Isogeny-based cryptography on the (abelian) surface

Yan Bo **Ti**, DSO Singapore

DSO Singapore

8 June 2023

Isogeny-based cryptography on the (abelian) surface

What to do with your life when SIDH breaks?

Yan Bo **Ti**, DSO Singapore

DSO Singapore

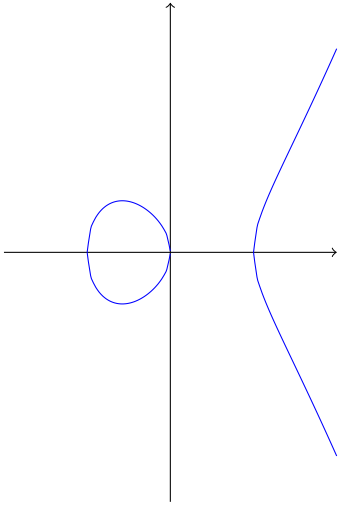
8 June 2023

- Arithmétique et Géométrie
  - Hyperelliptic curves
  - Isogenies
  - Isogeny graphs
- Cryptographie
  - Hash function
  - Cryptanalysis of hash function
  - State of genus-two isogeny cryptography
  - Cryptanalysis of FESTA
  - What is next?

Arithmétique et Géométrie

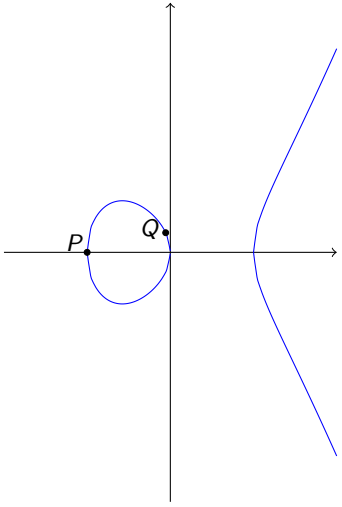
An *elliptic curve* E is a curve in $\mathbb{P}^2(k)$ given by

$$E : y^2 = \text{cubic in } x$$



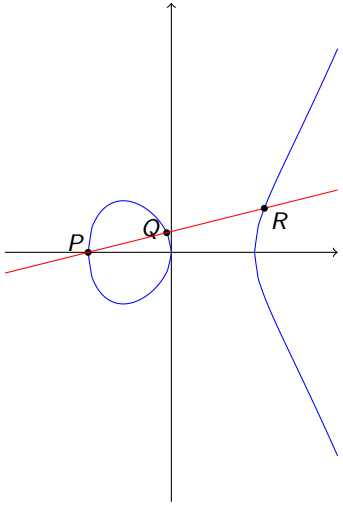
An *elliptic curve* E is a curve in $\mathbb{P}^2(k)$ given by

$$E : y^2 = \text{cubic in } x$$



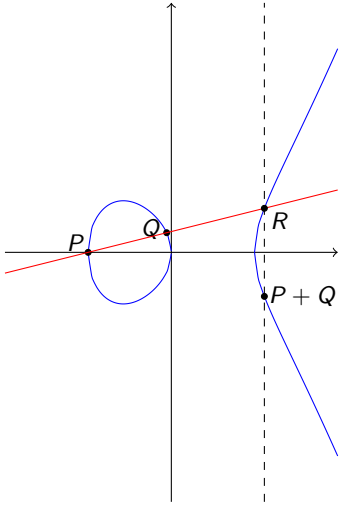
An *elliptic curve* E is a curve in $\mathbb{P}^2(k)$ given by

$$E : y^2 = \text{cubic in } x$$



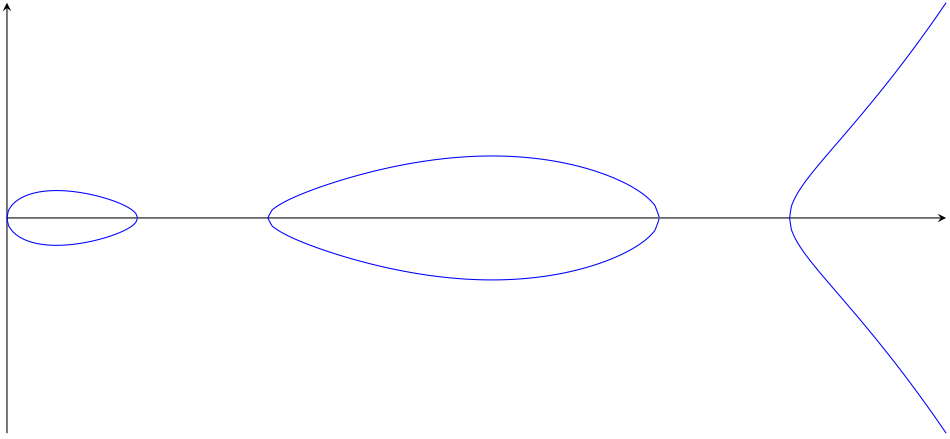
An *elliptic curve* E is a curve in $\mathbb{P}^2(k)$ given by

$$E : y^2 = \text{cubic in } x$$



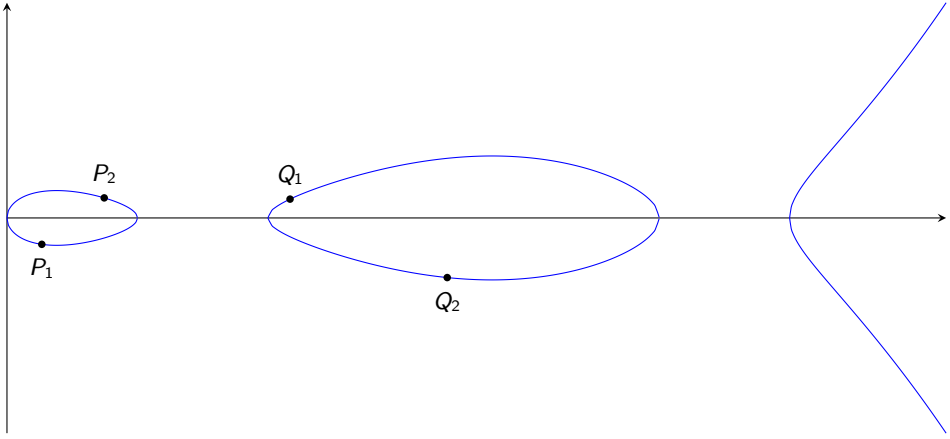
A *hyperelliptic curve* (of genus 2) H is a curve in $\mathbb{P}^2(k)$ given by

$$H : y^2 = \text{sextic or quintic in } x$$



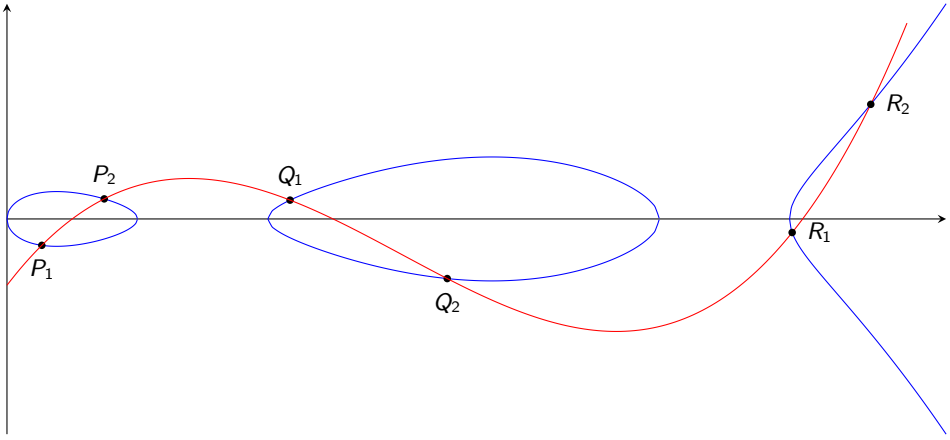
A *hyperelliptic curve* (of genus 2) H is a curve in $\mathbb{P}^2(k)$ given by

