SCALLOP: a somewhat scalable effective group action from isogenies

Luca De Feo Tako Boris Fouotsa Péter Kutas

Antonin Leroux **Simon-Philipp Merz** Lorenz Panny

Benjamin Wesolowski

February 2024

Isogeny Club

Cryptographic group actions

Definition

A group action of a group G on a set X is a function

$$\star : G \times X \to X$$

- $e \star x = x$
- $(gh) \star x = g \star (h \star x)$

Cryptographic group actions

Definition

A group action of a group G on a set X is a function

$$\star : G \times X \to X$$

- $e \star x = x$
- $(gh) \star x = g \star (h \star x)$

- Vectorization prob.: given $x, y \in X$, find $g \in G$ s.t. $y = g \star x$
- Parallelization prob.: given $x, g \star x, h \star x$, find $(gh) \star x$

Cryptographic group actions

Definition

A group action of a group G on a set X is a function

$$\star : G \times X \to X$$

- $e \star x = x$
- $(gh) \star x = g \star (h \star x)$

- Vectorization prob.: given $x, y \in X$, find $g \in G$ s.t. $y = g \star x$
- Parallelization prob.: given $x, g \star x, h \star x$, find $(gh) \star x$

- Typically group action-based cryptography has focussed on group actions that are both free and transitive

EGA: effective group action

Definition (EGA)

A group action $(G, X, \star)$ is effective, if there exist efficient (PPT) algorithms for

- membership testing, equality testing, sampling and computing the operation and inversion in G
- membership testing and unique representation in X
- computing $g \star x$ for any $g \in G$ and $x \in X$.

EGA: effective group action

Definition (EGA)

A group action $(G, X, \star)$ is effective, if there exist efficient (PPT) algorithms for

- membership testing, equality testing, sampling and computing the operation and inversion in G
- membership testing and unique representation in X
- computing $g \star x$ for any $g \in G$ and $x \in X$.

CSIDH is not an EGA!

For arbitrary $g \in G$ and $x \in X$, computing $g \star x$ is not efficient!

CSIDH: a restricted effective group action

- CSIDH is a *restricted* effective group action (REGA), i.e. evaluate group action only on certain (representations of) elements in G

CSIDH: a restricted effective group action

- CSIDH is a ***restricted*** effective group action (REGA), i.e. evaluate group action only on certain (representations of) elements in G

More precisely:

- Fix list of elements $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ spanning G such that $\mathfrak{l}_i \star E$ can be efficiently evaluated for every $E \in X$
- Can evaluate $\prod_i \mathfrak{l}_i^{e_i} \star E$ for $E \in X$ efficiently as long as exponents $(e_1, \ldots, e_n) \in \mathbb{Z}^n$ are sufficiently small, i.e. e_i sampled from $[-B, B]$ for some bound B in CSIDH

CSIDH: a restricted effective group action

- CSIDH is a *restricted* effective group action (REGA), i.e. evaluate group action only on certain (representations of) elements in G

More precisely:

- Fix list of elements $\mathfrak{l}_1, \ldots, \mathfrak{l}_n$ spanning G such that $\mathfrak{l}_i \star E$ can be efficiently evaluated for every $E \in X$
- Can evaluate $\prod_i \mathfrak{l}_i^{e_i} \star E$ for $E \in X$ efficiently as long as exponents $(e_1, \ldots, e_n) \in \mathbb{Z}^n$ are sufficiently small, i.e. e_i sampled from $[-B, B]$ for some bound B in CSIDH

So what?

EGA vs REGA: Identification protocols and Fiat-Shamir signatures

Let $(G, X, \star)$ be an EGA. Zero-knowledge proof of knowledge of secret $s \in G$ corresponding to public key $(E_0, E_1 := s \star E_0) \in X^2$:

- Prover commits to $E_c := r \star E_0$ for random $r \in G$

EGA vs REGA: Identification protocols and Fiat-Shamir signatures

Let $(G, X, \star)$ be an EGA. Zero-knowledge proof of knowledge of secret $s \in G$ corresponding to public key $(E_0, E_1 := s \star E_0) \in X^2$:

