The supersingular isogeny problem in genus 2 and beyond

Craig Costello and **Benjamin Smith**

ANR CIAO Kickoff meeting, Bordeaux, February 2020

Microsoft Research and **Inria** + **École polytechnique**

$$g = 1$$

The supersingular isogeny graph

For each prime p, we let $S_1(p)$ be the set of **supersingular elliptic curves** over $\mathbb{F}_{p^2}$, up to $\mathbb{F}_{p^2}$-isomorphism:

$$\#S_1(p) \approx \lfloor p/12 \rfloor ;$$

we can view $S_1(p) \subset \mathbb{F}_{p^2}$ via the j-invariant.

For primes $\ell \neq p$, we let $\Gamma_1(\ell; p)$ be the **ℓ-isogeny graph** on $S_1(p)$. This is

- A directed multigraph (but almost a graph)
- Connected
- $(\ell + 1)$-regular
- Ramanujan (excellent expansion properties)

Random walks in $\Gamma_1(\ell; p)$ of length $O(\log p)$ give a uniform distribution on $S_1(p)$.

Supersingular isogeny problem

The general supersingular elliptic **isogeny problem** for fixed ℓ:

Given $\mathcal{E}$ and $\mathcal{E}'$ in $S_1(p)$, find a path from $\mathcal{E}$ to $\mathcal{E}'$ in $\Gamma_1(\ell; p)$

classical solution in $O(\sqrt{\#S_1(p)}) = O(\sqrt{p})$

quantum solution in $O(\sqrt[4]{\#S_1(p)}) = O(\sqrt[4]{p})$

This **general** problem (our focus today) is related to the security of the Charles–Goren–Lauter hash function.

SIDH security is related to the special problem of finding very ***short paths*** *(length $< \log p$. Solving the general problem has important implications for this short-path problem (not in this talk).*

The Charles–Goren–Lauter hash function

Charles–Goren–Lauter (2009): a hash function with provable collision-resistance properties. System parameters:

- A prime p, an ordering on $\mathbb{F}_{p^2}$ (hence on $S_1(p)$), and a linear map $\pi : \mathbb{F}_{p^2} \to \mathbb{F}_p$
- An edge $j_{-1} \to j_0$ in $\Gamma_1(2; p)$

To compute the hash of an n-bit message $m = (m_0, \ldots, m_{n-1})$,
we compute a corresponding path $j_0 \to \cdots \to j_n$ in $\Gamma_1(\ell; p)$: for each $0 \leq i < n$,

1. the 3 edges out of j_i are $j_i \to j_{i-1}$, $j_i \to \alpha$, and $j_i \to \beta$ with $\alpha > \beta$
2. if $m_i = 0$, then set $j_{i+1} = \alpha$; otherwise, set $j_{i+1} = \beta$

The hash value is $H(m) = \pi(j_n)$.

Solving the **isogeny problem** for $\ell = 2 \implies$ finding preimages for this hash.

$g > 1$

Higher dimensions: superspecial and supersingular

A g-dimensional PPAV $\mathcal{A}$ is

Super*singular* if all slopes of the Newton polygon of its Frobenius are 1/2.
Any supersingular $\mathcal{A}$ is **isogenous** to a product of supersingular ECs.

Super*special* if Frobenius acts as 0 on $H^1(\mathcal{A}, \mathcal{O}_\mathcal{A})$.
Any superspecial $\mathcal{A}$ is **isomorphic** to a product of supersingular ECs, though generally only as unpolarized AVs.

- **Superspecial** $\implies$ **supersingular**.
- Superspeciality is preserved by $(\ell, \ldots, \ell)$-isogeny.

The superspecial set

For each $g > 0$ and prime p, we define

$$S_g(p) := \{\text{superspecial PPAVs over } \mathbb{F}_{p^2}\} /\cong .$$

We have

$$\#S_g(p) = O(p^{g(g+1)/2})$$

(with much more precise statements for $g \leq 3$).

The superspecial graph

For primes $\ell \neq p$, we let $\Gamma_g(\ell; p)$ be the $(\ell, \ldots, \ell)$-isogeny graph on $S_g(p)$.