$$H : y^2 = \text{sextic or quintic in } x$$



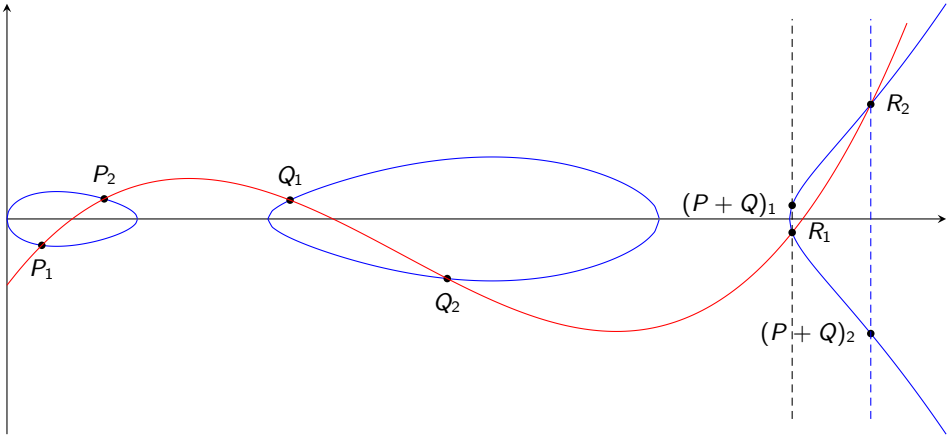
A *hyperelliptic curve* (of genus 2) H is a curve in $\mathbb{P}^2(k)$ given by

$$H : y^2 = \text{sextic or quintic in } x$$



A *hyperelliptic curve* (of genus 2) H is a curve in $\mathbb{P}^2(k)$ given by

$$H : y^2 = \text{sextic or quintic in } x$$



Group law comes from divisors.

Group law comes from divisors.
Let E be an elliptic curve.

Group law comes from divisors.

Let E be an elliptic curve.

- Weil divisor: Finite formal sum of points on E

$$D = \sum_{P \in E} n_P P \,,$$

where $n_P \in \mathbb{Z}$. The set of Weil divisors form a group under addition.

Group law comes from divisors.

Let E be an elliptic curve.

- Weil divisor: Finite formal sum of points on E

$$D = \sum_{P \in E} n_P P \,,$$

  where $n_P \in \mathbb{Z}$. The set of Weil divisors form a group under addition.
- Degree: $\deg D = \sum n_P$.

Group law comes from divisors.

Let E be an elliptic curve.

- Weil divisor: Finite formal sum of points on E

$$D = \sum_{P \in E} n_P P \,,$$

  where $n_P \in \mathbb{Z}$. The set of Weil divisors form a group under addition.
- Degree: $\deg D = \sum n_P$.
- Principal divisor: $\operatorname{div}(f) = \sum_{P \in E} \operatorname{ord}_P(f) P$.

Group law comes from divisors.

Let E be an elliptic curve.

- Weil divisor: Finite formal sum of points on E

$$D = \sum_{P \in E} n_P P \,,$$

  where $n_P \in \mathbb{Z}$. The set of Weil divisors form a group under addition.
- Degree: $\deg D = \sum n_P$.
- Principal divisor: $\operatorname{div}(f) = \sum_{P \in E} \operatorname{ord}_P(f) P$.
- Jacobian of E = Divisors of degree 0 modulo principal divisors (aka $\operatorname{Pic}^0(E)$).

Group law comes from divisors.

Let E be an elliptic curve.

- Weil divisor: Finite formal sum of points on E

$$D = \sum_{P \in E} n_P P\,,$$

  where $n_P \in \mathbb{Z}$. The set of Weil divisors form a group under addition.
- Degree: $\deg D = \sum n_P$.
- Principal divisor: $\operatorname{div}(f) = \sum_{P \in E} \operatorname{ord}_P(f) P$.
- Jacobian of E = Divisors of degree 0 modulo principal divisors (aka $\mathrm{Pic}^0(E)$).

Theorem

The map

$$\sigma : \mathrm{Pic}^0(E) \to E$$
$$D \sim (P) - (\mathcal{O}) \mapsto P$$

is an isomorphism.

- Jacobians of hyperelliptic curves are *abelian varieties*. We are interested in genus 2 hyperelliptic curves which give *abelian surfaces*.

- Jacobians of hyperelliptic curves are *abelian varieties*. We are interested in genus 2 hyperelliptic curves which give *abelian surfaces*.
- Abelian surfaces also include the product of two elliptic curves.

- Jacobians of hyperelliptic curves are *abelian varieties*. We are interested in genus 2 hyperelliptic curves which give *abelian surfaces*.
- Abelian surfaces also include the product of two elliptic curves.
- There is a special property: *principal polarisation*.

- Jacobians of hyperelliptic curves are *abelian varieties*. We are interested in genus 2 hyperelliptic curves which give *abelian surfaces*.
- Abelian surfaces also include the product of two elliptic curves.
- There is a special property: *principal polarisation*.
- We want to preserve this.

- Jacobians of hyperelliptic curves are *abelian varieties*. We are interested in genus 2 hyperelliptic curves which give *abelian surfaces*.
- Abelian surfaces also include the product of two elliptic curves.
- There is a special property: *principal polarisation*.
- We want to preserve this.
- There are other properties one can choose to keep that can be useful for other cryptographic schemes.

A morphism $f : A \to A'$ is called an *isogeny* if it is surjective, with finite kernel.

Vertices:

Edges:

A morphism $f : A \to A'$ is called an *isogeny* if it is surjective, with finite kernel.
Fun facts:

- Isogenies are group homomorphisms.
- If ϕ is a separable isogeny, then $\deg \phi = \# \ker \phi$.

Vertices:

Edges:

A morphism $f : A \to A'$ is called an *isogeny* if it is surjective, with finite kernel.
Fun facts:

- Isogenies are group homomorphisms.
- If ϕ is a separable isogeny, then $\deg \phi = \# \ker \phi$.

Theorem

There is a 1-1 correspondence between finite subgroups $K \subseteq A$ and separable isogenies $f : A \to A'$.

Vertices:

Edges:

A morphism $f : A \to A'$ is called an *isogeny* if it is surjective, with finite kernel.
Fun facts:

- Isogenies are group homomorphisms.
- If ϕ is a separable isogeny, then $\deg \phi = \# \ker \phi$.

Theorem

There is a 1-1 correspondence between finite subgroups $K \subseteq A$ and separable isogenies $f : A \to A'$.

Recall: Need principal polarisations. So we add a property to the subgroups: isotropy.

Vertices:

Edges:

A morphism $f : A \to A'$ is called an *isogeny* if it is surjective, with finite kernel.
Fun facts:

- Isogenies are group homomorphisms.
- If ϕ is a separable isogeny, then $\deg \phi = \# \ker \phi$.

Theorem

There is a 1-1 correspondence between finite subgroups $K \subseteq A$ and separable isogenies $f : A \to A'$.

Recall: Need principal polarisations. So we add a property to the subgroups: isotropy.

ℓ-Isogeny graphs:

Vertices: Abelian varieties

Edges: Separable ℓ-isogenies

A morphism $f : A \to A'$ is called an *isogeny* if it is surjective, with finite kernel.
Fun facts:

- Isogenies are group homomorphisms.
- If ϕ is a separable isogeny, then $\deg \phi = \# \ker \phi$.

Theorem

There is a 1-1 correspondence between finite subgroups $K \subseteq A$ and separable isogenies $f : A \to A'$.

Recall: Need principal polarisations. So we add a property to the subgroups: isotropy.

ℓ-Isogeny graphs:

Vertices: Abelian varieties

Edges: Separable ℓ-isogenies

We will focus on isogeny graphs of Principally Polarised Abelian Surfaces (PPAS).

A morphism $f : A \to A'$ is called an *isogeny* if it is surjective, with finite kernel.
Fun facts:

- Isogenies are group homomorphisms.
- If ϕ is a separable isogeny, then $\deg \phi = \# \ker \phi$.

Theorem

There is a 1-1 correspondence between finite subgroups $K \subseteq A$ and separable isogenies $f : A \to A'$.

