$ satisfies $\ker \theta \cap \ker \beta = \{0\}$. Unfortunately, there is no way to guarantee this will always be true, thus we need a way to handle this situation.

Note that if $\ker \beta \cap \ker \hat{\theta} = \{0\}$, we can simply replace θ by its conjugate $\hat{\theta}$ and run $\mathsf{IdealToIsogeny}^2_{\ell^{\mathsf{e}}}$ from there.

The really problematic situation happens when $\ker \theta \cap \ker \hat{\theta} \cap \ker \beta$ is non-trivial (note that since θ and β are in the same quadratic order replacing β by $\hat{\beta}$ would not change anything).

In that case, it is easy to see that both θ and β must be lollipop endomorphisms of the form $\rho \circ \theta_0 \circ \hat{\rho}$ and $\rho \circ \beta_0 \circ \hat{\rho}$ for some isogeny $\rho : E_0 \to E$ of degree 2^m for an integer $m \leq h/2$. If we assume that we have taken the biggest such ρ, then we must have $\ker \theta_0 \cap \ker \beta_0 = \{0\}$ or $\ker \theta_0 \cap \ker \beta_0 = \{0\}$ (otherwise we would be able to replace ρ with an isogeny of bigger degree).

In that case, we explain how to make the computation by using points of order $h/2 + m$. If this is smaller than f (the exponent of the available power of 2 torsion), then we will be able to use this method efficiently.

The idea is that we can decompose the isogeny F as $R \circ F_0 \circ \hat{R}$ where $F_0 : E_0^2 \to E_0^2$ is the isogeny embedding θ_0, β_0 and R is the diagonal isogeny associated to ρ.

We can compute $\hat{\rho}$ and $\hat{R}$ using $\mathsf{IdealToIsogeny}^1_{2^{\mathsf{m}}}$ (since $m < f$). If we can compute the kernel of F_0, then we are done.

By the formulas of Lemma 1, we have that $\ker F_0 = \{\theta_0(x), -\hat{\beta}_0(x), x \in E_0[2^{h-2m}]\}$. Since, we only know how to evaluate θ, β we need to rewrite this formula as

$$\ker F_0 = \{(\hat{\rho} \circ \theta(y), -\hat{\rho} \circ \hat{\beta}(y)), y \in \frac{1}{2^{h-m}} \ker \rho\}$$

The points of $\frac{1}{2^{h-m}} \ker \hat{\rho}$ are contained in $E[2^h]$. If $h \leq f$, then we can directly compute the kernel of F_0 which is enough to compute F_0 and F, and this is enough for the application to $\mathsf{IdealToIsogeny}^2_{\ell^\mathsf{e}}$.

Unfortunately, our minimal requirement in terms of available torsion is 2^f with $2f > h$. Thus, in the cases where h is actually smaller than f, the approach we just described will not work. We can reduce the torsion requirement to $2^{h/2+m}$ by dividing F_0 in two isogenies $F_{0,2} \circ F_{0,1}$. With that, we can handle all the cases where $m < f - h/2$. The remaining cases appear to be out of reach of our ideas.

Since the probability of having such β and θ deacreases as m increases (we believe that even having $m > 1$ should be very rare already), it seems reasonable to expect that the approach we described above should be enough to handle all cases in practice.