Verifiable Random Function from the Deuring correspondence and higher dimensional isogenies.

Antonin Leroux

Eurocrypt, May 5, 2025

DGA, Université de Rennes, France

Summary

Main Contribution: A new post-quantum Verifiable Unpredictable Function (**VUF**) and the resulting constructions :

1. The most compact post-quantum Verifiable Random Function (**VRF**) scheme in the ROM with the fastest implementation of a VRF scheme.
2. The first (hash-and-sign) isogeny-based **signature scheme** in the standard model.

On Verifiable Functions

Verifiable Functions

- **KeyGen**: $\texttt{params} \rightarrow pk, sk$.
- **Eval**: $sk, x \rightarrow v, \pi$.
- **Verif**: $pk, x, v, \pi \rightarrow b$.

Verifiable Functions

- **KeyGen**: $\texttt{params} \rightarrow pk, sk$.
- **Eval**: $sk, x \rightarrow v, \pi$.
- **Verif**: $pk, x, v, \pi \rightarrow b$.

Basic Security Properties:

- **Correctness**.
- **Uniqueness**: there must be only one output that passes the verification.

Verifiable Functions

- **KeyGen**: $\texttt{params} \to pk, sk$.
- **Eval**: $sk, x \to v, \pi$.
- **Verif**: $pk, x, v, \pi \to b$.

Basic Security Properties:

- **Correctness**.
- **Uniqueness**: there must be only one output that passes the verification.

Moreover, the function needs to be hard to evaluate without pk.

1. **Unpredictability**: output is hard to compute $\Rightarrow$ VUF.
2. **Pseudo-Randomness**: output is indistinguishable from random $\Rightarrow$ VRF.

Some remarks.

In the ROM : VUF $\Rightarrow$ VRF

Some remarks.

In the ROM : VUF $\Rightarrow$ VRF

A VUF can either be seen as a:

1. Verifiable unpredictable function
2. Unique signature.

Some remarks.

In the ROM : VUF $\Rightarrow$ VRF

A VUF can either be seen as a:

1. Verifiable unpredictable function
2. Unique signature.

Signature schemes are usually not designed to be unique (especially in the PQ setting):

1. All signature based on sigma-protocols are not suitable (this rules out any SQIsign-based idea).
2. Lattices are inherently poorly adapted to uniqueness (due to noise).

Mathematical Background

Elliptic curve and isogenies

Elliptic Curve over $\mathbb{F}_{p^k}$:

$$y^2 = x^3 + ax + b, \quad a, b \in \mathbb{F}_{p^k}$$

Elliptic curve and isogenies

Elliptic Curve over $\mathbb{F}_{p^k}$:

$$y^2 = x^3 + ax + b, \quad a, b \in \mathbb{F}_{p^k}$$

Isogeny: rational map between elliptic curves.

$$\begin{aligned} \varphi \ : \ E &\longrightarrow E' \\ (x, y) &\longmapsto \left(\frac{g(x)}{h(x)}, y \left(\frac{g(x)}{h(x)} \right)' \right) \end{aligned}$$

When separable, the **degree** is $\deg(\varphi) = \#\ker(\varphi) \approx \deg g, \deg h$.

Elliptic curve and isogenies

Elliptic Curve over $\mathbb{F}_{p^k}$:

$$y^2 = x^3 + ax + b, \quad a, b \in \mathbb{F}_{p^k}$$

Isogeny: rational map between elliptic curves.

$$\begin{aligned} \varphi \ : \ E &\longrightarrow E' \\ (x, y) &\longmapsto \left(\frac{g(x)}{h(x)}, y \left(\frac{g(x)}{h(x)} \right)' \right) \end{aligned}$$

When separable, the **degree** is $\deg(\varphi) = \#\ker(\varphi) \approx \deg g, \deg h$.

Endomorphisms, supersingular curves and quaternions

An **endomorphism** is an isogeny $\varphi : E \to E$.

$End(E)$ is a ring. In characteristic p, it has dimension 2 or 4.

Endomorphisms, supersingular curves and quaternions

An **endomorphism** is an isogeny $\varphi : E \to E$.

$End(E)$ is a ring. In characteristic p, it has dimension 2 or 4.

Supersingular curves $\Leftrightarrow$ $\mathrm{End}(E)$ is a **max. order** in $\mathcal{B}(-q,-p)$.

$$\mathcal{B}(-q,-p) = \mathbb{Q} + i\mathbb{Q} + j\mathbb{Q} + k\mathbb{Q}, \qquad i^2 = -q, j^2 = -p.$$

Endomorphisms, supersingular curves and quaternions

An **endomorphism** is an isogeny $\varphi : E \to E$.