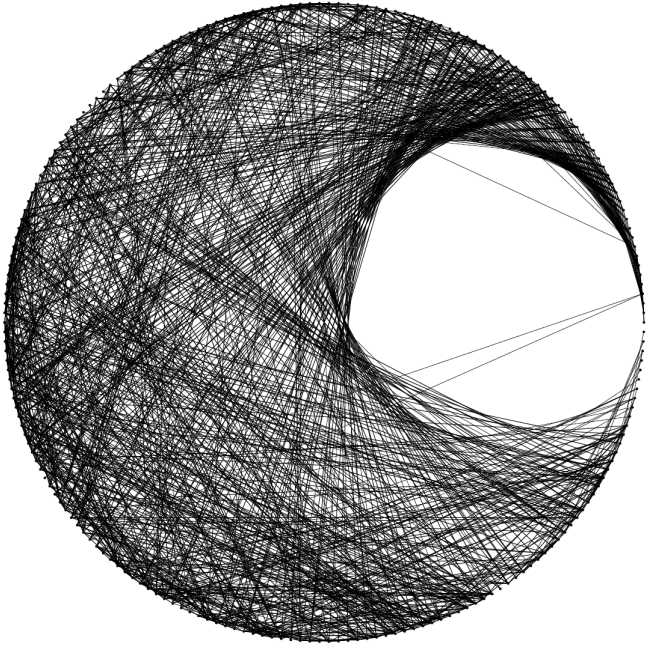
Recall: Need principal polarisations. So we add a property to the subgroups: isotropy.

ℓ-Isogeny graphs:

Vertices: Isomorphism classes of PPASs

Edges: (ℓ, ℓ)-isogenies

We will focus on isogeny graphs of Principally Polarised Abelian Surfaces (PPAS).



Cryptographie

- Hash function

$$H : \{0,1\}^* \to \{0,1\}^n .$$

- Hash function

$$H : \{0,1\}^* \to \{0,1\}^n .$$

- Security properties:
  1. Collision resistance: Finding x_1, x_2 such that $H(x_1) = H(x_2)$ is hard.
  2. Pre-image resistance: Given $y = H(x)$, finding x is hard.

- Hash function
$$H : \{0,1\}^* \to \{0,1\}^n .$$
- Security properties:
  1. Collision resistance: Finding x_1, x_2 such that $H(x_1) = H(x_2)$ is hard.
  2. Pre-image resistance: Given $y = H(x)$, finding x is hard.
- Preview of hash function on isogeny graph:

Input	String of bits.
Ouput	Vertex on graph.
Method	Use input as random walk with random starting vertex.

- Charles, Goren, and Lauter showed that Brandt matrices[1] are Ramanujan [CGL09].

[1] Connections to supersingular abelian varieties exist, but are beyond the scope of this talk.

- Charles, Goren, and Lauter showed that Brandt matrices[1] are Ramanujan [CGL09].
- Hinted at the use of superspecial graphs on hash functions, but proposed a hash function in genus one [CLG09].

[1] Connections to supersingular abelian varieties exist, but are beyond the scope of this talk.

- Charles, Goren, and Lauter showed that Brandt matrices[1] are Ramanujan [CGL09].
- Hinted at the use of superspecial graphs on hash functions, but proposed a hash function in genus one [CLG09].
- Takashima outlined a hash function using (2,2)-isogenies on the superspecial graph of abelian surfaces [Tak18].

[1] Connections to supersingular abelian varieties exist, but are beyond the scope of this talk.

- Charles, Goren, and Lauter showed that Brandt matrices[1] are Ramanujan [CGL09].
- Hinted at the use of superspecial graphs on hash functions, but proposed a hash function in genus one [CLG09].
- Takashima outlined a hash function using (2,2)-isogenies on the superspecial graph of abelian surfaces [Tak18].
  - Set a prime p, and a vertex (superspecial abelian surface).

[1] Connections to supersingular abelian varieties exist, but are beyond the scope of this talk.

- Charles, Goren, and Lauter showed that Brandt matrices[1] are Ramanujan [CGL09].
- Hinted at the use of superspecial graphs on hash functions, but proposed a hash function in genus one [CLG09].
- Takashima outlined a hash function using (2,2)-isogenies on the superspecial graph of abelian surfaces [Tak18].
  - Set a prime p, and a vertex (superspecial abelian surface).
  - Use input bits to choose a non-backtracking path at each vertex.

[1] Connections to supersingular abelian varieties exist, but are beyond the scope of this talk.

- Charles, Goren, and Lauter showed that Brandt matrices[1] are Ramanujan [CGL09].
- Hinted at the use of superspecial graphs on hash functions, but proposed a hash function in genus one [CLG09].
- Takashima outlined a hash function using (2,2)-isogenies on the superspecial graph of abelian surfaces [Tak18].
  - Set a prime p, and a vertex (superspecial abelian surface).
  - Use input bits to choose a non-backtracking path at each vertex.
  - Output final vertex of path.

[1] Connections to supersingular abelian varieties exist, but are beyond the scope of this talk.

- Charles, Goren, and Lauter showed that Brandt matrices[1] are Ramanujan [CGL09].
- Hinted at the use of superspecial graphs on hash functions, but proposed a hash function in genus one [CLG09].
- Takashima outlined a hash function using (2,2)-isogenies on the superspecial graph of abelian surfaces [Tak18].
  - Set a prime p, and a vertex (superspecial abelian surface).
  - Use input bits to choose a non-backtracking path at each vertex.
  - Output final vertex of path.
- Jordan and Zaytman proved connectedness and expander properties of superspecial isogeny graph [JZ20].

[1] Connections to supersingular abelian varieties exist, but are beyond the scope of this talk.

- Charles, Goren, and Lauter showed that Brandt matrices[1] are Ramanujan [CGL09].
- Hinted at the use of superspecial graphs on hash functions, but proposed a hash function in genus one [CLG09].
- Takashima outlined a hash function using (2,2)-isogenies on the superspecial graph of abelian surfaces [Tak18].
  - Set a prime p, and a vertex (superspecial abelian surface).
  - Use input bits to choose a non-backtracking path at each vertex.
  - Output final vertex of path.
- Jordan and Zaytman proved connectedness and expander properties of superspecial isogeny graph [JZ20].

Security properties:

1. Collision resistance
2. Pre-image resistance

[1] Connections to supersingular abelian varieties exist, but are beyond the scope of this talk.

- Charles, Goren, and Lauter showed that Brandt matrices[1] are Ramanujan [CGL09].
- Hinted at the use of superspecial graphs on hash functions, but proposed a hash function in genus one [CLG09].
- Takashima outlined a hash function using (2,2)-isogenies on the superspecial graph of abelian surfaces [Tak18].
  - Set a prime p, and a vertex (superspecial abelian surface).
  - Use input bits to choose a non-backtracking path at each vertex.
  - Output final vertex of path.
- Jordan and Zaytman proved connectedness and expander properties of superspecial isogeny graph [JZ20].

Security properties:

1. Collision resistance
2. Pre-image resistance

[1] Connections to supersingular abelian varieties exist, but are beyond the scope of this talk.

Proposition

Let H be a hyperelliptic curve of genus 2 over $\mathbb{F}_q$. Let K be a finite, non-trivial, $\mathbb{F}_q$-rational subgroup of $J_H(\mathbb{F}_q)$. There exists a PPAS A over $\mathbb{F}_q$, and an isogeny $\phi : J_H \to A$ with kernel K, if and only if K is a maximal ℓ-isotropic subgroup of $J_H[\ell]$ for some positive integer ℓ.

- Isogenies can be studied by looking at their kernels.

Proposition ([FT19])

Let A be a PPAS. The maximal ℓ^n-isotropic subgroups of $A[\ell^n] \cong C_{\ell^n}^4$ are isomorphic to

$$C_{\ell^n} \times C_{\ell^n} \quad \text{or} \quad C_{\ell^n} \times C_{\ell^{n-k}} \times C_{\ell^k} \quad \text{or } A[\ell^m]$$

where $1 \leq k \leq \lfloor n/2 \rfloor$ and $m \leq n$.

Proposition ([FT19])

Let A be a PPAS. The maximal ℓ^n-isotropic subgroups of $A[\ell^n] \cong C_{\ell^n}^4$ are isomorphic to

$$C_{\ell^n} \times C_{\ell^n} \quad \text{or} \quad C_{\ell^n} \times C_{\ell^{n-k}} \times C_{\ell^k} \quad \text{or } A[\ell^m]$$

where $1 \leq k \leq \lfloor n/2 \rfloor$ and $m \leq n$.

Proof.

If K is cyclic, then $K \cong C_\ell \subseteq C_\ell \times C_\ell$, hence not maximal.
For rank 2: Use maximality of subgroups.
For rank 3: Use symmetry of the kernel of the dual isogeny. □

Proposition ([FT19])

Let A be a PPAS. The maximal ℓ^n-isotropic subgroups of $A[\ell^n] \cong C_{\ell^n}^4$ are isomorphic to

$$C_{\ell^n} \times C_{\ell^n} \quad \text{or} \quad C_{\ell^n} \times C_{\ell^{n-k}} \times C_{\ell^k} \quad \text{or } A[\ell^m]$$

where $1 \leq k \leq \lfloor n/2 \rfloor$ and $m \leq n$.