- Prover commits to $E_c := r \star E_0$ for random $r \in G$
- Challenger sends bit b to prover who reveals $s^b r^{-1}$

EGA vs REGA: Identification protocols and Fiat-Shamir signatures

Let $(G, X, \star)$ be an EGA. Zero-knowledge proof of knowledge of secret $s \in G$ corresponding to public key $(E_0, E_1 := s \star E_0) \in X^2$:

- Prover commits to $E_c := r \star E_0$ for random $r \in G$
- Challenger sends bit b to prover who reveals $s^b r^{-1}$
- Challenger checks whether E_b is equal to $s^b r^{-1} \star E_c$

EGA vs REGA: Identification protocols and Fiat-Shamir signatures

Let $(G, X, \star)$ be an EGA. Zero-knowledge proof of knowledge of secret $s \in G$ corresponding to public key $(E_0, E_1 := s \star E_0) \in X^2$:

- Prover commits to $E_c := r \star E_0$ for random $r \in G$
- Challenger sends bit b to prover who reveals $s^b r^{-1}$
- Challenger checks whether E_b is equal to $s^b r^{-1} \star E_c$

Can turn protocol into (non-interactive) signature scheme with Fiat-Shamir transform.

EGA vs REGA: Identification protocols and Fiat-Shamir signatures

Let $(G, X, \star)$ be an EGA. Zero-knowledge proof of knowledge of secret $s \in G$ corresponding to public key $(E_0, E_1 := s \star E_0) \in X^2$:

- Prover commits to $E_c := r \star E_0$ for random $r \in G$
- Challenger sends bit b to prover who reveals $s^b r^{-1}$
- Challenger checks whether E_b is equal to $s^b r^{-1} \star E_c$

Can turn protocol into (non-interactive) signature scheme with Fiat-Shamir transform.

- Zero-knowledge proof breaks for REGA, $s^b r^{-1}$ can leak information about s
- Fix: rejection sampling (see SeaSign) $\Rightarrow$ considerable increase in parameters, much less efficient

General strategy: REGA to EGA

For simplicity, assume acting group $G = \langle \mathfrak{l}_1 \rangle$ is cyclic.

General strategy: REGA to EGA

For simplicity, assume acting group $G = \langle \mathfrak{l}_1 \rangle$ is cyclic.

Precomputation done once:

- Compute cardinality of acting group $|G|$
- Compute lattice of relations $\mathcal{L}$ of $\mathfrak{l}_i$, i.e. lattice spanned by vectors $(e_1, \ldots, e_n) \in \mathbb{Z}^n$ such that $\prod_i \mathfrak{l}_i^{e_i}$ acts trivially on X
- Compute reduced basis of $\mathcal{L}$ which allows to solve CVP instances efficiently

General strategy: REGA to EGA

For simplicity, assume acting group $G = \langle \mathfrak{l}_1 \rangle$ is cyclic.

Precomputation done once:

- Compute cardinality of acting group $|G|$
- Compute lattice of relations $\mathcal{L}$ of $\mathfrak{l}_i$, i.e. lattice spanned by vectors $(e_1, \ldots, e_n) \in \mathbb{Z}^n$ such that $\prod_i \mathfrak{l}_i^{e_i}$ acts trivially on X
- Compute reduced basis of $\mathcal{L}$ which allows to solve CVP instances efficiently

Online phase to evaluate $\mathfrak{l}_1^e \star E$ (for all $e \in \mathbb{Z}$):

- Solve (approximate) CVP of $(e, 0, \ldots, 0)$ in $\mathcal{L}$ to find decomposition $\mathfrak{l}_1^e = \prod_i \mathfrak{l}_i^{e_i}$ with small exponents e_i
- Evaluate the restricted group action $\prod_i \mathfrak{l}_i^{e_i} \star E$

General strategy: REGA to EGA

For simplicity, assume acting group $G = \langle \mathfrak{l}_1 \rangle$ is cyclic.