$End(E)$ is a ring. In characteristic p, it has dimension 2 or 4.

Supersingular curves $\Leftrightarrow$ $\mathrm{End}(E)$ is a **max. order** in $\mathcal{B}(-q, -p)$.

$$\mathcal{B}(-q, -p) = \mathbb{Q} + i\mathbb{Q} + j\mathbb{Q} + k\mathbb{Q}, \qquad i^2 = -q, j^2 = -p.$$

The problem of recovering $\mathrm{End}(E)$ from E is the fundamental problem behind isogeny-based cryptography.

The Deuring Correspondence

p : prime char, $\mathcal{B}(-q,-p)$ where $q>0$ depends only on p.

Supersingular elliptic curves over $\mathbb{F}_{p^2}$ E (up to Galois conjugacy)	Maximal Orders in $\mathcal{B}(-q,-p)$ $\mathcal{O} \cong \mathrm{End}(E)$
Isogeny with $\varphi : E \to E_1$	Ideal I_φ left $\mathcal{O}$-ideal
Degree $\deg(\varphi)$	Norm $n(I_\varphi)$

The Deuring Correspondence

p : prime char, $\mathcal{B}(-q,-p)$ where $q>0$ depends only on p.

Supersingular elliptic curves over $\mathbb{F}_{p^2}$ E (up to Galois conjugacy)	Maximal Orders in $\mathcal{B}(-q,-p)$ $\mathcal{O} \cong \mathrm{End}(E)$
Isogeny with $\varphi : E \to E_1$	Ideal I_φ left $\mathcal{O}$-ideal
Degree $\deg(\varphi)$	Norm $n(I_\varphi)$

In the quaternion world all relevant operations are efficient!

Isogeny representations

Isogeny representations

What does it mean to "compute an isogeny"?

Most of the times: compute the polynomials defining it.

Isogeny representations

What does it mean to "compute an isogeny"?

Most of the times: compute the polynomials defining it.

Crypto sizes: the degree might be too big!

Solution when N is smooth: factor φ in isogenies of prime degree.

Isogeny representations

What does it mean to "compute an isogeny"?
Most of the times: compute the polynomials defining it.

Crypto sizes: the degree might be too big!
Solution when N is smooth: factor φ in isogenies of prime degree.

Not satisfactory in general $\rightarrow$ abstraction of isogeny representation:

Representation $\texttt{xx}$: $s_{\texttt{xx}}^{\varphi}$ for unique isog. $\varphi : E \to E'$ of deg. N is data with algorithm $\texttt{IsogEval}_{\texttt{xx}}$ to evaluate φ.

Isogeny representations

What does it mean to "compute an isogeny"?
Most of the times: compute the polynomials defining it.

Crypto sizes: the degree might be too big!
Solution when N is smooth: factor φ in isogenies of prime degree.

Not satisfactory in general $\rightarrow$ abstraction of isogeny representation:

Representation $\texttt{xx}$: $s_{\texttt{xx}}^{\varphi}$ for unique isog. $\varphi : E \to E'$ of deg. N is data with algorithm $\texttt{IsogEval}_{\texttt{xx}}$ to evaluate φ.

Efficient: $s_{\texttt{xx}}^{\varphi}$ has size polylog (pN), $\texttt{IsogEval}_{\texttt{xx}}$ runs in polylog $\left(p^k N\right)$ on input contained in $E[\mathbb{F}_p^k]$.

Existing representations: an overview

1. `Poly`: defining polynomials.

Existing representations: an overview

1. `Poly`: defining polynomials.
2. `Ker`: generator(s) of the kernel (points of the domain EC).

Existing representations: an overview

1. `Poly`: defining polynomials.
2. `Ker`: generator(s) of the kernel (points of the domain EC).
3. `Id`: ideal associated to φ under the Deuring correspondence.
   Efficient!

Existing representations: an overview

1. `Poly`: defining polynomials.
2. `Ker`: generator(s) of the kernel (points of the domain EC).
3. `Id`: ideal associated to φ under the Deuring correspondence. Efficient!
4. `HD`: $\varphi(P), \varphi(Q)$ for P, Q a basis of $E[D]$ where D is smooth. Used to embed φ into a D-isogeny of higher dimension. Efficient!

The `HD` representation appeared with the attacks to break SIDH [CD22,MMPPW22,R22].