Proof.

If K is cyclic, then $K \cong C_\ell \subseteq C_\ell \times C_\ell$, hence not maximal.
For rank 2: Use maximality of subgroups.
For rank 3: Use symmetry of the kernel of the dual isogeny. □

How does structure of subgroup affect isogenies?

Proposition ([FT19])

Let $\mathcal{G}_{p,\ell}$ be the (ℓ,ℓ)-isogeny graph of PPAS over $\overline{F}_p$. Then the number of elements in the n-sphere, where $n > 2$, centred around an arbitrary vertex is

$$\ell^{2n-3}(\ell^2+1)(\ell+1)\left(\ell^n + \ell\frac{\ell^{n-2}-1}{\ell-1}+1\right)$$

if n is even, and

$$\ell^{2n-3}(\ell^2+1)(\ell+1)\left(\ell^n + \frac{\ell^{n-1}-1}{\ell-1}\right)$$

if n is odd.

Proposition ([FT19])

Let $\mathcal{G}_{p,\ell}$ be the (ℓ,ℓ)-isogeny graph of PPAS over $\overline{F}_p$. Then the number of elements in the n-sphere, where $n > 2$, centred around an arbitrary vertex is

$$\ell^{2n-3}(\ell^2+1)(\ell+1)\left(\ell^n + \ell\frac{\ell^{n-2}-1}{\ell-1}+1\right)$$

if n is even, and

$$\ell^{2n-3}(\ell^2+1)(\ell+1)\left(\ell^n + \frac{\ell^{n-1}-1}{\ell-1}\right)$$

if n is odd.

Proof.

- Count number of ℓ^n-maximal isotropic subgroups.
- Sum them together.

□

- Fix primes p and ℓ, and a PPAS A.
- Consider kernel $K \subseteq A[\ell^n]$, i.e. fix a ℓ^n-maximal isotropic subgroup.
- How many ways can we get from $A \to A/K$?

- Fix primes p and ℓ, and a PPAS A.
- Consider kernel $K \subseteq A[\ell^n]$, i.e. fix a ℓ^n-maximal isotropic subgroup.
- How many ways can we get from $A \to A/K$?

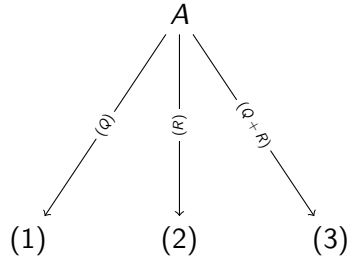
The key observation is that the number of $C_\ell \times C_\ell$ isotropic subgroups of K corresponds with the number choices for the first isogeny.

- Fix p, and a PPAS A.
- Let $\ell = 2$ and let $K = \langle P, Q, R \rangle \cong C_4 \times C_2 \times C_2$.
- K has order 16, so we expect $A \to A/K$ to be a sequence of 2 $(2,2)$-isogenies.

A

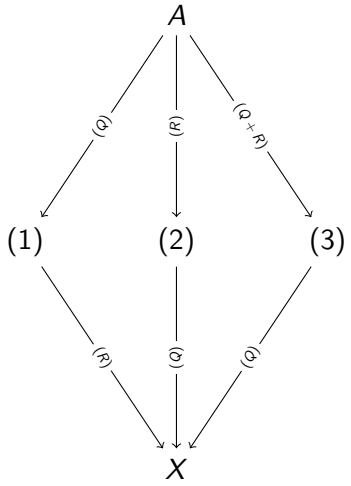
X

- Fix p, and a PPAS A.
- Let $\ell = 2$ and let $K = \langle P, Q, R \rangle \cong C_4 \times C_2 \times C_2$.
- K has order 16, so we expect $A \to A/K$ to be a sequence of 2 $(2,2)$-isogenies.
- First step: $\langle [2]P, Q \rangle$, $\langle [2]P, R \rangle$, $\langle [2]P, Q + R \rangle$.

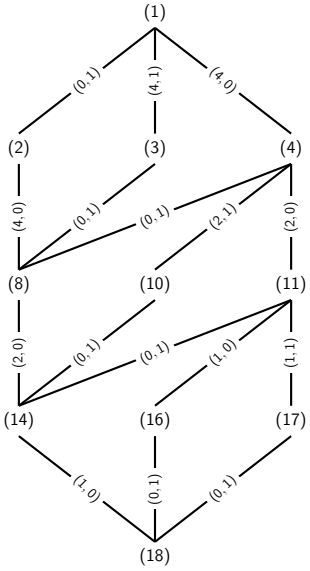


X

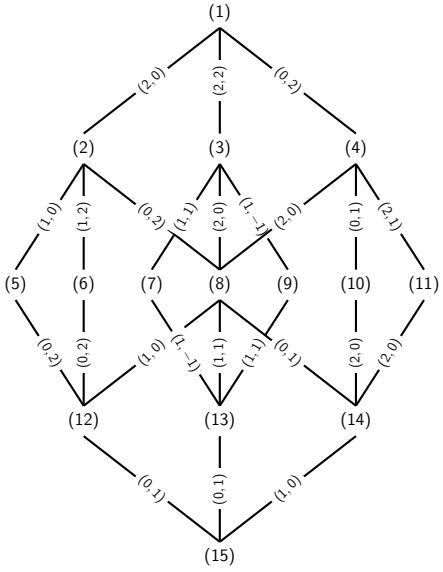
- Fix p, and a PPAS A.
- Let $\ell = 2$ and let $K = \langle P, Q, R \rangle \cong C_4 \times C_2 \times C_2$.
- K has order 16, so we expect $A \to A/K$ to be a sequence of 2 $(2,2)$-isogenies.
- First step: $\langle [2]P, Q \rangle$, $\langle [2]P, R \rangle$, $\langle [2]P, Q + R \rangle$.
- Second step: No choices.

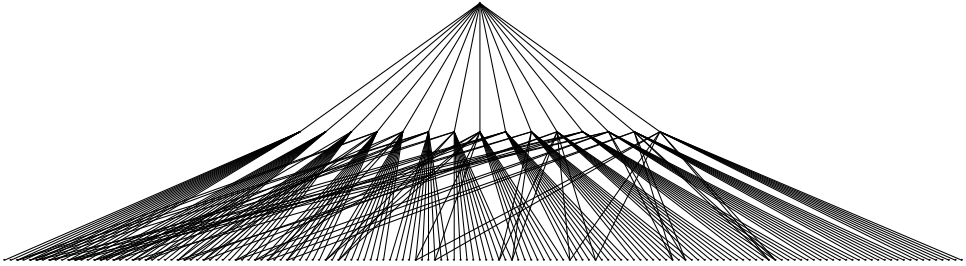


Example: $C_{16} \times C_8 \times C_2$



Example: $C_{16} \times C_4 \times C_4$





Proposition ([FT19])

Let $P(n, a)$ be the number of paths in a $(C_{\ell^n} \times C_{\ell^{n-a}} \times C_{\ell^a})$-isogeny. Then $P(n, a)$ satisfies the following recursive equation:

$$P(n, a) = \ell P(n-1, a-1) + P(n-1, a)\,,$$

where $1 \leq a < n/2$, and with the following boundary conditions:

$$P(n, 0) = 1, \quad P(2, 1) = \ell + 1\,.$$

Proposition ([FT19])

Let $P(n, a)$ be the number of paths in a $(C_{\ell^n} \times C_{\ell^{n-a}} \times C_{\ell^a})$-isogeny. Then $P(n, a)$ satisfies the following recursive equation:

$$P(n, a) = \ell P(n-1, a-1) + P(n-1, a)\,,$$

where $1 \leq a < n/2$, and with the following boundary conditions:

$$P(n, 0) = 1, \quad P(2, 1) = \ell + 1\,.$$

Proof.

Similar to diamond example: consider the number of choices available as the first step, then obtain the recursive relation. □

- Easy to encounter collisions if paths contain diamonds

- Easy to encounter collisions if paths contain diamonds
- Castryck, Decru, and Smith [CDS20] proposed a hash function whose paths:

- Easy to encounter collisions if paths contain diamonds
- Castryck, Decru, and Smith [CDS20] proposed a hash function whose paths:
  - do not back-track;
  - avoid collisions.

