Cryptographic Categories

Andrea Basso Luca De Feo Sikhar Patranabis **Ilinca Radulescu*** Benjamin Wesolowski

*ENS de Lyon and CNRS

April 29, 2025

Overview

Motivation

Motivation

- There are several versions of SQIsign, all with the same structure, so we want to extract what's essential.

Motivation

- There are several versions of SQIsign, all with the same structure, so we want to extract what's essential.
- SQIsign without all the algebraic machinery - more accessible conceptually.

Motivation

- There are several versions of SQIsign, all with the same structure, so we want to extract what's essential.
- SQIsign without all the algebraic machinery - more accessible conceptually.
- A new perspective to formalize such concepts using category theory.

Background

Introduction to Categories

A (small) category is defined by a set of *objects* and a set of *morphisms* with the following properties:

1. Every morphism ϕ has a *domain* object D and a *codomain* object C, denoted by $\phi : D \to C$.

$$D \xrightarrow{\phi} C$$

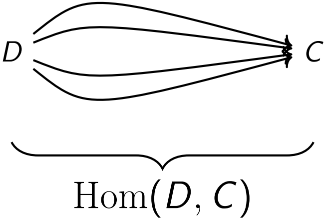
Introduction to Categories

A (small) category is defined by a set of *objects* and a set of *morphisms* with the following properties:

1. Every morphism ϕ has a *domain* object D and a *codomain* object C, denoted by $\phi : D \to C$.

2. The set of all morphisms with domain D and codomain C is denoted by $\mathrm{Hom}(D, C)$; this is called a *homset*.

$$D \xrightarrow{\phi} C$$



Introduction to Categories

3. There exists a *composition law*, written $\circ$, that maps a morphism $\phi : A \to B$ and a morphism $\psi : B \to C$ to a morphism $\psi \circ \phi : A \to C$, and that is associative: $(\phi \circ \psi) \circ \chi = \phi \circ (\psi \circ \chi)$.

$$\underbrace{A \xrightarrow{\phi} B \xrightarrow{\psi} C}_{\psi \circ \phi}$$

Introduction to Categories

3. There exists a *composition law*, written $\circ$, that maps a morphism $\phi : A \to B$ and a morphism $\psi : B \to C$ to a morphism $\psi \circ \phi : A \to C$, and that is associative: $(\phi \circ \psi) \circ \chi = \phi \circ (\psi \circ \chi)$.

$$\underbrace{A \xrightarrow{\phi} B \xrightarrow{\psi} C}_{\psi \circ \phi}$$

4. For every object A, there exists a morphism $1_A : A \to A$ such that $1_A \circ \phi = \phi$ and $\psi \circ 1_A = \psi$ for every $\phi : Z \to A$ and every $\psi : A \to B$.

$$A \xrightarrow{1_A} A$$

Our Axioms

Intuitive Axioms

A *cryptographic category* must satisfy the following computational axioms:

- Uniqueness
- Origin
- Walk

A Running Example - Classical SQIsign

Fix a large $N \in \mathbb{Z}$ and a large prime p.

- **Objects**: Pairs of supersingular elliptic curves over $\mathbb{F}_{p^2}$ together with their N-torsion points, $(E, E[N])$.
- **Morphisms**: Isogenies between the elliptic curves, $\psi : E \to E'$.

Uniqueness

Every object and every morphism has a unique representation as a binary string

Uniqueness

Every object and every morphism has a unique representation as a binary string

instantiation

- For unique representation of the objects, use the j-invariants for the isomorphism classes of elliptic curves, together with a basis of $E[N]$.
- For the unique representation of isogenies, use a deterministic algorithm to pick 2-torsion points, 3-torsion points, $\cdots$ $\log N$-torsion points and use interpolation to put everything together.

Origin

There exists an *origin* object O whose representation is known.

Origin

There exists an *origin* object O whose representation is known.

$\downarrow$ instantiation

Given p a prime, there always exists a polynomial time algorithm to find an origin curve E_0, of known endomorphism ring.

Walk

Definition

A *walk*, $\mathcal{W}$, is a deterministic algorithm which takes as input an object A and random coins r and produces a morphism $\psi : A \to B$. When r is uniformly random coins, B follows distribution μ. Moreover, $\mathcal{W}(A) := \mathcal{W}(A; r)$ is a randomized algorithm for r uniformly random coins.