Precomputation done once:

- Compute cardinality of acting group $|G|$
- Compute lattice of relations $\mathcal{L}$ of $\mathfrak{l}_i$, i.e. lattice spanned by vectors $(e_1, \ldots, e_n) \in \mathbb{Z}^n$ such that $\prod_i \mathfrak{l}_i^{e_i}$ acts trivially on X
- Compute reduced basis of $\mathcal{L}$ which allows to solve CVP instances efficiently

Online phase to evaluate $\mathfrak{l}_1^e \star E$ (for all $e \in \mathbb{Z}$):

- Solve (approximate) CVP of $(e, 0, \ldots, 0)$ in $\mathcal{L}$ to find decomposition $\mathfrak{l}_1^e = \prod_i \mathfrak{l}_i^{e_i}$ with small exponents e_i
- Evaluate the restricted group action $\prod_i \mathfrak{l}_i^{e_i} \star E$

Caution

Depending on the group G, the precomputation might be computationally infeasible!

CSI-FiSh signature scheme [BKV19]

- Based on group action of CSIDH-512
- Precompute lattice of relations $\mathcal{L}$ for the generators used in CSIDH-512 using an index-calculus approach
- CSI-FiSh required a world-record class group computation to obtain the lattice for the smallest CSIDH parameters

CSI-FiSh signature scheme [BKV19]

- Based on group action of CSIDH-512
- Precompute lattice of relations $\mathcal{L}$ for the generators used in CSIDH-512 using an index-calculus approach
- CSI-FiSh required a world-record class group computation to obtain the lattice for the smallest CSIDH parameters

Caution

Computing the structure of the acting group for larger CSIDH parameters is infeasible with currently known algorithms.

Idea

Motivation

Introduce group action that solves the scaling issue of CSI-FiSh

(to some extent..)

Cryptographic group actions $(G, X, \star)$ for which structure of G can be computed more easily?

Idea

Motivation

Introduce group action that solves the scaling issue of CSI-FiSh

(to some extent..)

Cryptographic group actions $(G, X, \star)$ for which structure of G can be computed more easily?

Idea

Can compute class number $|\mathrm{Cl}(\mathfrak{O})|$ for $\mathfrak{O}$ of the form $\mathbb{Z} + f\mathfrak{O}_0$ from class number $|\mathrm{Cl}(\mathfrak{O}_0)|$ and factorization of f.

Let $f \in \mathbb{Z}$, let $\mathfrak{O}_0$ be a quadratic order of class number h_0 and discriminant d_0 and let $u_0 := |\mathfrak{O}^\times|/2$.
For $\mathfrak{O}$ of the form $\mathbb{Z} + f\mathfrak{O}_0$ we have

$$|\mathrm{Cl}(\mathfrak{O})| = \left(f - \left(\frac{d_0}{f}\right)\right)\frac{h_0}{u_0}.$$

Oriented elliptic curves

Let $\mathfrak{O}$ be an imaginary quadratic order, e.g. $\mathbb{Z}[i]$, $\mathbb{Z}[\sqrt{-p}]$, in an imaginary quadratic field K.

Definition

For any elliptic curve E, a K-orientation is a ring homomorphism $\iota : K \to \text{End}(E) \otimes \mathbb{Q}$. A K-orientation induces a primitive $\mathfrak{O}$-orientation if $\iota(\mathfrak{O}) = \text{End}(E) \cap \iota(K)$. In that case, the pair (E, ι) is called an $\mathfrak{O}$-oriented curve.

Oriented elliptic curves

Let $\mathfrak{O}$ be an imaginary quadratic order, e.g. $\mathbb{Z}[i]$, $\mathbb{Z}[\sqrt{-p}]$, in an imaginary quadratic field K.

Definition

For any elliptic curve E, a K-orientation is a ring homomorphism $\iota : K \to \mathrm{End}(E) \otimes \mathbb{Q}$. A K-orientation induces a primitive $\mathfrak{O}$-orientation if $\iota(\mathfrak{O}) = \mathrm{End}(E) \cap \iota(K)$. In that case, the pair (E, ι) is called an <u>$\mathfrak{O}$-oriented</u> curve.