- Easy to encounter collisions if paths contain diamonds
- Castryck, Decru, and Smith [CDS20] proposed a hash function whose paths:
  - do not back-track;
  - avoid collisions.

Comparing to dimension 1:

- Easy to encounter collisions if paths contain diamonds
- Castryck, Decru, and Smith [CDS20] proposed a hash function whose paths:
  - do not back-track;
  - avoid collisions.

Comparing to dimension 1:

- SSECs have three 2-isogenies

- Easy to encounter collisions if paths contain diamonds
- Castryck, Decru, and Smith [CDS20] proposed a hash function whose paths:
  - do not back-track;
  - avoid collisions.

Comparing to dimension 1:

- SSECs have three 2-isogenies, two non-backtracking 2-isogenies.

- Easy to encounter collisions if paths contain diamonds
- Castryck, Decru, and Smith [CDS20] proposed a hash function whose paths:
  - do not back-track;
  - avoid collisions.

Comparing to dimension 1:

- SSECs have three 2-isogenies, two non-backtracking 2-isogenies.
- PPSSASs have fifteen (2,2)-isogenies

- Easy to encounter collisions if paths contain diamonds
- Castryck, Decru, and Smith [CDS20] proposed a hash function whose paths:
  - do not back-track;
  - avoid collisions.

Comparing to dimension 1:

- SSECs have three 2-isogenies, two non-backtracking 2-isogenies.
- PPSSASs have fifteen (2,2)-isogenies, eight non-backtracking and "good" (2,2)-isogenies.

- Easy to encounter collisions if paths contain diamonds
- Castryck, Decru, and Smith [CDS20] proposed a hash function whose paths:
  - do not back-track;
  - avoid collisions.

Comparing to dimension 1:

- SSECs have three 2-isogenies, two non-backtracking 2-isogenies.
- PPSSASs have fifteen (2,2)-isogenies, eight non-backtracking and "good" (2,2)-isogenies.

Problem: What happens when you hit a product?

- Superspecial graph ($g \geq 2$) contains reducible abelian varieties.

- Superspecial graph ($g \geq 2$) contains reducible abelian varieties.
- Probability of reducible abelian varieties is $O(1/p)$.

- Superspecial graph ($g \geq 2$) contains reducible abelian varieties.
- Probability of reducible abelian varieties is $O(1/p)$.

Theorem ([CS20])

Let A, A' be PPSSAV over $\overline{\mathbb{F}}_p$ of dimension $g \geq 2$.

1. *There is a classical $\tilde{O}(p^{g-1})$ algorithm which finds an isogeny $\phi : A \to A'$ in the superspecial graph.*
2. *There is a quantum $\tilde{O}(\sqrt{p^{g-1}})$ algorithm which finds an isogeny $\phi : A \to A'$ in the superspecial graph.*

- Superspecial graph ($g \geq 2$) contains reducible abelian varieties.
- Probability of reducible abelian varieties is $O(1/p)$.

Theorem ([CS20])

Let A, A' be PPSSAV over $\overline{\mathbb{F}}_p$ of dimension $g \geq 2$.

1. *There is a classical $\tilde{O}(p^{g-1})$ algorithm which finds an isogeny $\phi : A \to A'$ in the superspecial graph.*
2. *There is a quantum $\tilde{O}(\sqrt{p^{g-1}})$ algorithm which finds an isogeny $\phi : A \to A'$ in the superspecial graph.*

- AIM of attacker: Find reducible ones.
  This is finding special vertices in the graph.
  E.g. Delfs–Galbraith [DG16] finding $E/\mathbb{F}_p$ in the full SSEC graph.

- Superspecial graph ($g \geq 2$) contains reducible abelian varieties.
- Probability of reducible abelian varieties is $O(1/p)$.

Theorem ([CS20])

Let A, A' be PPSSAV over $\overline{\mathbb{F}}_p$ of dimension $g \geq 2$.

1. *There is a classical $\tilde{O}(p^{g-1})$ algorithm which finds an isogeny $\phi : A \to A'$ in the superspecial graph.*
2. *There is a quantum $\tilde{O}(\sqrt{p^{g-1}})$ algorithm which finds an isogeny $\phi : A \to A'$ in the superspecial graph.*

- AIM of attacker: Find reducible ones.
  This is finding special vertices in the graph.
  E.g. Delfs–Galbraith [DG16] finding $E/\mathbb{F}_p$ in the full SSEC graph.

Genus 2 the sweet spot?

- Superspecial graph ($g \geq 2$) contains reducible abelian varieties.
- Probability of reducible abelian varieties is $O(1/p)$.

Theorem ([CS20])

Let A, A' be PPSSAV over $\overline{\mathbb{F}}_p$ of dimension $g \geq 2$.

1. *There is a classical $\tilde{O}(p^{g-1})$ algorithm which finds an isogeny $\phi : A \to A'$ in the superspecial graph.*
2. *There is a quantum $\tilde{O}(\sqrt{p^{g-1}})$ algorithm which finds an isogeny $\phi : A \to A'$ in the superspecial graph.*

- AIM of attacker: Find reducible ones.
  This is finding special vertices in the graph.
  E.g. Delfs–Galbraith [DG16] finding $E/\mathbb{F}_p$ in the full SSEC graph.

Genus 2 the sweet spot? Can we remove products?

Definition

Let $k = \mathbb{F}_{p^n}$, then E/k is *supersingular* if any one (hence all) of the following is true:

(i) $E[p^r] = 0$ for one (all) $r \geq 1$.

(ii) $\mathrm{End}(E)$, the endomorphism ring over the closure of k is an order in a quaternion algebra.

Definition

Let $k = \mathbb{F}_{p^n}$, then E/k is *supersingular* if any one (hence all) of the following is true:

(i) $E[p^r] = 0$ for one (all) $r \geq 1$.

(ii) $\mathrm{End}(E)$, the endomorphism ring over the closure of k is an order in a quaternion algebra.

Definition

A/k is *supersingular* if A is isogenous over $\overline{k}$ to a product of SSEC.
A/k is *superspecial* if A is isomorphic over $\overline{k}$ to a product of SSEC.

Definition

Let $k = \mathbb{F}_{p^n}$, then E/k is *supersingular* if any one (hence all) of the following is true:

(i) $E[p^r] = 0$ for one (all) $r \geq 1$.

(ii) $\mathrm{End}(E)$, the endomorphism ring over the closure of k is an order in a quaternion algebra.

Definition

A/k is *supersingular* if A is isogenous over $\overline{k}$ to a product of SSEC.
A/k is *superspecial* if A is isomorphic over $\overline{k}$ to a product of SSEC.

Theorem (Shioda, Deligne, Oort)

Let A be an abelian variety over a field of characteristic p and of dimension $g \geq 2$, and let $E^g \to A$ be an isogeny of degree d, where E is a supersingular elliptic curve. If $p \nmid d$, then $A \cong E^g$.

Let k be a finite field of characteristic p. Consider the finite group schemes

$$\alpha_p \cong \mathrm{Spec}(k[X]/X^p) \quad \text{and} \quad \mu_p \cong \mathrm{Spec}(k[X]/(X^p - 1)) .$$

Let k be a finite field of characteristic p. Consider the finite group schemes

$$\alpha_p \cong \mathrm{Spec}(k[X]/X^p) \quad \text{and} \quad \mu_p \cong \mathrm{Spec}(k[X]/(X^p - 1)) .$$

Let A/k be an abelian variety of dimension g.

Let k be a finite field of characteristic p. Consider the finite group schemes

$$\alpha_p \cong \mathrm{Spec}(k[X]/X^p) \quad \text{and} \quad \mu_p \cong \mathrm{Spec}(k[X]/(X^p - 1)).$$

Let A/k be an abelian variety of dimension g.

- The p-rank of A is given by $f = \dim_{\mathbb{F}_p}(\mathrm{Hom}(\mu_p, A[p]))$,

Let k be a finite field of characteristic p. Consider the finite group schemes

$$\alpha_p \cong \mathrm{Spec}(k[X]/X^p) \quad \text{and} \quad \mu_p \cong \mathrm{Spec}(k[X]/(X^p - 1)).$$

Let A/k be an abelian variety of dimension g.

- The p-rank of A is given by $f = \dim_{\mathbb{F}_p}(\mathrm{Hom}(\mu_p, A[p]))$,
- the a-number of A is given by $a = \dim_k(\mathrm{Hom}(\alpha_p, A[p]))$.