Walk

Definition

A *walk*, $\mathcal{W}$, is a deterministic algorithm which takes as input an object A and random coins r and produces a morphism $\psi : A \to B$. When r is uniformly random coins, B follows distribution μ. Moreover, $\mathcal{W}(A) := \mathcal{W}(A; r)$ is a randomized algorithm for r uniformly random coins.

There exists a walk in the category called Walk.

Walk

Definition

A *walk*, $\mathcal{W}$, is a deterministic algorithm which takes as input an object A and random coins r and produces a morphism $\psi : A \to B$. When r is uniformly random coins, B follows distribution μ. Moreover, $\mathcal{W}(A) := \mathcal{W}(A; r)$ is a randomized algorithm for r uniformly random coins.

There exists a walk in the category called Walk. → A random walk in the ℓ-isogeny graph. Due to the rapid mixing properties of the ℓ-isogeny graph, the target curve, E_B, follows the uniform distribution.

The fingerprint

Definition

A **fingerprint**, fp, is a collection of maps:

$$\mathrm{fp} : \mathrm{Hom}(-, -) \to \mathcal{M} \cup \{\perp\}$$

where $\perp$ indicates undefined values.

Instantiation to elliptic curves

Let $\psi : E_1 \to E_2$ be an isogeny, ℓ a small prime and $\ker(\psi) \cap E_1[\ell^n] = \ker(\psi)[\ell^n]$

$$\mathrm{prefp}(\psi) = \begin{cases} \ker(\psi)[\ell^n], & \text{if } \ker(\psi)[\ell^n] \cong \mathbb{Z}/\ell^n\mathbb{Z} \\ \perp, & \text{otherwise.} \end{cases}$$

Instantiation to elliptic curves

Let $\psi : E_1 \to E_2$ be an isogeny, ℓ a small prime and $\ker(\psi) \cap E_1[\ell^n] = \ker(\psi)[\ell^n]$

$$\mathrm{prefp}(\psi) = \begin{cases} \ker(\psi)[\ell^n], & \text{if } \ker(\psi)[\ell^n] \cong \mathbb{Z}/\ell^n\mathbb{Z} \\ \perp, & \text{otherwise.} \end{cases}$$

However,

$$\ker(\psi)[\ell^n] = \langle P \rangle = \langle aP_1 + bQ_1 \rangle.$$

Instantiation to elliptic curves

Let $\psi : E_1 \to E_2$ be an isogeny, ℓ a small prime and $\ker(\psi) \cap E_1[\ell^n] = \ker(\psi)[\ell^n]$

$$\text{prefp}(\psi) = \begin{cases} \ker(\psi)[\ell^n], & \text{if } \ker(\psi)[\ell^n] \cong \mathbb{Z}/\ell^n\mathbb{Z} \\ \perp, & \text{otherwise.} \end{cases}$$

However,

$$\ker(\psi)[\ell^n] = \langle P \rangle = \langle aP_1 + bQ_1 \rangle.$$

Definition

$$\mathrm{fp}(\psi) = \begin{cases} (1, a^{-1}b), & \text{if } a \in (\mathbb{Z}/\ell^n\mathbb{Z})^\times \\ (b^{-1}a, 1), & \text{if } a \notin (\mathbb{Z}/\ell^n\mathbb{Z})^\times \wedge b \in (\mathbb{Z}/\ell^n\mathbb{Z})^\times. \end{cases}$$

Not So Intuitive Axioms

We want the *fingerprint* to have some, if not all of the following properties, depending on which protocol we want to obtain:

- Evaluatable
- Walkability
- Hard
- Triangularizability
- Indistinguishable Walkability

Evaluatable

Given ϕ, one can find $\mathrm{fp}(\phi)$ efficiently.

Evaluatable

Given ϕ, one can find $\mathrm{fp}(\phi)$ efficiently.

$\downarrow$ instantiation

Given $\psi : E_1 \to E_2$, find a, b such that $P = aP_1 + bQ_1$ for P a generator of $\ker(\psi)[\ell^n]$.

Evaluatable

Given ϕ, one can find $\mathrm{fp}(\phi)$ efficiently.

$\downarrow$ instantiation

Given $\psi : E_1 \to E_2$, find a, b such that $P = aP_1 + bQ_1$ for P a generator of $\ker(\psi)[\ell^n]$.