- ι embeds $\mathfrak{O}$ into $\mathrm{End}(E)$ (and no superorder of $\mathfrak{O}$)

Oriented elliptic curves

Let $\mathfrak{O}$ be an imaginary quadratic order, e.g. $\mathbb{Z}[i]$, $\mathbb{Z}[\sqrt{-p}]$, in an imaginary quadratic field K.

Definition

For any elliptic curve E, a K-orientation is a ring homomorphism $\iota : K \to \mathrm{End}(E) \otimes \mathbb{Q}$. A K-orientation induces a primitive $\mathfrak{O}$-orientation if $\iota(\mathfrak{O}) = \mathrm{End}(E) \cap \iota(K)$. In that case, the pair (E, ι) is called an <u>$\mathfrak{O}$-oriented</u> curve.

- ι embeds $\mathfrak{O}$ into $\mathrm{End}(E)$ (and no superorder of $\mathfrak{O}$)
- We will represent the orientation by a kernel representation of an endomorphism corresponding to a generator of $\mathfrak{O}$

Group actions on oriented curves

- Let X be the set of primtively $\mathfrak{O}$-oriented curves (E, ι) up to isomorphism and Galois conjugacy

Group actions on oriented curves

- Let X be the set of primtively $\mathfrak{O}$-oriented curves (E, ι) up to isomorphism and Galois conjugacy
- Invertible ideals of $\mathfrak{O}$ act on X, principal ideals act trivially, i.e. get group action by class group $\mathrm{Cl}(\mathfrak{O})$

$$\mathrm{Cl}(\mathfrak{O}) \times X \to X$$

Group actions on oriented curves

- Let X be the set of primtively $\mathfrak{O}$-oriented curves (E, ι) up to isomorphism and Galois conjugacy
- Invertible ideals of $\mathfrak{O}$ act on X, principal ideals act trivially, i.e. get group action by class group $\mathrm{Cl}(\mathfrak{O})$

$$\mathrm{Cl}(\mathfrak{O}) \times X \to X$$

- Group action is free and transitive (see [Onu21])

Group actions on oriented curves

- Let X be the set of primtively $\mathfrak{O}$-oriented curves (E, ι) up to isomorphism and Galois conjugacy
- Invertible ideals of $\mathfrak{O}$ act on X, principal ideals act trivially, i.e. get group action by class group $\mathrm{Cl}(\mathfrak{O})$

$$\mathrm{Cl}(\mathfrak{O}) \times X \to X$$

- Group action is free and transitive (see [Onu21])
- Example: CSIDH, where $\mathfrak{O} = \mathbb{Z}[\sqrt{-p}]$ with orientations that send $\sqrt{-p}$ to Frobenius endomorphisms

Group actions on oriented curves cont.

- Computing group action using isogenies:
  - Let $\mathfrak{a} \subset \mathfrak{O}$ ideal, (E, ι_E) an elliptic curve with $\mathfrak{O}$-orientation
  - Define $E[\mathfrak{a}] = \bigcap_{\alpha \in \mathfrak{a}} \ker \iota_E(\alpha)$ and let

$$\varphi_{\mathfrak{a}}^E := E \to E_{\mathfrak{a}} := E/E[\mathfrak{a}] \quad \text{and} \quad \iota_{E_{\mathfrak{a}}}(x) = \frac{1}{n(\mathfrak{a})}\varphi_{\mathfrak{a}}^E \circ \iota(x) \circ \hat{\varphi}_{\mathfrak{a}}^E$$

  - $\mathfrak{a} \star (E, \iota_E) = (E_{\mathfrak{a}}, \iota_{E_{\mathfrak{a}}})$

Computing with oriented curves

How to represent and compute with different orientation effectively?

Computing with oriented curves

How to represent and compute with different orientation effectively?

CSIDH:

- Ideal $\mathfrak{l}_i \subset \mathbb{Z}[\sqrt{-p}]$ acts through an isogeny of degree $\ell_i = n(\mathfrak{l}_i)$ whose kernel is stabilized by the Frobenius endomorphism π corresponding to $\sqrt{-p}$
- To compute $\mathfrak{l}_i \star E$ it is sufficient to evaluate the Frobenius endomorphism π on $E[\ell_i]$ and determine its eigenspaces

Computing with oriented curves

How to represent and compute with different orientation effectively?