Let k be a finite field of characteristic p. Consider the finite group schemes

$$\alpha_p \cong \mathrm{Spec}(k[X]/X^p) \quad \text{and} \quad \mu_p \cong \mathrm{Spec}(k[X]/(X^p - 1)).$$

Let A/k be an abelian variety of dimension g.

- The p-rank of A is given by $f = \dim_{\mathbb{F}_p}(\mathrm{Hom}(\mu_p, A[p]))$,
- the a-number of A is given by $a = \dim_k(\mathrm{Hom}(\alpha_p, A[p]))$.
- It holds that $0 \leq f \leq g$ and $1 \leq a + f \leq g$,

Let k be a finite field of characteristic p. Consider the finite group schemes

$$\alpha_p \cong \mathrm{Spec}(k[X]/X^p) \quad \text{and} \quad \mu_p \cong \mathrm{Spec}(k[X]/(X^p - 1)).$$

Let A/k be an abelian variety of dimension g.

- The p-rank of A is given by $f = \dim_{\mathbb{F}_p}(\mathrm{Hom}(\mu_p, A[p]))$,
- the a-number of A is given by $a = \dim_k(\mathrm{Hom}(\alpha_p, A[p]))$.
- It holds that $0 \leq f \leq g$ and $1 \leq a + f \leq g$,
- hence, geometrically, $A[p] \cong (\mathbb{Z}/p\mathbb{Z})^f$.

Let k be a finite field of characteristic p. Consider the finite group schemes

$$\alpha_p \cong \mathrm{Spec}(k[X]/X^p) \quad \text{and} \quad \mu_p \cong \mathrm{Spec}(k[X]/(X^p-1)).$$

Let A/k be an abelian variety of dimension g.

- The p-rank of A is given by $f = \dim_{\mathbb{F}_p}(\mathrm{Hom}(\mu_p, A[p]))$,
- the a-number of A is given by $a = \dim_k(\mathrm{Hom}(\alpha_p, A[p]))$.
- It holds that $0 \leq f \leq g$ and $1 \leq a + f \leq g$,
- hence, geometrically, $A[p] \cong (\mathbb{Z}/p\mathbb{Z})^f$.

f	a	$A[p]$	Type	Codim.
2	0	L^2	Ordinary	0
1	1	$L \oplus I_{1,1}$	Non-ordinary	1
0	1	$I_{2,1}$	Supersingular	2
0	2	$I_{1,1}^2$	Superspecial	3

Moving to supersingular non-superspecial:

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

Ending on a reducible surface for CDS hash:

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

Ending on a reducible surface for CDS hash:

- Three strategies for ending on reducible surface:

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

Ending on a reducible surface for CDS hash:

- Three strategies for ending on reducible surface:
  1. Ignore;

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

Ending on a reducible surface for CDS hash:

- Three strategies for ending on reducible surface:
  1. Ignore;
  2. Deterministically find a non-reducible neighbour;

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

Ending on a reducible surface for CDS hash:

- Three strategies for ending on reducible surface:
  1. Ignore;
  2. Deterministically find a non-reducible neighbour;
  3. Glue 2-torsion.

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

Ending on a reducible surface for CDS hash:

- Three strategies for ending on reducible surface:
  1. Ignore;
  2. Deterministically find a non-reducible neighbour;
  3. Glue 2-torsion.
- Raises possibility of failures.

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

Ending on a reducible surface for CDS hash:

- Three strategies for ending on reducible surface:
  1. Ignore;
  2. Deterministically find a non-reducible neighbour;
  3. Glue 2-torsion.
- Raises possibility of failures.

Encountering reducible surfaces en route:

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

Ending on a reducible surface for CDS hash:

- Three strategies for ending on reducible surface:
  1. Ignore;
  2. Deterministically find a non-reducible neighbour;
  3. Glue 2-torsion.
- Raises possibility of failures.

Encountering reducible surfaces en route:

- Simple side-channel analysis would show that.

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

Ending on a reducible surface for CDS hash:

- Three strategies for ending on reducible surface:
  1. Ignore;
  2. Deterministically find a non-reducible neighbour;
  3. Glue 2-torsion.
- Raises possibility of failures.

Encountering reducible surfaces en route:

- Simple side-channel analysis would show that.
- Attack using meet-in-the-middle.

Moving to supersingular non-superspecial:

- Working with (ℓ, ℓ)-isogenies for ℓ, p coprime, we do not get reducible surfaces.
- Work over extensions of $\mathbb{F}_{p^2}$.

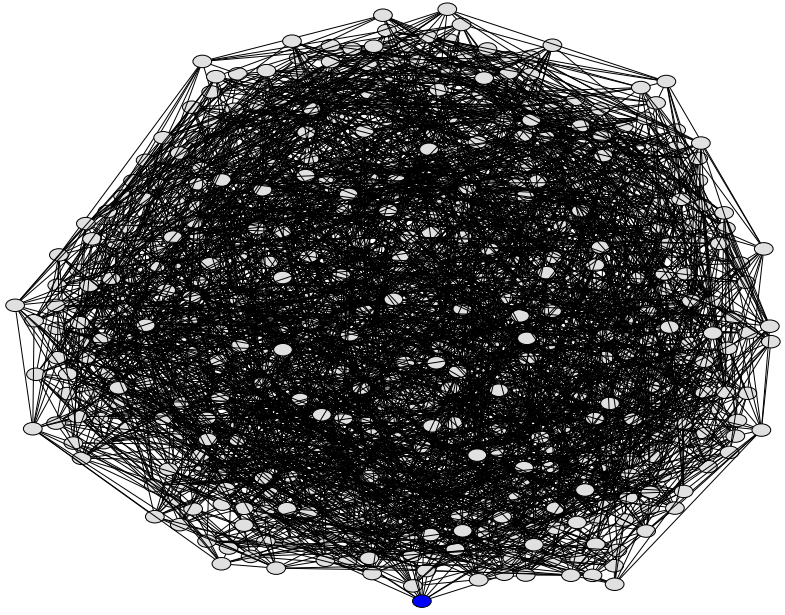
Ending on a reducible surface for CDS hash:

- Three strategies for ending on reducible surface:
  1. Ignore;
  2. Deterministically find a non-reducible neighbour;
  3. Glue 2-torsion.
- Raises possibility of failures.

Encountering reducible surfaces en route:

- Simple side-channel analysis would show that.
- Attack using meet-in-the-middle.

See [LTZ22] for more information.



- G2SIDH: Generalisation of SIDH. [FT19]
- Attacks on G2SIDH. [KTW21, GLT22]
- Failing to hash into isomorphism classes of SSECs. [BBD+22]

- G2SIDH: Generalisation of SIDH. [FT19]
- Attacks on G2SIDH. [KTW21, GLT22]
- ~~Failing to hash into isomorphism classes of SSECs. [BBD+22]~~

- G2SIDH: Generalisation of SIDH. [FT19]
- Attacks on G2SIDH. [KTW21, GLT22]
- ~~Failing to hash into isomorphism classes of SSECs. [BBD+22]~~
- Breaking SIDH using genus-two.

- ~~G2SIDH: Generalisation of SIDH. [FT19]~~
- ~~Attacks on G2SIDH. [KTW21, GLT22]~~
- ~~Failing to hash into isomorphism classes of SSECs. [BBD+22]~~
- Breaking SIDH using genus-two.

- ~~G2SIDH: Generalisation of SIDH. [FT19]~~
- ~~Attacks on G2SIDH. [KTW21, GLT22]~~
- ~~Failing to hash into isomorphism classes of SSECs. [BBD+22]~~
- Breaking SIDH using genus-two.
- SQISignHD: Embedding large isogenies over SSEC as higher dimensional isogenies.

- ~~G2SIDH: Generalisation of SIDH. [FT19]~~
- ~~Attacks on G2SIDH. [KTW21, GLT22]~~
- ~~Failing to hash into isomorphism classes of SSECs. [BBD+22]~~
- Breaking SIDH using genus-two.
- SQISignHD: Embedding large isogenies over SSEC as higher dimensional isogenies.
- FESTA: Using Kani's lemma constructively.

- Ongoing work with Sabrina Kunzweiler, Luciano Maino, Lukas Zobernig. Thanks to Giacomo Pope for help with code.

- Ongoing work with Sabrina Kunzweiler, Luciano Maino, Lukas Zobernig. Thanks to Giacomo Pope for help with code.
- Uses a trapdoor function for encryption.