$\downarrow$ equivalent

$\ell-$smooth DLP for Elliptic Curves

Walkability

There exists a randomized algorithm, Walkable, that on input an object A returns an object B and a morphism $\psi : A \rightarrow B$ such that B follows distribution μ and $\mathrm{fp}(\psi) \in \mathcal{M}$.

Walkability

There exists a randomized algorithm, Walkable, that on input an object A returns an object B and a morphism $\psi : A \to B$ such that B follows distribution μ and $\mathrm{fp}(\psi) \in \mathcal{M}$.

$$\Big\downarrow \text{instantiation}$$

Random walk in the ℓ-isogney graph. A random walk in the ℓ-isogeny graph. Due to the rapid mixing properties of the ℓ-isogeny graph, the target curve, E_B, follows the uniform distribution.

Hard

$\mathcal{L} = \{(A, (\phi, \psi)) \mid \exists B$ such that $\phi : A \rightarrow B, \psi : A \rightarrow B, \mathrm{fp}(\phi) \neq \mathrm{fp}(\psi)$, and $\mathrm{fp}(\phi), \mathrm{fp}(\psi) \neq \perp\}$ is a hard language

Hard

$\mathcal{L} = \{(A, (\phi, \psi)) \mid \exists B$ such that $\phi : A \to B, \psi : A \to B, \mathrm{fp}(\phi) \neq \mathrm{fp}(\psi)$, and $\mathrm{fp}(\phi), \mathrm{fp}(\psi) \neq \perp\}$ is a hard language

↓ instantiation

It's (computationally) hard to find two isogenies, $\phi, \psi : E_1 \to E_2$ such that $\ker(\phi)[\ell^n], \ker(\psi)[\ell^n] \cong \mathbb{Z}/\ell^n\mathbb{Z}$ and $\ker(\phi)[\ell^n] \neq \ker(\psi)[\ell^n]$

Hard

$\mathcal{L} = \{(A, (\phi, \psi)) \mid \exists B$ such that $\phi : A \to B, \psi : A \to B, \mathrm{fp}(\phi) \neq \mathrm{fp}(\psi)$, and $\mathrm{fp}(\phi), \mathrm{fp}(\psi) \neq \perp\}$ is a hard language

↓ instantiation

It's (computationally) hard to find two isogenies, $\phi, \psi : E_1 \to E_2$ such that $\ker(\phi)[\ell^n], \ker(\psi)[\ell^n] \cong \mathbb{Z}/\ell^n\mathbb{Z}$ and $\ker(\phi)[\ell^n] \neq \ker(\psi)[\ell^n]$

→

Computationally hard to find two "distinct" isogenies $\phi, \psi : E_1 \to E_2$

Hard

$\mathcal{L} = \{(A, (\phi, \psi)) \mid \exists B$ such that $\phi : A \to B, \psi : A \to B, \mathrm{fp}(\phi) \neq \mathrm{fp}(\psi)$, and $\mathrm{fp}(\phi), \mathrm{fp}(\psi) \neq \perp\}$ is a hard language

instantiation

It's (computationally) hard to find two isogenies, $\phi, \psi : E_1 \to E_2$ such that $\ker(\phi)[\ell^n], \ker(\psi)[\ell^n] \cong \mathbb{Z}/\ell^n\mathbb{Z}$ and $\ker(\phi)[\ell^n] \neq \ker(\psi)[\ell^n]$

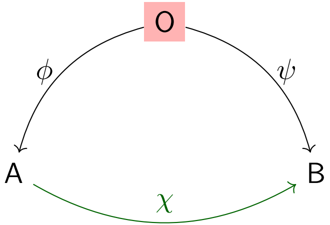
Computationally hard to find two "distinct" isogenies $\phi, \psi : E_1 \to E_2$

$$\begin{tikzcd} E_1 \arrow[r, bend left, "\phi"] \arrow[r, bend right, "\psi"'] & E_2 \end{tikzcd}$$

ONEEND problem

Triangularizability

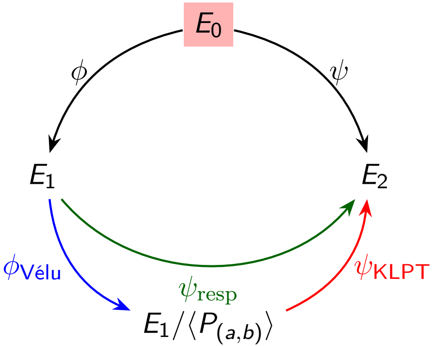
There exists an efficient polynomial time algorithm, Triangle, that on inputs $\phi : O \to A, \psi : O \to B, m \in \mathcal{M}$, returns $\chi : A \to B$ such that $\mathrm{fp}(\chi) = m$.