The graph $\Gamma_g(\ell; p)$ is connected and $N_g(\ell)$-regular, where

$$N_g(\ell) := \sum_{d=0}^{g} \begin{bmatrix} g \\ d \end{bmatrix}_\ell \cdot \ell^{\binom{g-d+1}{2}}$$

where $\begin{bmatrix} n \\ k \end{bmatrix}_\ell := \frac{(n)_\ell \cdots (n-k+1)_\ell}{(k)_\ell \cdots (1)_\ell}$, where $(i)_\ell := \frac{\ell^i - 1}{\ell - 1}$ counts the k-diml subspaces of $\mathbb{F}_\ell^n$.

Expander hypothesis: we assume $\Gamma_g(\ell; p)$ is Ramanujan.

If the hypothesis fails, then our algorithm might be less efficient,
but commensurately so with the cryptosystems that it attacks.

Generalizing CGL to genus 2: Takashima

Takashima was the first to generalize CGL to AVs of dimension $g = 2$.

Takashima's hash works exactly like CGL, but

- $S_1(p)$ becomes $S_2(p)$ (Takashima wants to use the full supersingular graph, but ends up stuck in the superspecial component)
- $\Gamma_1(2; p)$ becomes $\Gamma_2(2; p)$: i.e. 2-isogenies become $(2, 2)$-isogenies,

To compute the walks in $\Gamma_2(2; p)$, Takashima uses

- supersingular **genus-2 curves** to represent the vertices (with the j-invariant becomes the Igusa–Clebsch invariants), and
- **Richelot's formulæ** to compute the isogeny steps

Note that $\Gamma_1(2; p)$ is 15-regular, so the data to be hashed is coded in base ≤ 14!

Trivial 4-cycles in the genus-2 graph

Flynn and Ti observe a serious issue with Takashima's hash function:
It is easy to construct **cycles of length 4** starting at any vertex of $\Gamma_2(\ell; p)$.

Take $P \in \mathcal{A}_0[\ell^2]$, $Q, R \in \mathcal{A}_0[\ell]$ s.t. $e_\ell([\ell]P, R) = e_\ell([\ell]P, Q) = 1$; form (ℓ,ℓ)-isogenies

$$\begin{aligned} \phi_0 : \mathcal{A}_0 &\longrightarrow \mathcal{A}_1 = \mathcal{A}_0/K_0 && \text{where } K_0 := \langle [\ell]P, Q \rangle \\ \phi_0' : \mathcal{A}_0 &\longrightarrow \mathcal{A}_1' = \mathcal{A}_0/K_0' && \text{where } K_0' := \langle [\ell]P, Q \rangle \\ \phi_1 : \mathcal{A}_1 &\longrightarrow \mathcal{A}_2 = \mathcal{A}_1/K_1 && \text{where } K_1 := \phi_0(K_0') \\ \phi_1' : \mathcal{A}_1 &\longrightarrow \mathcal{A}_2' = \mathcal{A}_1/K_1' && \text{where } K_1' := \phi_0'(K_0) \end{aligned}$$

Now $\ker(\phi_1 \circ \phi_0) = \ker(\phi_1' \circ \phi_0')$, so $\mathcal{A}_2 \cong \mathcal{A}_2'$, and so we get a cycle

$$\mathcal{A}_0 \xrightarrow{\phi_0} \mathcal{A}_1 \xrightarrow{\phi_1} \mathcal{A}_2 \cong \mathcal{A}_2' \xrightarrow{(\phi_1')^\dagger} \mathcal{A}_1' \xrightarrow{(\phi_0')^\dagger} \mathcal{A}_0 .$$

$\implies$ in $g > 1$, **non-backtracking is not strong enough** to avoid hash collisions.

Generalizing CGL to genus 2: Castryck–Decru–Smith

Castryck–Decru–S. (Nutmic 2019): an attempt to fix Takashima.

- Explicitly restriction to the superspecial graph $\Gamma_2(2;p)$
- New rule for isogeny walks to replace non-backtracking:
  for each $(2,2)$-isogeny $\phi_i : \mathcal{A}_i \to \mathcal{A}_{i+1}$, we must choose one of the **eight** $(2,2)$-isogenies $\phi_{i+1} : \mathcal{A}_{i+1} \to \mathcal{A}_{i+2}$ such that $\phi_{i+1} \circ \phi_i$ is a $(4,4)$-isogeny.

Implementation: again, represent vertices with (Jacobians of) genus-2 curves, and compute edges using Richelot isogenies.

The superspecial genus 2 graph

Minor inconvenience: there are *two types* of PPAVs in dimension $g = 2$: **Jacobians** of genus-2 curves, and **elliptic products**.