~~CSIDH~~ General:

- Ideal $\mathfrak{l}_i \subset$ ~~$\mathbb{Z}[\sqrt{-p}]$~~ $\mathfrak{O}$ acts through an isogeny of degree $\ell_i = n(\mathfrak{l}_i)$ whose kernel is stabilized by ~~the Frobenius endomorphism π corresponding to $\sqrt{-p}$~~ endomorphism ω corresponding to a generator of $\mathfrak{O}$
- To compute $\mathfrak{l}_i \star E$ it is sufficient to evaluate ~~the Frobenius endomorphism π~~ endomorphism ω on $E[\ell_i]$ and determine its eigenspaces
- Compute (kernel) representation of endomorphism corresponding to generator of $\mathfrak{O}$ under orientation

Wishlist

To compute the class group structure, we want:

- $|\mathrm{Cl}(\mathfrak{O}_0)|$
- $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_0$ such that factorisation of conductor f known
- $|\mathrm{Cl}(\mathfrak{O})|$ smooth enough to be able to compute the lattice of relations between ideal actions

Wishlist

To compute the class group structure, we want:

- $|\mathrm{Cl}(\mathfrak{O}_0)|$
- $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_0$ such that factorisation of conductor f known
- $|\mathrm{Cl}(\mathfrak{O})|$ smooth enough to be able to compute the lattice of relations between ideal actions

To represent and compute with oriented curves explicitly, we want:

- A generator α of $\mathfrak{O}$ of smooth norm $L_1^2 L_2$ to efficiently compute and represent corresponding endomorphisms
- A primitively $\mathfrak{O}$-oriented starting curve

SCALLOP: Precomputation

SCALable isogeny action based on Oriented supersingular curves with Prime conductor

- Take $\mathfrak{O}_0$ with $|\mathrm{Cl}(\mathfrak{O}_0)| = 1$, we take $\mathfrak{O}_0 = \mathbb{Z}[i]$

SCALLOP: Precomputation

SCALable isogeny action based on Oriented supersingular curves with Prime conductor

- Take $\mathfrak{O}_0$ with $|\mathsf{Cl}(\mathfrak{O}_0)| = 1$, we take $\mathfrak{O}_0 = \mathbb{Z}[i]$
- Generate candidates for $\mathfrak{O}$ with smooth generator until
  - conductor $f \approx 2^{2\lambda}$ is prime (avoids factoring f)
  - class number $|\mathsf{Cl}(\mathfrak{O})|$ is reasonably smooth

SCALLOP: Precomputation (contd.)

- Fix $\ell_1, \ldots, \ell_n$ to be the smallest n split primes in $\mathbb{Z}[i]$, e.g. $(5) = (2+i)(2-i)$, $(13) = (3+2i)(3-2i)$ etc.

SCALLOP: Precomputation (contd.)

- Fix $\ell_1, \ldots, \ell_n$ to be the smallest n split primes in $\mathbb{Z}[i]$,
  e.g. $(5) = (2+i)(2-i)$, $(13) = (3+2i)(3-2i)$ etc.
- Randomly pick signs for ideals (or their squares) above ℓ_i and consider product of generators $\Rightarrow$ smooth norm $L_1^2 L_2$ by construction, i.e. generator corresponds to endomorphism with kernel representation points of order L_1 and $L_1 L_2$

SCALLOP: Precomputation (contd.)

- Fix $\ell_1, \ldots, \ell_n$ to be the smallest n split primes in $\mathbb{Z}[i]$,
  e.g. $(5) = (2+i)(2-i)$, $(13) = (3+2i)(3-2i)$ etc.
- Randomly pick signs for ideals (or their squares) above ℓ_i and consider product of generators $\Rightarrow$ smooth norm $L_1^2 L_2$ by construction, i.e. generator corresponds to endomorphism with kernel representation points of order L_1 and $L_1 L_2$
- Test primality of conductor f of product, then compute corresponding class number and test smoothness using ECM factoring with abort

SCALLOP: Precomputation (contd.)