Isogeny representations: algorithmic strengths & weaknesses

- $N = \deg \varphi$,
- $N' = \max_{d \mid N} d$,
- $k = \max_{d \mid N,\ E[d] \subset E[\mathbb{F}_{p^k}]} k$

	Required Knowledge	Compactness	Computation Efficiency	Evaluation Efficiency	Sampling Efficiency
`Poly`	$\varnothing$	$\mathrm{poly}(N')$	$\mathrm{poly}(N')$	$\mathrm{poly}(N')$	$\mathrm{poly}(N')$
`Ker`	$\varnothing$	$\mathrm{poly}(\min(N, k))$	$\mathrm{poly}(N')$	$\mathrm{poly}(N)$	$\mathrm{poly}(k \log(N))$
`Id`	$\mathrm{End}(E)$	$\mathrm{polylog}(N)$	$\mathrm{polylog}(N)$	$\mathrm{polylog}(N)$	$\mathrm{polylog}(N)$
`HD`	$\varnothing$	$\mathrm{polylog}(N)$	$\mathrm{poly}(N')$	$\mathrm{polylog}(N)$	$\mathrm{poly}(N')$

Isogeny representations: algorithmic strengths & weaknesses

- $N = \deg \varphi$,
- $N' = \max_{d \mid N} d$,
- $k = \max_{d \mid N,\ E[d] \subset E[\mathbb{F}_{p^k}]} k$

	Required Knowledge	Compactness	Computation Efficiency	Evaluation Efficiency	Sampling Efficiency
`Poly`	$\varnothing$	$\text{poly}(N')$	$\text{poly}(N')$	$\text{poly}(N')$	$\text{poly}(N')$
`Ker`	$\varnothing$	$\text{poly}(\min(N, k))$	$\text{poly}(N')$	$\text{poly}(N)$	$\text{poly}(k \log(N))$
`Id`	$\text{End}(E)$	$\text{polylog}(N)$	$\text{polylog}(N)$	$\text{polylog}(N)$	$\text{polylog}(N)$
`HD`	$\varnothing$	$\text{polylog}(N)$	$\text{poly}(N')$	$\text{polylog}(N)$	$\text{poly}(N')$

Conclusion: everything is efficient when N is smooth, but all representations are not equivalent in the generic case.

Worst case (or best case for us): N is **prime**.

The construction

VUF from isogeny representation: an overview

Building a VUF from isogeny representations of big prime degree:

- **Keys**: hard to recover the secret key

VUF from isogeny representation: an overview

Building a VUF from isogeny representations of big prime degree:

- **Keys**: hard to recover the secret key
  $\rightarrow pk = E, sk = \mathrm{End}(E)$.

VUF from isogeny representation: an overview

Building a VUF from isogeny representations of big prime degree:

- **Keys**: hard to recover the secret key
  $\rightarrow pk = E, sk = \mathrm{End}(E)$.
- **Input**: easy to sample from the public key

VUF from isogeny representation: an overview

Building a VUF from isogeny representations of big prime degree:

- **Keys**: hard to recover the secret key
  $\rightarrow pk = E, sk = \mathrm{End}(E)$.
- **Input**: easy to sample from the public key
  $\rightarrow P_x = H(E, x) \in E[N]$.

VUF from isogeny representation: an overview

Building a VUF from isogeny representations of big prime degree:

- **Keys**: hard to recover the secret key
  $\rightarrow pk = E, sk = \mathrm{End}(E)$.
- **Input**: easy to sample from the public key
  $\rightarrow P_x = H(E, x) \in E[N]$.
- **Output**: hard to compute without the secret key

VUF from isogeny representation: an overview

Building a VUF from isogeny representations of big prime degree:

- **Keys**: hard to recover the secret key
  $\rightarrow pk = E, sk = \mathrm{End}(E)$.
- **Input**: easy to sample from the public key
  $\rightarrow P_x = H(E, x) \in E[N]$.
- **Output**: hard to compute without the secret key
  $\rightarrow v = \varphi(E)$ with $\ker \varphi = \langle P_x \rangle$, $\deg \varphi$ a big prime.

VUF from isogeny representation: an overview

Building a VUF from isogeny representations of big prime degree:

- **Keys**: hard to recover the secret key
  $\rightarrow pk = E, sk = \mathrm{End}(E)$.
- **Input**: easy to sample from the public key
  $\rightarrow P_x = H(E, x) \in E[N]$.
- **Output**: hard to compute without the secret key
  $\rightarrow v = \varphi(E)$ with $\ker \varphi = \langle P_x \rangle$, $\deg \varphi$ a big prime.
- **Proof:** easy to verify from the public key

VUF from isogeny representation: an overview

Building a VUF from isogeny representations of big prime degree:

- **Keys**: hard to recover the secret key
  $\rightarrow pk = E, sk = \mathrm{End}(E)$.
- **Input**: easy to sample from the public key
  $\rightarrow P_x = H(E, x) \in E[N]$.
- **Output**: hard to compute without the secret key
  $\rightarrow v = \varphi(E)$ with $\ker \varphi = \langle P_x \rangle$, $\deg \varphi$ a big prime.
- **Proof:** easy to verify from the public key
  $\rightarrow$ `HD` representation of φ.