- Ongoing work with Sabrina Kunzweiler, Luciano Maino, Lukas Zobernig. Thanks to Giacomo Pope for help with code.
- Uses a trapdoor function for encryption.
- Decryption function:

$$\mathrm{Dec}(sk, c) = m \text{ or } \perp$$

- Ongoing work with Sabrina Kunzweiler, Luciano Maino, Lukas Zobernig. Thanks to Giacomo Pope for help with code.
- Uses a trapdoor function for encryption.
- Decryption function:

$$\mathrm{Dec}(sk, c) = m \text{ or } \perp$$

  - Ciphertext (to be choosen by attacker) $c = E_1,\ R_1,\ S_1,\ E_2,\ R_2,\ S_2$.

- Ongoing work with Sabrina Kunzweiler, Luciano Maino, Lukas Zobernig. Thanks to Giacomo Pope for help with code.
- Uses a trapdoor function for encryption.
- Decryption function:

$$\mathrm{Dec}(sk, c) = m \text{ or } \perp$$

  - Ciphertext (to be choosen by attacker) $c = E_1, R_1, S_1, E_2, R_2, S_2$.
  - Secret key: $sk = \langle K_1 \rangle, \langle K_2 \rangle, B$
    $\langle K_1 \rangle$ and $\langle K_2 \rangle$ are secret kernel subgroups
    B is a secret 2×2 matrix.

- Ongoing work with Sabrina Kunzweiler, Luciano Maino, Lukas Zobernig. Thanks to Giacomo Pope for help with code.
- Uses a trapdoor function for encryption.
- Decryption function:

$$\mathrm{Dec}(sk, c) = m \text{ or } \perp$$

  - Ciphertext (to be choosen by attacker) $c = E_1, R_1, S_1, E_2, R_2, S_2$.
  - Secret key: $sk = \langle K_1 \rangle, \langle K_2 \rangle, B$
    $\langle K_1 \rangle$ and $\langle K_2 \rangle$ are secret kernel subgroups
    B is a secret 2×2 matrix.
  - Output: m or $\perp$

- Ongoing work with Sabrina Kunzweiler, Luciano Maino, Lukas Zobernig. Thanks to Giacomo Pope for help with code.
- Uses a trapdoor function for encryption.
- Decryption function:

$$\mathrm{Dec}(sk, c) = m \text{ or } \perp$$

  - Ciphertext (to be choosen by attacker) $c = E_1, R_1, S_1, E_2, R_2, S_2$.
  - Secret key: $sk = \langle K_1 \rangle, \langle K_2 \rangle, B$
    $\langle K_1 \rangle$ and $\langle K_2 \rangle$ are secret kernel subgroups
    B is a secret 2×2 matrix.
  - Output: m or $\perp$

- Oracle model

$$O(c, m) = \begin{cases} \perp & \text{if } \mathrm{Dec}(sk, c) \neq m, \\ 1 & \text{if } \mathrm{Dec}(sk, c) = m. \end{cases}$$

- Attacker has a pair (m, c).

- Attacker has a pair (m, c).
- Recall:

$$c = E_1, R_1, S_1, E_2, R_2, S_2.$$

- Attacker has a pair (m, c).
- Recall:

$$c = E_1, R_1, S_1, E_2, R_2, S_2.$$

- Oracle computes

$$H = \left\langle \begin{pmatrix} R_1 \\ [\alpha]R_2 \end{pmatrix}, \begin{pmatrix} S_1 \\ [\beta]S_2 \end{pmatrix} \right\rangle \subset E_1 \times E_2.$$

- Attacker has a pair (m, c).
- Recall:

$$c = E_1, R_1, S_1, E_2, R_2, S_2.$$

- Oracle computes

$$H = \left\langle \begin{pmatrix} R_1 \\ [\alpha]R_2 \end{pmatrix}, \begin{pmatrix} S_1 \\ [\beta]S_2 \end{pmatrix} \right\rangle \subset E_1 \times E_2.$$

- Note that H has many generators:

$$H = \left\langle \begin{pmatrix} R_1 + S_1 \\ [\alpha]R_2 + [\beta]S_2 \end{pmatrix}, \begin{pmatrix} [\lambda]S_1 \\ [\lambda][\beta]S_2 \end{pmatrix} \right\rangle \subset E_1 \times E_2.$$

Oracle model

$$O(c, m) = \begin{cases} 0 & \text{if } \mathrm{Dec}(sk, c) \neq m, \\ 1 & \text{if } \mathrm{Dec}(sk, c) = m. \end{cases}$$

Oracle model

$$O(c, m) = \begin{cases} 0 & \text{if } \mathrm{Dec}(sk, c) \neq m, \\ 1 & \text{if } \mathrm{Dec}(sk, c) = m. \end{cases}$$

Attacker chooses c:

$$R_1' = [1 + 2^{n-2}]R_1, \quad R_2' = [1 + 2^{n-2}]R_2,$$
$$S_1' = S_1 - [2^{n-2}]R_1, \quad S_2' = S_2 - [2^{n-2}]R_2.$$

Oracle model

$$O(c, m) = \begin{cases} 0 & \text{if } \mathrm{Dec}(sk, c) \neq m, \\ 1 & \text{if } \mathrm{Dec}(sk, c) = m. \end{cases}$$

Oracle computes

Attacker chooses c:

$$R_1' = [1 + 2^{n-2}]R_1, \quad R_2' = [1 + 2^{n-2}]R_2,$$
$$S_1' = S_1 - [2^{n-2}]R_1, \quad S_2' = S_2 - [2^{n-2}]R_2.$$

Oracle model

$$O(c, m) = \begin{cases} 0 & \text{if } \mathrm{Dec}(sk, c) \neq m, \\ 1 & \text{if } \mathrm{Dec}(sk, c) = m. \end{cases}$$

Oracle computes

$$\left\langle \begin{pmatrix} [1+2^{n-2}]R_1 \\ [\alpha][1+2^{n-2}]R_2 \end{pmatrix}, \begin{pmatrix} S_1 - [2^{n-2}]R_1 \\ [\beta](S_2 - [2^{n-2}]R_2) \end{pmatrix} \right\rangle$$

Attacker chooses c:

$$R_1' = [1+2^{n-2}]R_1, \quad R_2' = [1+2^{n-2}]R_2,$$
$$S_1' = S_1 - [2^{n-2}]R_1, \quad S_2' = S_2 - [2^{n-2}]R_2.$$

Oracle model

$$O(c,m)=\begin{cases}0 & \text{if } \mathrm{Dec}(sk,c)\neq m,\\ 1 & \text{if } \mathrm{Dec}(sk,c)=m.\end{cases}$$

Attacker chooses c:

$$R_1'=[1+2^{n-2}]R_1,\quad R_2'=[1+2^{n-2}]R_2,$$
$$S_1'=S_1-[2^{n-2}]R_1,\quad S_2'=S_2-[2^{n-2}]R_2.$$

Oracle computes

$$\left\langle\begin{pmatrix}[1+2^{n-2}]R_1\\ [\alpha][1+2^{n-2}]R_2\end{pmatrix},\begin{pmatrix}S_1-[2^{n-2}]R_1\\ [\beta](S_2-[2^{n-2}]R_2)\end{pmatrix}\right\rangle$$
$$=\left\langle\begin{pmatrix}[1+2^{n-2}]R_1\\ [\alpha]R_2+[\alpha_0][2^{n-2}]R_2+[\alpha_1][2^{n-1}]R_2\end{pmatrix},\begin{pmatrix}S_1-[2^{n-2}]R_1\\ [\beta]S_2-[\beta_0][2^{n-2}]R_2+[\beta_1][2^{n-1}]R_2\end{pmatrix}\right\rangle$$

Oracle model

$$O(c,m)=\begin{cases}0 & \text{if } \mathrm{Dec}(sk,c)\neq m,\\ 1 & \text{if } \mathrm{Dec}(sk,c)=m.\end{cases}$$

Attacker chooses c:

$$R_1'=[1+2^{n-2}]R_1,\quad R_2'=[1+2^{n-2}]R_2,$$
$$S_1'=S_1-[2^{n-2}]R_1,\quad S_2'=S_2-[2^{n-2}]R_2.$$

