KU LEUVEN

fwo

Proving Knowledge of Isogenies: a Unified Approach

IBM Zurich

J. K. Eriksen, R. Invernizzi*, J. Spiessens, F. Vercauteren

COSIC - KU Leuven

24 jun 2026

Outline

Outline

Proof of Knowledge

Given a *relation*

$$\mathcal{R} = \{(x, w) \mid x \in L, w \in W(x)\}$$

- $x \in L$ is a *statement* of a language
- w is a *witness* for x
- *proof of knowledge*: a prover convinces a verifier of the knowledge of $(x, w) \in \mathcal{R}$

Proof of Knowledge

- *completeness*: an honest prover succeeds
- *soundness*: a dishonest prover fails
- *knowledge soundness*: the prover knows the witness
- *zero knowledge*: the verifier doesn't learn anything

Proving knowledge of isogenies

A survey (2023/671):

- $\mathcal{R}_{\mathsf{isog}} = \{((E_0, E_1), \phi) \mid \phi \text{ is an isogeny}\}$
- $\mathcal{R}_{\mathsf{deg}} = \{((E_0, E_1, d), \phi) \mid \phi \text{ is an isogeny of degree } d\}$
- $\mathcal{R}_{\mathsf{CSIDH}} = \{((E_0, E_1), \phi) \mid \phi \text{ is a CSIDH isogeny}\}$

Approaches

- general purpose:
  - graph isomorphism (CSI-FiSh)
  - modular polynomials (SSCYCT)
  - *this work!!*
- isogeny heavy:
  - SQIsign
  - ...

Isogeny Heavy - SQIsign

- starts from graph isomorphism
- plugs in *Deuring correspondence* (KLPT - ideal to isogeny)
- reduced soundness error
- the coolest team (according to website)
- requires $End(E)$

General purpose* - CSI-FiSh

- graph isomorphism
- group action properties
- high soundness error (repetitions / larger PK)

General purpose - Modular Polynomials

- encode an isogeny as path of j-invariants
- *low degree* algebraic relations (modular polynomials)
- prove the algebraic relations
- only works for *smooth degree isogenies*

Open questions

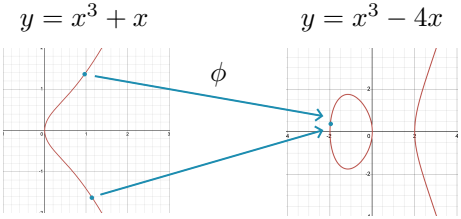
- a *generic proof of knowledge* for $\mathcal{R}_{\text{isog}}$
  - no knowledge of $End(E)$
  - no bound / leakage on the degree
- lower soundness error for csidh
- proving knowledge under standard assumptions

Outline

Isogenies

$$\phi : (x, y) \to \left(\frac{x^2 + 1}{x}, \frac{x^2 - 1}{x^2} y \right)$$

- maps between elliptic curves
- complexity grows with the *degree*
- also works in *higher dimensions*



Level 2 theta structure

- a map $A \to \mathbb{P}^{2^g-1}$ such that addition by $A[2]$ is nice
- invariant *theta null point* $\theta_A(0_A)$
- works on $A/\pm 1$
- nice compatibility with isogenies

Theta isogenies

- $A[2]$ has a *canonical decomposition* as $K_1 + K_2$
- this decomposition can be read from $\theta_A(0_A)$
- special 2-isogeny $\phi : (A, \theta_A) \to (B, \theta_B)$ with kernel K_1
- θ_B defines a 2-isogeny which is a *good extension* of ϕ

Theta isogenies

Define:

$$\mathcal{S} : \mathbb{P}^{2^g-1} \to \mathbb{P}^{2^g-1} : (x_0 : \cdots : x_{2^g}) \to (x_0^2 : \cdots : x_{2^g}^2)$$

$$\mathcal{H} : \mathbb{P}^{2^g-1} \to \mathbb{P}^{2^g-1} : (x_0 : \cdots : x_{2^g}) \to H_g(x_0 : \cdots : x_{2^g})^T$$

where

$$H_g = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}^{\bigotimes g}$$

It holds:

$$\mathcal{S} \circ \mathcal{H}(\theta^B(0_B)) = \mathcal{H} \circ \mathcal{S}(\theta^A(0_A))$$