- Fix $\ell_1, \ldots, \ell_n$ to be the smallest n split primes in $\mathbb{Z}[i]$,
  e.g. $(5) = (2+i)(2-i)$, $(13) = (3+2i)(3-2i)$ etc.
- Randomly pick signs for ideals (or their squares) above ℓ_i and consider product of generators $\Rightarrow$ smooth norm $L_1^2 L_2$ by construction, i.e. generator corresponds to endomorphism with kernel representation points of order L_1 and $L_1 L_2$
- Test primality of conductor f of product, then compute corresponding class number and test smoothness using ECM factoring with abort
- Asymptotically, $L_f(1/2)$ search for $L_f(1/2)$-smooth $|\mathrm{Cl}(\mathfrak{O})|$

SCALLOP: Precomputation (contd.)

- Choose prime characteristic p to maximise efficiency of evaluating the group action (and large enough to prevent attacks), i.e. take $p = \prod_i \ell_i \pm 1$

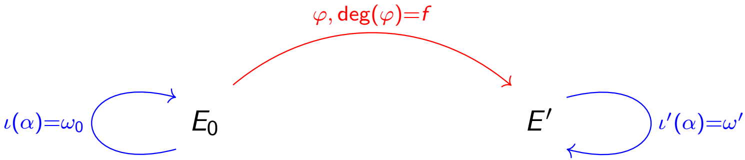
SCALLOP: Precomputation (contd.)

- Choose prime characteristic p to maximise efficiency of evaluating the group action (and large enough to prevent attacks), i.e. take $p = \prod_i \ell_i \pm 1$
- Compute lattice of relations $\mathcal{L}$ by solving instances of discrete logarithm problem in $\mathrm{Cl}(\mathfrak{O})$ (in smooth enough group)
- Compute reduced basis of $\mathcal{L}$ using BKZ as in CSI-FiSh

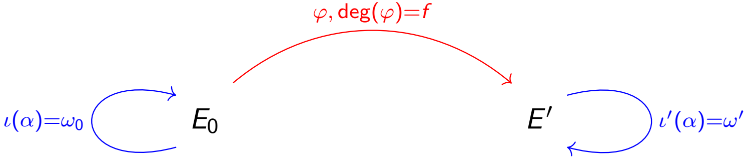
SCALLOP: Precomputation (contd.)

- Choose prime characteristic p to maximise efficiency of evaluating the group action (and large enough to prevent attacks), i.e. take $p = \prod_i \ell_i \pm 1$
- Compute lattice of relations $\mathcal{L}$ by solving instances of discrete logarithm problem in $\mathrm{Cl}(\mathfrak{O})$ (in smooth enough group)
- Compute reduced basis of $\mathcal{L}$ using BKZ as in CSI-FiSh
- Generate a starting curve with $\mathfrak{O}$-orientation

Precomputation: Generation of starting curve



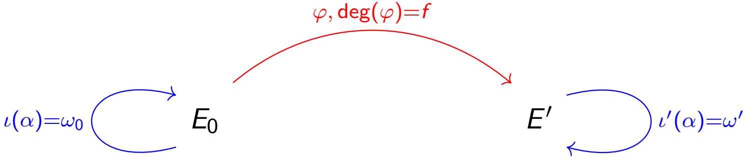
Precomputation: Generation of starting curve



Given characteristic p and large prime f with $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_0 = \mathbb{Z}[\alpha]$ for some α of norm $L_1^2 L_2$. How to compute effective primitive $\mathfrak{O}$ orientation (E', ι')?

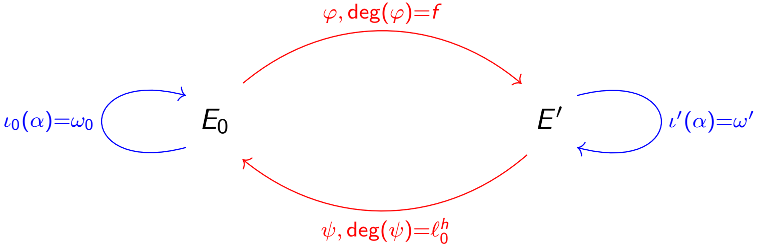
- Push kernel of ω_0 through φ, but deg(f) large prime
  $\Rightarrow$ can't use Vélu's formulae