Oracle computes

$$\left\langle\begin{pmatrix}[1+2^{n-2}]R_1\\ [\alpha][1+2^{n-2}]R_2\end{pmatrix},\begin{pmatrix}S_1-[2^{n-2}]R_1\\ [\beta](S_2-[2^{n-2}]R_2)\end{pmatrix}\right\rangle$$
$$=\left\langle\begin{pmatrix}[1+2^{n-2}]R_1\\ [\alpha]R_2+[\alpha_0][2^{n-2}]R_2+[\alpha_1][2^{n-1}]R_2\end{pmatrix},\begin{pmatrix}S_1-[2^{n-2}]R_1\\ [\beta]S_2-[\beta_0][2^{n-2}]R_2+[\beta_1][2^{n-1}]R_2\end{pmatrix}\right\rangle$$
$$=\begin{cases}\left\langle\begin{pmatrix}R_1+S_1\\ [\alpha]R_2+[\beta]S_2\end{pmatrix},\begin{pmatrix}S_1-[2^{n-2}]R_1\\ [\beta](S_2-[2^{n-2}]R_2)\end{pmatrix}\right\rangle & \text{if } (\alpha_1,\beta_1)=(0,0)\\ \left\langle\begin{pmatrix}R_1+S_1\\ [\alpha]R_2+[\beta]S_2+[2^{n-1}]R_2\end{pmatrix},\begin{pmatrix}S_1-[2^{n-2}]R_1\\ [\beta](S_2-[2^{n-2}]R_2)\end{pmatrix}\right\rangle & \text{if } (\alpha_1,\beta_1)=(0,1)\\ \left\langle\begin{pmatrix}R_1+S_1\\ [\alpha]R_2+[\beta]S_2+[2^{n-1}]R_2\end{pmatrix},\begin{pmatrix}S_1-[2^{n-2}]R_1\\ [\beta](S_2-[2^{n-2}]R_2)\end{pmatrix}\right\rangle & \text{if } (\alpha_1,\beta_1)=(1,0)\\ \left\langle\begin{pmatrix}R_1+S_1\\ [\alpha]R_2+[\beta]S_2\end{pmatrix},\begin{pmatrix}S_1-[2^{n-2}]R_1\\ [\beta](S_2-[2^{n-2}]R_2)\end{pmatrix}\right\rangle & \text{if } (\alpha_1,\beta_1)=(1,1)\end{cases}$$

- Decryption protected by OAEP.

- Decryption protected by OAEP.
- Recovered one-bit despite that.

- Decryption protected by OAEP.
- Recovered one-bit despite that.
- Recovery of next bit is underway, but thwarted by OAEP so far.

- Decryption protected by OAEP.
- Recovered one-bit despite that.
- Recovery of next bit is underway, but thwarted by OAEP so far.
- *SHOULD* be able to attack trapdoor without OAEP; trapdoor model does not consider this.

- Decryption protected by OAEP.
- Recovered one-bit despite that.
- Recovery of next bit is underway, but thwarted by OAEP so far.
- *SHOULD* be able to attack trapdoor without OAEP; trapdoor model does not consider this.
- Attacking a variant of FESTA with diagonal matrices.

- Decryption protected by OAEP.
- Recovered one-bit despite that.
- Recovery of next bit is underway, but thwarted by OAEP so far.
- *SHOULD* be able to attack trapdoor without OAEP; trapdoor model does not consider this.
- Attacking a variant of FESTA with diagonal matrices.
- Attacking trapdoor can be made to attack encryption by adding in side-channel information.

- Generalising CSIDH to genus-two.
  - Better implementations of isogenies on abelian surfaces.
  - Better understanding of endomorphism rings of abelian surfaces.
  - Greater understanding on isogeny graph in genus-two.

- Generalising CSIDH to genus-two.
  - Better implementations of isogenies on abelian surfaces.
  - Better understanding of endomorphism rings of abelian surfaces.
  - Greater understanding on isogeny graph in genus-two.
- Generalising SQISign to genus-two.
  - Generalise KLPT to genus-two.
  - Computing Deuring correspondence in genus-two.

- Generalising CSIDH to genus-two.
  - Better implementations of isogenies on abelian surfaces.
  - Better understanding of endomorphism rings of abelian surfaces.
  - Greater understanding on isogeny graph in genus-two.
- Generalising SQISign to genus-two.
  - Generalise KLPT to genus-two.
  - Computing Deuring correspondence in genus-two.

WARNING: Cryptosystems probably won't be efficient!

- Generalising CSIDH to genus-two.
  - Better implementations of isogenies on abelian surfaces.
  - Better understanding of endomorphism rings of abelian surfaces.
  - Greater understanding on isogeny graph in genus-two.
- Generalising SQISign to genus-two.
  - Generalise KLPT to genus-two.
  - Computing Deuring correspondence in genus-two.

WARNING: Cryptosystems probably won't be efficient! But can be fun!

- Arithmétique et Géométrie
  - (Hyper)elliptic curves and jacobians
  - Isotropic subgroups
- Cryptographie
  - Hash functions, collisions, and patch
  - Constructive and destructive applications
  - FESTA and one-bit recovery
- Théorie des Codes

- Arithmétique et Géométrie
  - (Hyper)elliptic curves and jacobians
  - Isotropic subgroups
- Cryptographie
  - Hash functions, collisions, and patch
  - Constructive and destructive applications
  - FESTA and one-bit recovery
- Théorie des Codes
  - Sorry

- Arithmétique et Géométrie
  - (Hyper)elliptic curves and jacobians
  - Isotropic subgroups
- Cryptographie
  - Hash functions, collisions, and patch
  - Constructive and destructive applications
  - FESTA and one-bit recovery
- Théorie des Codes
  - Sorry

Thank you and questions?


Jeremy Booher, Ross Bowden, Javad Doliskani, Tako Boris Fouotsa, Steven D. Galbraith, Sabrina Kunzweiler, Simon-Philipp Merz, Christophe Petit, Benjamin Smith, Katherine E. Stange, Yan Bo Ti, Christelle Vincent, José Felipe Voloch, Charlotte Weitkämper, and Lukas Zobernig.

Failing to hash into supersingular isogeny graphs.

IACR Cryptol. ePrint Arch., page 518, 2022.


Wouter Castryck, Thomas Decru, and Benjamin Smith.

Hash functions from superspecial genus-2 curves using richelot isogenies.

J. Math. Cryptol., 14(1):268–292, 2020.


Denis Charles, Eyal Goren, and Kristin Lauter.

Families of ramanujan graphs and quaternion algebras.

Groups and symmetries, CRM Proc. Lecture Notes, 47:53–80, 2009.


Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren.

Cryptographic hash functions from expander graphs.

J. Cryptology, 22(1):93–113, 2009.


Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes.
CSIDH: an efficient post-quantum commutative group action.
In Thomas Peyrin and Steven D. Galbraith, editors, *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*, volume 11274 of *Lecture Notes in Computer Science*, pages 395–427. Springer, 2018.


Craig Costello and Benjamin Smith.
The supersingular isogeny problem in genus 2 and beyond.
In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings*, volume 12100 of *Lecture Notes in Computer Science*, pages 151–168. Springer, 2020.


Christina Delfs and Steven D. Galbraith.
Computing isogenies between supersingular elliptic curves over $\mathfrak{O}_\mathfrak{p}$.
Des. Codes Cryptogr., 78(2):425–440, 2016.


E. Victor Flynn and Yan Bo Ti.

Genus two isogeny cryptography.

In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers*, volume 11505 of *Lecture Notes in Computer Science*, pages 286–306. Springer, 2019.


Ariana Goh, Chu-Wee Lim, and Yan Bo Ti.

Generalising fault attacks to genus two isogeny cryptosystems.

In *Workshop on Fault Detection and Tolerance in Cryptography, FDTC 2022, Virtual Event / Italy, September 16, 2022*, pages 38–49. IEEE, 2022.


Bruce W. Jordan and Yevgeny Zaytman.

Isogeny graphs of superspecial abelian varieties and generalized brandt matrices.

arXiv: Number Theory, 2020.


Sabrina Kunzweiler, Yan Bo Ti, and Charlotte Weitkämper.
Secret keys in genus-2 SIDH.
In Riham AlTawy and Andreas Hülsing, editors, *Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers*, volume 13203 of *Lecture Notes in Computer Science*, pages 483–507. Springer, 2021.


Jason T. LeGrow, Yan Bo Ti, and Lukas Zobernig.
Supersingular non-superspecial abelian surfaces in cryptography.
IACR Cryptol. ePrint Arch., page 650, 2022.


Katsuyuki Takashima.
Efficient Algorithms for Isogeny Sequences and Their Cryptographic Applications, pages 97–114. 2018.