Radical isogenies

- $\theta_A(0_A)$ is given, $\mathcal{H}$ invertible
- $\mathcal{S}$ gives $2^g - 1$ possible choices
- $g(g+1)/2$ are valid and give *good extensions*
- radical isogenies $\rightsquigarrow$ send bits to (valid) square root choices (2024/1732)

Radical isogenies

- dimension 1 and 2: $2^g - 1 = g(g+1)/2$ (any choice is valid)
- dimension 3: 7 possibilities, 6 valid choices
- explicit condition from Riemann relations
- dimension 4: open (probably too expensive)

Embedding

- isogeny computation is exponential in the degree
- can *embed isogenies* in higher dimensions
- d_1 and d_2 isogeny $\rightsquigarrow$ $d_1 + d_2$ isogeny in dimension $2g$
- $\phi : A \to B$ can be embedded in $\Phi : A^4 \times B^4 \to A^4 \times B^4$

Outline

Path to constraints

- start from a chain

$$A_0 \xrightarrow{\phi_0} A_1 \xrightarrow{\phi_1} \cdots \xrightarrow{\phi_{n-1}} A_n$$

- each step gives an equation

$$\mathcal{S} \circ \mathcal{H}(\theta^{A_{i+1}}(0_{A_{i+1}})) = \mathcal{H} \circ \mathcal{S}(\theta^{A_i}(0_{A_i}))$$

Path to constraints

Let

$$X^j = (X_0^g : \cdots : X_{2^g-1}^j) = \theta^{A_j}(0_{A_j})$$

Then:

$$\begin{cases} X^0 = \theta^{A_0}(0_{A_0}), \\ \mathcal{S} \circ \mathcal{H}(X^{j+1}) = \mathcal{H} \circ \mathcal{S}(X^j), \quad 0 \leq j \leq n-1, \\ X^n = \theta^{A_n}(0_{A_n}) \end{cases}$$

Path to constraints

- concretely: a transcript needs the X^j
- theta isogeny algorithms *already compute null points*
- normalize to ensure the equation holds affinely
- *gluing and splitting* are included

Constraints to path

- theta null points encode isogenies
- generally X^j are (almost) enough to evaluate ϕ
- can extract *an honest transcript*

Soundness

- (knowledge) soundness requires that all sets of X^j are isogenies
- *guaranteed* in dimension 1 and 2
- extra equations from dimension 3
- are those really needed?

Pseudo Isogeny Path Problem

- *pseudo-isogeny path problem*: given (A_0, θ^{A_0}), (A_n, θ^{A_n}) find any solution $(X^1, ..., X^{n-1})$ to the previous system
- *non-isogeny path problem*: given (A_0, θ^{A_0}), (A_n, θ^{A_n}) find a solution $(X^1, ..., X^{n-1})$ non corresponding to an isogeny
- isogeny path + non-isogeny path = pseudo-isogeny path

Cryptanalysis

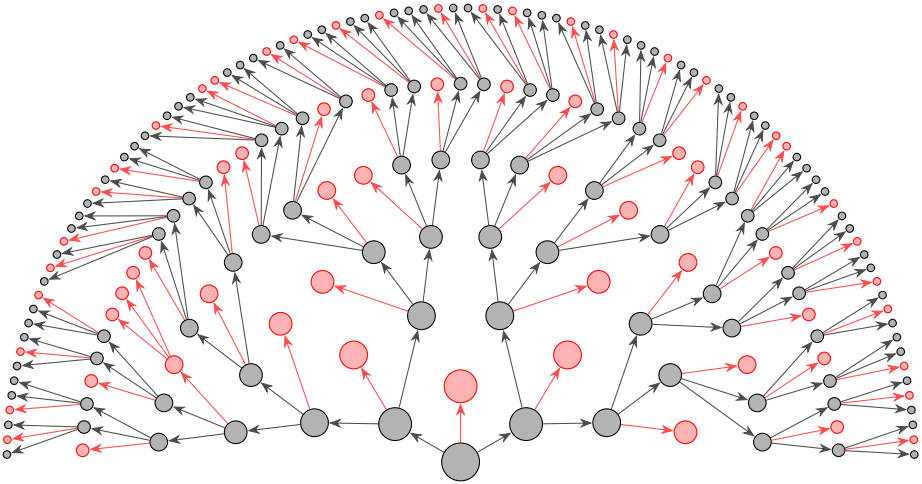
$$X^1 = \frac{1}{2^g}\mathcal{H} \circ \mathcal{T}_s \circ \mathcal{H} \circ \mathcal{S}(X^0)$$