The `Id` representation will main tool behind the evaluation process of our VUF.

The main algorithmic challenges

Use of the ideal representation to compute $s^{\varphi}_{\mathtt{HD}}$ (i.e. evaluate φ) from I_x.

The main algorithmic challenges

Use of the ideal representation to compute $s_{\mathrm{HD}}^{\varphi}$ (i.e. evaluate φ) from I_x.

This is done in a single execution of the ideal-to-isogeny algorithm from SQIsign2D-West [BDDFLMPRW24].

A new hard problem.

N-FIXDIO$_{\mathtt{xx}}$ **oracle:**

Input: $E, P \in E[N]$

Output: isogeny representation $s^{\varphi}_{\mathtt{xx}}$ for the N-isogeny $\varphi : E \to E/\langle P \rangle$

One-More Isogeny Problem (OMIP$_{\mathtt{xx}}$) Given access to the N-FIXDIO$_{\mathtt{xx}}$ oracle on input E, compute the codomain of an isogeny not given as the output of N-FIXDIO$_{\mathtt{xx}}$.

The OMIPxx is related to the complexity of $\mathtt{IsogEval}_{\mathtt{ker}}$.

First application: VRF

	Public Key (bytes)	Proof (bytes)	Unrestricted evaluation	Uniqueness	Assumption	Security level
LB - VRF [EK+21]	3.3K	4.9K	✗	Computational	MSIS/MLWE(Latt.)	128
SL - VRF [BD+22]	48	40K	✓	Computational	LowMC(Hash)	128
LaV [ESLR22]	8.81K	10.27K	✓	Computational	MSIS/MLWR(Latt.)	128
CAPYBARA [L23]	8.3K	39K	✓	Computational	DDH(Isog.)	< 128
TSUBAKI [L23]	5.3K	34K	✓	Computational	sDDH(Isog.)	< 128
`DeuringVRF`	192	256	✓	Unconditional	$\mathrm{OMIP}_{\dim 2}$(Isog.)	128

Table 1: Comparison of the sizes and security properties of several post-quantum VRF schemes

First application: VRF

	Public Key (bytes)	Proof (bytes)	Unrestricted evaluation	Uniqueness	Assumption	Security level
LB - VRF [EK+21]	3.3K	4.9K	✗	Computational	MSIS/MLWE(Latt.)	128
SL - VRF [BD+22]	48	40K	✓	Computational	LowMC(Hash)	128
LaV [ESLR22]	8.81K	10.27K	✓	Computational	MSIS/MLWR(Latt.)	128
CAPYBARA [L23]	8.3K	39K	✓	Computational	DDH(Isog.)	< 128
TSUBAKI [L23]	5.3K	34K	✓	Computational	sDDH(Isog.)	< 128
`DeuringVRF`	192	256	✓	Unconditional	$\mathrm{OMIP}_{\mathrm{dim}\,2}$(Isog.)	128

Table 1: Comparison of the sizes and security properties of several post-quantum VRF schemes

Proof of concept C implementation :

1. Verification: 18 ms
2. Signature : 160 ms

Hope of improvement: $\sim$x3

Second Application : hash-and-sign signature

Comparison with SQIsign: performance and sizes are comparable (within a factor 2). More precisely:

1. Faster Signing
2. Slower Evaluation
3. Bigger Signature

Second Application : hash-and-sign signature

Comparison with SQIsign: performance and sizes are comparable (within a factor 2). More precisely:

1. Faster Signing
2. Slower Evaluation
3. Bigger Signature

The main interest is that the principle is very different technique (and falls within the hash-and-sign paradigm). It might open new possibilities for isogeny-based cryptography.

Conclusion

Future work:

- Analysis of the OMIP (in particular in the quantum setting).
- More constructions from this framework?

Conclusion

Future work:

- Analysis of the OMIP (in particular in the quantum setting).
- More constructions from this framework?

Main Take-away: VUF/VRF might be the most promising application for isogeny-based cryptography.

`https://eprint.iacr.org/2023/1251`