PRISM: PRime degree ISogeny Mechanism

Isogeny Club #5.5

Riccardo Invernizzi

COSIC - KU Leuven

December 3rd

Outline

KU LEUVEN

Credits

- Andrea Basso
- Giacomo Borin
- Wouter Castryck
- Maria Corte-Real Santos
- Riccardo Invernizzi (Me)
- Luciano Maino
- Antonin Leroux
- Frederik Vercauteren
- Benjamin Wesolowski

High degree isogenies

Main idea: computing high (prime) degree isogenies is

- **hard** without knowledge of the endomorphism ring
  - best way: work rationally
  - exponential in the smallest factor
- **easy** with it
  - look at the corresponding ideals

This is exactly what we want in crypto.

Identification Protocol

- Verification key: random curve E_{vk}
- secret key: I_{sk}, $\mathrm{End}(E_{vk})$
- query: a big prime number q
- response: a degree $q(2^a - q)$ isogeny starting from E_{vk}
- verification: evaluate the isogeny and check the degree

KU LEUVEN

Signature Scheme

- Public key: random curve E_{vk}
- secret key: I_{sk}, $\mathrm{End}(E_{vk})$
- *query: hash the message m into a prime number q*
- signature: a degree $q(2^a - q)$ isogeny starting from E_{vk}
- verification: evaluate the isogeny and check the degree

Outline

Secret Key

- Start from E_0
- heuristic key-generation (QFESTA)
  - endomorphism of degree $n(2^a - n)$
  - isogeny factorization
- rigorous key-generation (SQISign2D-West)
  - uniformly random ideal
  - `IdealToIsogeny`
- any other idea to obtain a known EndRing curve without consuming 2-torsion.

Response

- Generate an $\mathcal{O}_0$-ideal I_{chall} of norm $q(2^a - q)$
  - `RandomFixedNormIdeal`
- compute $I_\sigma = [I_{sk}]_* I_{chall}$
- translate $I_\sigma I_{sk}$ into the corresp. isogeny
  - `IdealToIsogeny`
- recover σ
- given $\langle P_{vk}, Q_{vk} \rangle = E_{vk}[2^a]$ compute $\sigma(P_{vk}), \sigma(Q_{vk})$
- send $P_{sig} = [q^{-1}]\sigma(P_{vk}), Q_{sig} = [q^{-1}]\sigma(Q_{vk})$

Ideal to Isogeny

Main idea:

- given an ideal I find ideals I_1, I_2 and integers u, v s.t.

$$d_1 u + d_2 v = 2^a$$

- use I_1, I_2 to build a Kani square and recover I.

We can use it as a black box. But:

- we need $p = f2^e - 1$ with f small cofactor
- we can reuse SQISign2D-West parameters
- better keep things odd

Hash to Prime

- Take any cryptographic hash function $H_a : \{0,1\}^* \rightarrow (2^{a-1}, 2^a)$
- compute $H_a(E_{vk}||\texttt{msg}||\texttt{counter})$
- increment counter until a prime is found

Remarks:

- expected hits: $1/a$
- each hash computation requires a primality testing $(O(a^2))$
- `counter` can be attached to the signature

Verification

- Recover q from the message (and counter)
- compute the isogeny with kernel

$$\langle (P_{vk}, P_{sig}), (Q_{vk}, Q_{sig}) \rangle$$

- obtain an isogeny Φ that embeds (factors of) σ
- compute $(P', -) = \Phi((P_{vk}, 0))$ and $(Q', -) = \Phi((Q_{vk}, 0))$
- verify that

$$e_{2^a}(P', Q') = e_{2^a}(P_{vk}, Q_{vk})^n$$

  with $n \in \{q, 2^a - q\}$

Outline

High degree isogeny oracles

- Many schemes (SQISign-HD...) rely on isogeny oracles
- hard to argue security without
- there is no known way to compute them
- claimed to leak no information
  - smooth isogenies cover the isogeny graph
  - for every given rough isogeny, there is an equivalent smooth one
- *security by common belief*

Key Recovery

- Recovering the secret key means computing $\mathrm{End}(E_{vk})$
- E_{vk} is a random curve
- best known algorithms: Delfs-Galbraith & friends, $\widetilde{O}(p^{1/2})$
- requires $p \approx 2\lambda$

KU LEUVEN

Forgery and impersonation

- *Main problem*: given a prime q compute a degree $q(2^a - q)$ isogeny from a given curve E_{vk}
- best way: *Velu-sqrt*
  - computes the isogeny in $O(q^{1/2})$
  - but: requires available q-torsion in $O(q)$
  - total complexity $O(q^{3/2})$
- other ways (kernel polynomial, ...) are even slower
- implies $q > 2^{2\lambda/3}$ so $a > 2\lambda/3$ (already ok)
- means $a \approx 90$ bits for Level 1

