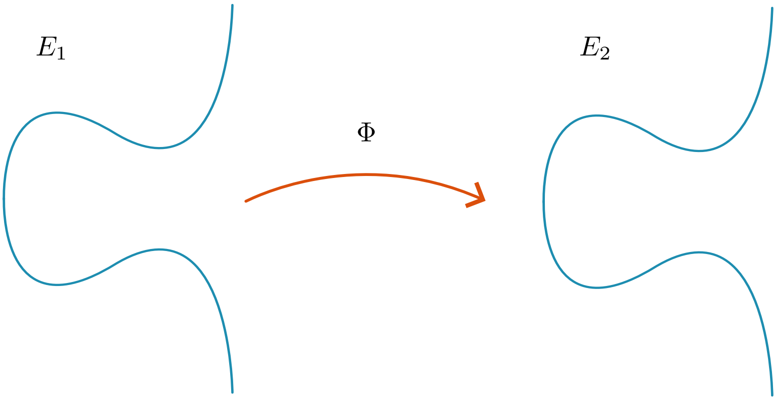
PRISM - Signatures from Large Prime Degree Isogenies

PKC 2025 - Røros

R. Invernizzi - joint work with A. Basso, G. Borin, W. Castryck, M. Corte-Real Santos, A. Leroux, L. Maino, F. Vercauteren and B. Wesolowski

May 14

What is an isogeny



What is an isogeny

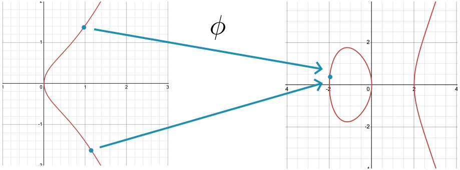
$$\phi : (x, y) \to \left(\frac{x^2 + 1}{x}, \frac{x^2 - 1}{x^2} y \right)$$

- map between elliptic curves
- respect *group structure*:

$$\phi(P + Q) = \phi(P) + \phi(Q)$$

- kernel: $\{P \mid \phi(P) = 0\}$
- size of the kernel $=$ *degree* of the isogeny

$y = x^3 + x$ $\qquad$ $y = x^3 - 4x$



KU LEUVEN

Important properties

- an isogeny $\phi : E \rightarrow E$ is called *endomorphism*
- an example: $[N](P) = P + ... + P$
- can *add* endomorphisms: $(\phi + \psi)(P) = \phi(P) + \psi(P)$
- can *multiply* endomorphisms: $(\phi \cdot \psi)(P) = \phi(\psi(P))$
- endomorphisms form a *ring* $(End(E), +, \cdot)$
- $[1], [2], ... \in End(E) \quad \rightarrow \quad \mathbb{Z} \subset End(E)$

Isogeny representations

Rational maps:

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

- derived explicitly
- easy to evaluate
- complexity polynomial in the degree

Interpolation data:

$$(\phi(P), \phi(Q))$$

- requires evaluating the isogeny first
- more complex evaluation procedure
- independent of the degree

The isogeny world

Hard:

- compute $End(E)$ for random E
- compute $\phi : E_0 \to E_1$

Easy:

- compute $End(E)$ for some special curves
- given $End(E_0)$ and $\phi : E_0 \to E_1$, compute $End(E_1)$
- given $End(E_0), End(E_1)$ compute $\phi : E_0 \to E_1$

Secret key generation

- start from a *special curve* E_0 s.t. $End(E_0)$ is known
- compute a *random isogeny* φ_{sk} from E_0 to the public key E_{pk}
- with $End(E_0)$ and $\varphi_{sk} : E_0 \to E_{pk}$ compute $End(E_{pk})$ (secret key)
- done in most isogeny protocols