- $\mathcal{T}_s$ extracts square roots with sign s
- for random X_i, $\mathcal{T}_s \circ \mathcal{H} \circ \mathcal{S}(X^i)$ will not be rational
- most non-isogeny paths *end immediately*

Cryptanalysis

- abelian varieties has codimension ≥ 1 (hard to come back)
- non-isogeny path $\rightsquigarrow$ abelian varieties of *unknown endomorphism ring*?
- best apprach seems still bruteforce
- are we missing something?

Cryptanalysis



Proving knowledge of any isogeny

- under pseudo-isogeny path problem proving 2-isogenies is equivalent to proving a transcript of X^j
- any isogeny can be embedded in a 2-isogeny in dimension $4g$
- we can now *prove any isogeny*

Outline

General structure

We want to prove knowledge of

$$\begin{cases} X^0 = \theta^{A_0}(0_{A_0}), \\ \mathcal{S} \circ \mathcal{H}(X^{j+1}) = \mathcal{H} \circ \mathcal{S}(X^j), \quad 0 \leq j \leq n-1, \\ X^n = \theta^{A_n}(0_{A_n}) \end{cases}$$

- easy to turn into R1CS: $(M_1 z) \circ (M_1 z) = M_2 z$
- M_1, M_2 are shifts of $\mathcal{H}$
- try different proof systems

Hash based proof systems

- BaseFold
- rely on (random oracle $+$) *proximity gap conjecture*
- not optimized for *small* witnesses
- field agnostic

Lattice based proof systems

- based on common lattice assumptions (LWE, SIS)
- tailored to our instance
- much smaller proof and faster proving
- recursive proofs

Some challenges

- simulate $\mathbb{F}_{p^2}$ over $\mathbb{F}_p$
- proving small statements
- field agnostic proof systems
- work affinely $\rightsquigarrow$ less constraints (not very relevant)

Outline

Theoretical results

$$\mathcal{R}_{\text{isog}} = \{((E_0, E_1), \phi) \mid \phi \text{ is an isogeny}\}$$

- can prove ϕ of any (secret) degree
- no requirement on $End(E)$
- standard assumptions*

Theoretical results

$$\mathcal{R}_{\mathsf{CSIDH}} = \{((E_0, E_1), \phi) \mid \phi \text{ is a CSIDH isogeny}\}$$

- avoid repetitions
- comparable timings (but larger proof sizes)
- field size issues

Additional relations

- *point evaluation*: similar to isogeny evaluation
- *proving degrees*: point evaluation + pairings
- can prove $\mathcal{R}_{\mathsf{deg}}$, $\mathcal{R}_{\mathsf{MSIDH}}$, ...

Dimension 2 CGL

- CGL hash function: turn message bits into isogenies
- broken by KLPT $\rightsquigarrow$ need trusted setup
- 2-dimensional version broken by KLPT^2
- *2D trusted setup*

Other HD isogeny constructions

- MIKE
- natively 2-dimensional constructions (Shimura CGA?)
- something else?

BaseFold proof system

		$g = 1, k = 256$	$g = 2, k = 108$	$g = 4, k = 498$
$\mathbb{F}_p$	Prover time	1288 ms	966 ms	10961 ms
	Proof size	4826 KiB	6320 KiB	7833 KiB
	Verif. time	74 ms	45 ms	228 ms
$\mathbb{F}_{p^2}$	Prover time	1846ms	1074ms	
	Proof size	3344KiB	3767KiB	n/a
	Verif. time	286ms	179ms	

Lattice based proof system

d	Prover time (s)	Proof size (KiB)	Verification time (s)
0	5.2	1147	105.5
6	6.0	1156	37.4
8	6.5	1200	10.8
11	8.6	1648	3.8

Thank you for your attention.
Questions?