Reusing a signature

- Can always reuse part of a signature (e.g. factors of $2^a - q$)
  - chosing $q > 2^{a-1}$ prime prevents it
- does looking at signatures help? *Hopefully not*
  - can already compute high degree isogenies from E_{vk}
  - being *prime degree* is most likely a disadvantage

Hashing

- Another attack: *hash collisions*
- we hash into primes but the hash is repeated
  - the image space for the hash is only $2^\lambda/\lambda$
  - each message needs λ hash ($+$ primality tests)
- $a \approx 2\lambda$ is enough
- can do slightly better (e.g. $a = 219$ for $\lambda = 128$)
- by far the biggest constraint on a

Outline

The SQIsign Family

A different paradigm form the SQIsign family

- no Fiat-Shamir
- *hash & sign*
- more flexible (Erebor and Durian...)
- comparable size and speed

Other prime-degree isogeny signatures

Not really *the first* prime-degree isogeny based signature

- DeuringVRF
- SQIPrime

DeuringVRF idea

- Pick d prime
- P, Q basis of $E[d]$
- $\iota \in End(E)$ s.t. $\iota(P) = Q$
- I_P the ideal corresponding to $\langle P \rangle$
- ideal of $\langle P + kQ \rangle$ is easy to compute
- evaluate with the `IdealToIsogeny` machinery

DeuringVRF-based protocols

Then in practice

- in DeuringVRF, k is the input
  - the random output is the codomain of the isogeny
- in SQIPrime, k is the challenge
  - evaluation by completing the diagram
  - verifier cannot compute the challenge

Comparison

- setting: fixed prime (VRF) vs. different primes (PRISM)
- consequence: $d|(p+1)$ with d big prime
- bigger cofactor on p impacts `IdealToIsogeny`
- hash & sign and verification are similar
- overall PRISM is faster than VRF in all steps
- SQIPrime: VRF setting but different protocol

Outline

Public key size

- E_{vk}: 4λ bits
- Deterministic basis of $E_{vk}[2^a]$
  - included (fast) or computed (compact)
  - tradeoff: hints (SQISign2D-West)

Signature Sizes

Standard method:

- codomain E_{sig}: 4λ bits
- kernel (P_{sig}, Q_{sig}): 16λ bits (8λ per point)
- deterministic basis: 2 coefficients per point
- $a \approx 2\lambda$, so 2λ bits per coefficient
- 12λ bits in total
- det. basis + point compression: 11λ bits in total

Signature Sizes

Instead:

- send P_{sig}: 8λ bits
- recover E_{sig} (one field inversion)
- send $x(Q_{sig})$ (4λ bits) + one bit (and one square root)
- no det. basis or pairings
- 12λ bits but much faster

KU LEUVEN

Signature Sizes - Comparison

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

- SQISign2D-East: can go to 10λ at the cost of more 2D steps
- SQISign2D-West: can go to 8λ at the cost of pairings
- SQIPrime: can go to 12λ using 4D isogenies

KU LEUVEN

Communication cost

- Only require $a \approx \lambda + \log \lambda$
- curve: 4λ bits
- points: $4a \approx 4\lambda + 4\log\lambda$ (+ det. basis)
- compressed points: $3a \approx 3\lambda + 3\log\lambda$ (+ pairings)
- total: $8\lambda + 4\log\lambda$ / $7\lambda + 3\log\lambda$

Performance

		Type of isogeny				
Protocol		2	3	5	$(2,2)$	$(2,2,2,2)$
This Work	KeyGen	-	-	-	496	-
	Sign	-	-	-	496	-
	Verify	-	-	-	219	-
SQIsignHD	KeyGen	378	234	-	-	-
	Sign	252	312	-	-	-
	Verify	-	78	-	-	142
SQIsign2D-West	KeyGen	-	-	-	496	-
	Sign	(248)	-	-	992	-
	Verify	(248)	-	-	(126)	-
SQIsign2D-West (Heuristic)	KeyGen	-	-	-	496	-
	Sign	(122)	-	-	624	-
	Verify	(122)	-	-	(126)	-
SQIsign2D-East	KeyGen	-	-	-	253	-
	Sign	127	(2)	(1)	641	-
	Verify	127	(2)	(1)	129	-

Performance - Comparison

- Key generation is the same
- no commitment / challenge isogenies
- hash to prime (signing) but no det. basis (verification)
- PRISM-id verification twice as fast:
  - only need a 2D isogenies
  - $a = 135$ for NIST Level 1

Performance - Comparison

SQIsign2D-West	KeyGen	77.4
	Sign	285.7
	Verify	11.9
This work	KeyGen	78.2
	Sign	157.6
	Verify	16.9

- Implemented on the SQISign2D-West codebase
- signing $1.8\times$ faster, verification $1.4\times$ slower
- currently: unoptimized LLL takes 40% of the time
- other schemes: not implemented but we can estimate

Conclusions

- Totally different idea from traditional schemes
- Competitive performance / size
- More flexible

Thank you for your attention.