- Isomorphism invariants are incompatible
- Richelot's formulæ break down when the codomain is an elliptic product

Partition $S_2(p)$ into corresponding subsets, $S_2(p)^J$ and $S_2(p)^E$; then

$$\#S_2(p)^J = \frac{1}{2880}p^3 + \frac{1}{120}p^2 \qquad \text{and} \qquad \#S_2(p)^E = \frac{1}{288}p^2 + O(p)\,.$$

Being a proof of concept, CDS takes a simple solution: *fail on elliptic products*. Justification: a random $\mathcal{A} \in S_2(p)$ has only a $O(1/p)$ chance of being in $S_2(p)^E$.

The superspecial genus 2 graph

Minor inconvenience: there are *two types* of PPAVs in dimension $g = 2$: **Jacobians** of genus-2 curves, and **elliptic products**.

- Isomorphism invariants are incompatible
- Richelot's formulæ break down when the codomain is an elliptic product

Partition $S_2(p)$ into corresponding subsets, $S_2(p)^J$ and $S_2(p)^E$; then

$$\#S_2(p)^J = \frac{1}{2880}p^3 + \frac{1}{120}p^2 \qquad \text{and} \qquad \#S_2(p)^E = \frac{1}{288}p^2 + O(p)\,.$$

Being a proof of concept, CDS takes a simple solution: *fail on elliptic products*. Justification: a random $\mathcal{A} \in S_2(p)$ has only a $O(1/p)$ chance of being in $S_2(p)^E$.

Bad news: from a cryptanalytic point of view, **this is not rare enough**.

Solving the isogeny problem in $g > 1$

Results

Theorem (Costello–S., PQCrypto 2020):

1. There exists a **classical algorithm** which solves isogeny problems in $\Gamma_g(\ell; p)$ with probability $\geq 1/2^{g-1}$ in expected time $\widetilde{O}((p^{g-1}/P))$ on P processors as $p \to \infty$ (with ℓ fixed).
2. There exists a **quantum algorithm** which solves isogeny problems in $\Gamma_g(\ell; p)$ in expected time $\widetilde{O}(\sqrt{p^{g-1}})$ as $p \to \infty$ (with ℓ fixed).

This talk: the classical algorithm.

Details: `https://eprint.iacr.org/2019/1387`

Attacking the isogeny problem

Recall: if we just view $\Gamma_g(\ell; p)$ as a generic $N_g(\ell)$-regular Ramanujan graph, then solving the path-finding problem would cost $O(p^{g(g+1)/4})$ (classical) isogeny steps.

Key observation: in $g = 2$, we have $\#S_2(p)^E > \sqrt{\#S_2(p)^J}$. This pattern continues in $g > 2$. We beat square-root algorithms by exploiting this special subset.

Let's look at the algorithm for $g = 2$ first. Recursive application will give us $g > 2$.

The algorithm in $g = 2$: Step 1

The algorithm in dimension $g = 2$ (attacking Takashima and Castryck–Decru–S.):

The algorithm in $g = 2$: Step 1

The algorithm in dimension $g = 2$ (attacking Takashima and Castryck–Decru–S.):

Step 1: Compute paths from our target PPASes into elliptic product vertices:

$$\phi : \mathcal{A} \to \cdots \to \mathcal{E}_1 \times \mathcal{E}_2 \in S_2(p)^E$$
$$\phi' : \mathcal{A}' \to \cdots \to \mathcal{E}_1' \times \mathcal{E}_2' \in S_2(p)^E$$

Expander hypothesis $\implies$ we find ϕ (and ϕ') after $O(p)$ random walks of length in $O(\log p)$: total cost is $\widetilde{O}(p/P)$ isogeny steps on P classical processors.

It remains to compute a path $\mathcal{E}_1 \times \mathcal{E}_2 \to \cdots \to \mathcal{E}_1' \times \mathcal{E}_2'$ in $\Gamma_2(\ell; p)$ in $\widetilde{O}(p)$ steps.

The algorithm in $g = 2$: Step 2

Step 2: to compute a path $\mathcal{E}_1 \times \mathcal{E}_2 \to \cdots \to \mathcal{E}_1' \times \mathcal{E}_2'$ in $\Gamma_2(\ell; p)$,

1. Compute paths $\psi_1 : \mathcal{E}_1 \to \cdots \to \mathcal{E}_1'$ and $\psi_2 : \mathcal{E}_2 \to \cdots \to \mathcal{E}_2'$ in $\Gamma_1(\ell; p)$.