Precomputation: Generation of starting curve



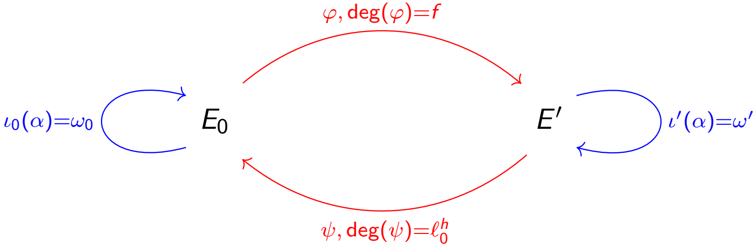
- $\mathfrak{O}_0$ special extremal order (see [KLPT14]) $\Rightarrow$ can find $\gamma \in \mathfrak{O}_0$ of norm M efficiently as soon as $M > p$

Precomputation: Generation of starting curve



- $\mathfrak{O}_0$ special extremal order (see [KLPT14]) $\Rightarrow$ can find $\gamma \in \mathfrak{O}_0$ of norm M efficiently as soon as $M > p$
- Let ℓ_0 small prime not dividing $L_1 L_2$ and $h \in \mathbb{Z}$ such that $\ell_0^h > p/f$ and compute $\gamma \in \mathfrak{O}_0$ of norm $f\ell_0^h$ whose ideal corresponds to endomorphism $\psi \circ \varphi$

Precomputation: Generation of starting curve



- $\mathfrak{O}_0$ special extremal order (see [KLPT14]) $\Rightarrow$ can find $\gamma \in \mathfrak{O}_0$ of norm M efficiently as soon as $M > p$
- Let ℓ_0 small prime not dividing $L_1 L_2$ and $h \in \mathbb{Z}$ such that $\ell_0^h > p/f$ and compute $\gamma \in \mathfrak{O}_0$ of norm $f\ell_0^h$ whose ideal corresponds to endomorphism $\psi \circ \varphi$
- Push kernel of ω_0 through $\psi \circ \varphi$ (see e.g. [FKMT22]), brute-force ψ and compute ω'

SCALLOP: Online phase

- Generator of smooth norm of $\mathfrak{O}$ corresponds to endomorphism ω_E of smooth degree which we represented by kernels of two isogenies
- ω_E stabilizes kernels of isogenies used to compute group action

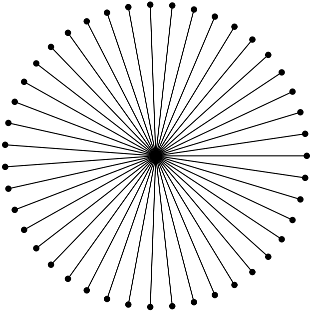


Figure: Isogeny volcano for $\mathfrak{O}$-oriented curves in SCALLOP.

SCALLOP: Online phase

- Generator of smooth norm of $\mathfrak{O}$ corresponds to endomorphism ω_E of smooth degree which we represented by kernels of two isogenies
- ω_E stabilizes kernels of isogenies used to compute group action
- Evaluate group action by transporting explicit orientation along the group action
- Computing explicit orientation leads to slowdown compared to CSI-FiSh with canonical orientation

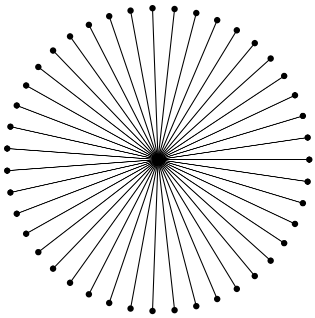


Figure: Isogeny volcano for $\mathfrak{O}$-oriented curves in SCALLOP.