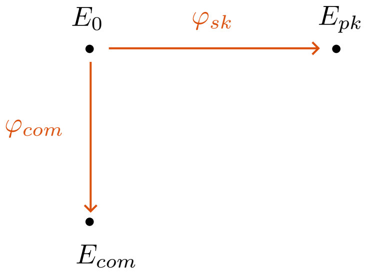
Isogeny-based signatures

Main idea of *SQISign*: prove knowledge of $End(E_{pk})$ with a *sigma protocol*

$$E_0 \xrightarrow{\varphi_{sk}} E_{pk}$$

Isogeny-based signatures

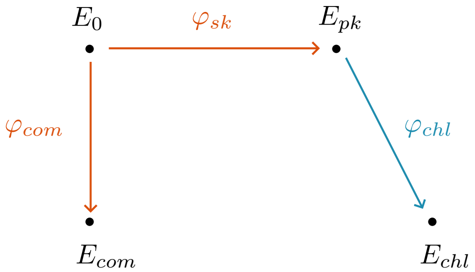
Main idea of *SQISign*: prove knowledge of $End(E_{pk})$ with a *sigma protocol*



KU LEUVEN

Isogeny-based signatures

Main idea of *SQISign*: prove knowledge of $End(E_{pk})$ with a *sigma protocol*



Isogeny-based signatures

Main idea of *SQISign*: prove knowledge of $End(E_{pk})$ with a *sigma protocol*

$$
\begin{tikzcd}
E_0 \arrow[r, "\varphi_{sk}"] \arrow[d, "\varphi_{com}"'] & E_{pk} \arrow[d, "\varphi_{chl}"] \\
E_{com} \arrow[r, "\varphi_{rsp}"'] & E_{chl}
\end{tikzcd}
$$

Isogeny signatures

Pros:

- very short signatures

Cons:

- complicated signing procedure
- not very fast

A new hard problem

Computing an isogeny of *large prime degree* from a random curve is hard.

- becomes easy if we know $End(E)$
- large and prime is needed
- *small:* computing a 2-isogeny $\rightarrow$ fast
- *smooth:* computing a 2^{256}-isogeny $= 256$ 2-isogenies $\rightarrow$ fast

A new hard problem

Computing an isogeny of *large prime degree* from a random curve is hard.

- well known problem in the math community
- best algorithms are *exponential in the degree*
- security proofs (e.g. SQISign) could become easier
- other schemes (e.g. PEGASIS) could become much faster

A simpler signature scheme

- given a message m
- hash m into a *prime* $q = H(m) > 2^a$
- a determines our *security / efficiency*
- using $End(E_{pk})$, compute an isogeny σ from E_{pk} of degree q
- the signature is an *interpolation* of σ: $(\sigma(P), \sigma(Q))$
- *verification:* check the validity of the representation

A simpler signature scheme

$$\begin{tikzcd} E_0 \arrow[r, "\varphi_{sk}"] \arrow[d, "\varphi_{com}"'] & E_{pk} \arrow[d, "\varphi_{chl}"] \\ E_{com} \arrow[r, "\varphi_{rsp}"'] & E_{chl} \end{tikzcd}$$

$$\begin{tikzcd} E_0 \arrow[r, "\varphi_{sk}"] & E_{pk} \arrow[d, "\sigma"] \\ & E \end{tikzcd}$$

Concrete security

- key recovery: $O(p^{1/2}) \to p \approx 2\lambda$
- forgery: $O(q^2) \to a \approx \lambda/2$
- hash collisions: $O(q^{1/2}) \to a \approx 2\lambda$
- observing past signatures *does not help*:
  - can already compute high (smooth) degree isogenies from E_{pk}
  - being *prime degree* is most likely a disadvantage (SQISign oracles)

Signature Sizes

Protocol	**This Work**	SQIsign	SQIsign2D-East	SQIsign2D-West	SQIPrime
Sig. size (bits)	12λ	$\approx 11\lambda$	12λ	9λ	19λ

- signature size: 12λ bits (1 point $+$ 1 x coordinate)
- can do 11λ bits but slower

Performance

SQIsign2D-West	`Key Gen`	77.4
	`Sign`	285.7
	`Verify`	11.9
This work	`Key Gen`	78.2
	`Sign`	157.6
	`Verify`	16.9

- implemented on the SQISign2D-West codebase
- signing $1.8\times$ faster, verification $1.4\times$ slower

Conclusions

Computing an isogeny of *large prime degree* from a random curve is hard.

- simple isogeny-based signatures
- performance / size on par with SQIsign, but more flexible
- advanced protocols based on PRISM

Thank you for your attention.