Instantiated Triangle

Input: $\phi, \psi, (a, b)$
1. Compute $\phi_{\text{Vélu}} : E_1 \to E_1/\langle P_{(a,b)} \rangle$ using Vélu's formulas.
2. Let ℓ' be a prime coprime to ℓ. Use KLPT$(\phi_{\text{Vélu}} \circ \phi, \psi, \ell')$ to compute $\psi_{\text{KLPT}} : E_1/\langle P_{(a,b)} \rangle \to E_2$ of degree ℓ'.

Return: $\psi_{\mathrm{resp}} = \psi_{\text{KLPT}} \circ \phi_{\text{Vélu}}$



Indistinguishable Walkability

There exists an efficient polynomial time algorithm IndWalk such that for any $\phi : O \to A$, the output of IndWalk(A) is (perfectly, statistically, or computationally) indistinguishable from the following distribution:

1. Run $(B, \psi) \leftarrow \mathcal{W}(A)$.
2. Sample m from $\mathcal{M}$ with distribution μ: $m \xleftarrow{\$} \mu(\mathcal{M})$.
3. Return $\chi \leftarrow$ Triangle$(\phi, \psi \circ \phi, m)$, where $\chi : A \to B$ such that $\mathrm{fp}(\chi) = m$.

$$\begin{tikzcd}
 & O \arrow[dl, dotted, "\phi"'] & \\
A \arrow[rr, bend left, "\psi"] \arrow[rr, bend right, "\chi"'] & & B
\end{tikzcd}$$

Instantiated Indistinguishable Walkability

Input: E_1

1. Sample $G \subseteq E_1[\ell^n]$ such that $G \cong \mathbb{Z}/\ell^n\mathbb{Z}$
2. Use Vélu's formulas to compute $\tau \colon E_1 \to E_1/G$
3. Take a random walk in the ℓ'-isogeny graph from E_1/G:
   $(E_2, \sigma) \leftarrow \mathrm{Walk}(E_1/G, n)$

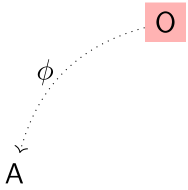
Return: $\psi_{rsp} = \sigma \circ \tau$, where $\psi_{rsp} \colon E_1 \to E_2$

$$\begin{tikzcd} E_1 \arrow[r, "\tau"] \arrow[rr, bend right, "\psi_{rsp} = \sigma \circ \tau"'] & E_1/G \arrow[r, "\sigma"] & E_2 \end{tikzcd}$$

Protocols

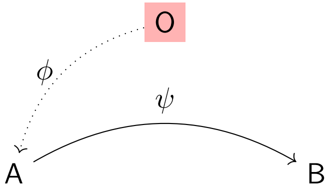
Basic Signature

- **KeyGen.** Run $(A, \phi) \leftarrow \mathrm{Walk}(O)$, return A as the public key and ϕ as the secret key.



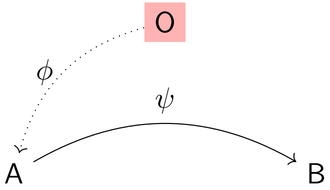
Basic Signature

- **KeyGen.** Run $(A, \phi) \leftarrow \mathsf{Walk}(O)$, return A as the public key and ϕ as the secret key.
- **Commitment.** Run $(B, \psi) \leftarrow \mathsf{Walk}(A)$, and return B as the commitment object.



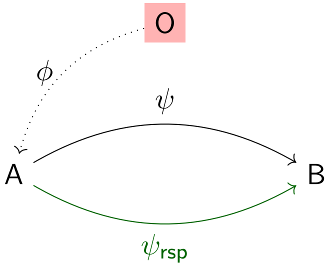
Basic Signature

- **KeyGen.** Run $(A, \phi) \leftarrow \mathrm{Walk}(O)$, return A as the public key and ϕ as the secret key.
- **Commitment.** Run $(B, \psi) \leftarrow \mathrm{Walk}(A)$, and return B as the commitment object.
- **Challenge.** Verifier selects a random fingerprint $m \in \mathcal{C}$ and sends this fingerprint to the Signer.