Effective Group Actions: CSI-FiSh vs SCALLOP

<u>CSI-FiSh</u>

- $\mathfrak{O} = \mathbb{Z}[\sqrt{-p}]$

<u>SCALLOP</u>

- $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_0$, f prime

Effective Group Actions: CSI-FiSh vs SCALLOP

CSI-FiSh

- $\mathfrak{O} = \mathbb{Z}[\sqrt{-p}]$
- Expensive class group computation, only feasible for CSIDH-512 parameters

SCALLOP

- $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_0$, f prime
- $|\mathrm{Cl}(\mathfrak{O})|$ free, sieve until smooth enough to compute lattice of relations

Effective Group Actions: CSI-FiSh vs SCALLOP

CSI-FiSh

- $\mathfrak{O} = \mathbb{Z}[\sqrt{-p}]$
- Expensive class group computation, only feasible for CSIDH-512 parameters
- Evaluation of group action with implicit orientation
- Online phase fast

SCALLOP

- $\mathfrak{O} = \mathbb{Z} + f\mathfrak{O}_0$, f prime
- $|\mathrm{Cl}(\mathfrak{O})|$ free, sieve until smooth enough to compute lattice of relations
- Need to compute explicit orientation along group action
- Online phase slower, but feasible for larger security levels

Implementation

Proof of concept implementation in C++ available at:
https://github.com/isogeny-scallop/scallop

- Concrete instantiation for SCALLOP matching the security levels of CSIDH-512 and CSIDH-1024
- Public keys of size roughly 1600bits for SCALLOP-512 and 2300bits for SCALLOP-1024

Implementation

Proof of concept implementation in C++ available at:
https://github.com/isogeny-scallop/scallop

- Concrete instantiation for SCALLOP matching the security levels of CSIDH-512 and CSIDH-1024
- Public keys of size roughly 1600bits for SCALLOP-512 and 2300bits for SCALLOP-1024
- Evaluation of the group action takes about 35 seconds for the smaller and 12.5 minutes for the larger parameter set
- Implementation shows feasibility, but further work needed to make the group action practical

Summary

- Provide framework to evaluate a new family of group actions on oriented elliptic curves via isogenies

Summary

- Provide framework to evaluate a new family of group actions on oriented elliptic curves via isogenies
- Concrete instantiations of class group action using action of class group of imaginary quadratic order with large prime conductor f inside an imaginary quadratic field of small discriminant (SCALLOP)
- This instantiates effective group actions for security levels previously out of reach

Summary

- Provide framework to evaluate a new family of group actions on oriented elliptic curves via isogenies
- Concrete instantiations of class group action using action of class group of imaginary quadratic order with large prime conductor f inside an imaginary quadratic field of small discriminant (SCALLOP)
- This instantiates effective group actions for security levels previously out of reach
- Can build schemes that require to uniquely represent and efficiently act by arbitrary group elements for larger security levels than with CSIDH-512 group action

Questions

Open

- How to make group action evaluation faster?
- How to resolve the scaling issues of SCALLOP?

Questions

Open

- How to make group action evaluation faster?
- How to resolve the scaling issues of SCALLOP?

Thank you!

More details:
ia.cr/2023/058

Questions

Open

- How to make group action evaluation faster?
- How to resolve the scaling issues of SCALLOP?

Thank you!

More details:
ia.cr/2023/058

References

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. CSI-FiSh: efficient isogeny based signatures through class group computations. In International Conference on the Theory and Application of Cryptology and Information Security, pages 227–247. Springer, 2019.

[FFK+23] Luca De Feo, Tako Boris Fouotsa, Péter Kutas, Antonin Leroux, Simon-Philipp Merz, Lorenz Panny, and Benjamin Wesolowski. SCALLOP: scaling the CSI-FiSh. In IACR International Conference on Public-Key Cryptography, pages 345–375. Springer, 2023.

[FKMT22] Tako Boris Fouotsa, Péter Kutas, Simon-Philipp Merz, and Yan Bo Ti. On the isogeny problem with torsion point information. In IACR International Conference on Public-Key Cryptography, pages 142–161. Springer, 2022.

[KLPT14] David Kohel, Kristin Lauter, Christophe Petit, and Jean-Pierre Tignol. On the quaternion ℓ-isogeny path problem. LMS Journal of Computation and Mathematics, 17(A):418–432, 2014.

[Onu21] Hiroshi Onuki. On oriented supersingular elliptic curves. Finite Fields and Their Applications, 69:101777, 2021.