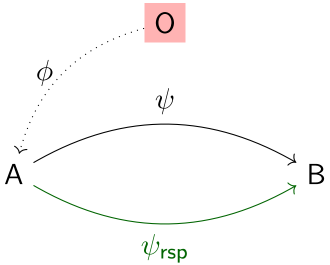
Basic Signature

- **KeyGen.** Run $(A, \phi) \leftarrow \mathsf{Walk}(O)$, return A as the public key and ϕ as the secret key.
- **Commitment.** Run $(B, \psi) \leftarrow \mathsf{Walk}(A)$, and return B as the commitment object.
- **Challenge.** Verifier selects a random fingerprint $m \in \mathcal{C}$ and sends this fingerprint to the Signer.
- **Response.** Run $\psi_{\mathsf{rsp}} \leftarrow \mathsf{Triangle}(\phi, \psi \circ \phi, m)$ to obtain a morphism ψ_{rsp} such that $\mathrm{fp}(\psi_{\mathsf{rsp}}) = m$.



Basic Signature

- **KeyGen.** Run $(A, \phi) \leftarrow \mathsf{Walk}(O)$, return A as the public key and ϕ as the secret key.
- **Commitment.** Run $(B, \psi) \leftarrow \mathsf{Walk}(A)$, and return B as the commitment object.
- **Challenge.** Verifier selects a random fingerprint $m \in \mathcal{C}$ and sends this fingerprint to the Signer.
- **Response.** Run $\psi_{\mathsf{rsp}} \leftarrow \mathsf{Triangle}(\phi, \psi \circ \phi, m)$ to obtain a morphism ψ_{rsp} such that $\mathrm{fp}(\psi_{\mathsf{rsp}}) = m$.
- **Verification.** Verify $\mathrm{fp}(\psi_{\mathsf{rsp}}) = m$.



Remarks on Basic Signature

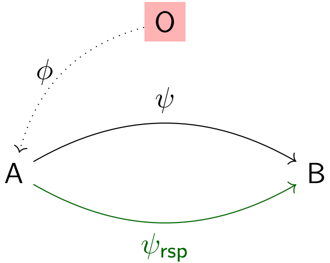
- Emulates SQIsign.

Remarks on Basic Signature

- Emulates SQIsign.
- In classical SQIsign, the challenge step prescribes an isogeny, but in the running example, the challenge step prescribes a kernel. BUT prescribing a kernel = prescribing an isogeny.

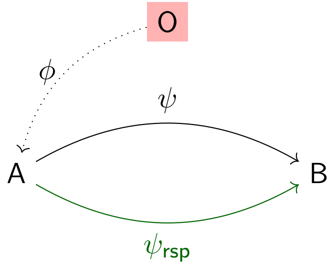
Secure?

- Secret key, ϕ, hard to recover $\checkmark$
(Hard and Triangularizability)



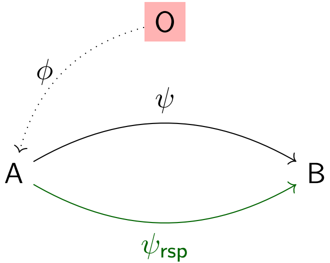
Secure?

- Secret key, ϕ, hard to recover ✓
(Hard and Triangularizability)
- Special soundness ✓
$(B, m, \psi_{\mathsf{rsp}}), (B, m', \psi'_{\mathsf{rsp}}) \to (A, (\psi_{\mathsf{rsp}}, \psi'_{\mathsf{rsp}})) \in \mathcal{L}$



Secure?

- Secret key, ϕ, hard to recover ✓
(Hard and Triangularizability)
- Special soundness ✓
$(B, m, \psi_{\text{rsp}}), (B, m', \psi'_{\text{rsp}}) \to (A, (\psi_{\text{rsp}}, \psi'_{\text{rsp}})) \in \mathcal{L}$
- Zero Knowledge ✓
(IndWalk)



Conclusion

Conclusion

- Just exploiting the axioms we can define SQIsign.

Conclusion

- Just exploiting the axioms we can define SQIsign.
- Can we instantiate the axioms differently to obtain different properties?

Yes, we can work with levels.

Conclusion

- Just exploiting the axioms we can define SQIsign.
- Can we instantiate the axioms differently to obtain different properties?

Yes, we can work with levels.

- Can we obtain other protocols besides than digital signature schemes?

Yes, we also obtain a chamaeleon hash function.