The algorithm in $g = 2$: Step 2

Step 2: to compute a path $\mathcal{E}_1 \times \mathcal{E}_2 \to \cdots \to \mathcal{E}_1' \times \mathcal{E}_2'$ in $\Gamma_2(\ell; p)$,

1. Compute paths $\psi_1 : \mathcal{E}_1 \to \cdots \to \mathcal{E}_1'$ and $\psi_2 : \mathcal{E}_2 \to \cdots \to \mathcal{E}_2'$ in $\Gamma_1(\ell; p)$.
2. If $\text{length}(\psi_1) \not\equiv \text{length}(\psi_2) \pmod 2$, then go back to Step 1 (or swap $\mathcal{E}_1 \leftrightarrow \mathcal{E}_2$).
3. Trivially **stretch** the shorter of the ψ_i to the same length as the other, by stepping back and forth on the last component isogeny.

The algorithm in $g = 2$: Step 2

Step 2: to compute a path $\mathcal{E}_1 \times \mathcal{E}_2 \to \cdots \to \mathcal{E}_1' \times \mathcal{E}_2'$ in $\Gamma_2(\ell; p)$,

1. Compute paths $\psi_1 : \mathcal{E}_1 \to \cdots \to \mathcal{E}_1'$ and $\psi_2 : \mathcal{E}_2 \to \cdots \to \mathcal{E}_2'$ in $\Gamma_1(\ell; p)$.
2. If $\mathrm{length}(\psi_1) \not\equiv \mathrm{length}(\psi_2) \pmod 2$, then go back to Step 1 (or swap $\mathcal{E}_1 \leftrightarrow \mathcal{E}_2$).
3. Trivially **stretch** the shorter of the ψ_i to the same length as the other, by stepping back and forth on the last component isogeny.
4. Compose the products of the i-th components of ψ_1 and ψ_2 to get a path

$$\psi^\times : \mathcal{E}_1 \times \mathcal{E}_2 \to \cdots \to \mathcal{E}_1' \times \mathcal{E}_2' \qquad \text{in } \Gamma_2(\ell; p)\,.$$

Cost: same as solving the isogeny problem in $\Gamma_1(\ell; p)$, i.e. $O(\sqrt{p}/P)$.

The composition $(\phi')^\dagger \circ \psi^\times \circ \phi$ is a path from $\mathcal{A}$ to $\mathcal{A}'$ in $\Gamma_2(\ell; p)$.

We can thus **solve the isogeny problem** in $\Gamma_2(\ell; p)$ in $\widetilde{O}(p)$ isogeny steps.

Attacking higher genus

The same idea works **in higher dimension** as follows.

Recall: $\#S_g(p) = O(p^{g(g+1)/2})$, so classical square-root algorithms solve the isogeny problem in $\Gamma_g(\ell; p)$ in $O(p^{g(g+1)/4})$ isogeny steps.

Let $T_g(p)$ be the image of $S_1(p) \times S_{g-1}(p)$ in $S_g(p)$ (product polarization).

We have $\#S_1(p) = O(p)$ and $\#S_{g-1}(p) = O(p^{g(g-1)/2})$, so

$$\#T_g(p) = O(p^{(g^2-g+2)/2})\,;$$

so the probability that a random $\mathcal{A}$ in $S_g(p)$ is in $T_g(p)$ is in $O(1/p^{(g-1)})$.

Key observation: $g - 1 < g(g+1)/4$ (and much smaller for large g).

We should be able to efficiently recognise steps into $T_g(p)$ by something analogous to the breakdown in Richelot's formulæ in $g = 2$ (theta relations?).

Solving the general isogeny problem

To find a path from $\mathcal{A}$ to $\mathcal{A}'$ in $\Gamma_g(\ell; p)$:

1. Compute paths $\phi : \mathcal{A} \to \mathcal{E} \times \mathcal{B} \in T_g(p)$ and $\phi' : \mathcal{A}' \to \mathcal{E}' \times \mathcal{B}' \in T_g(p)$ in $\Gamma_g(\ell; p)$
   Expander hypothesis $\implies \widetilde{O}(p^{g-1}/P)$ *isogeny steps. Dominant step*
2. Compute a path $\psi_E : \mathcal{E} \to \cdots \to \mathcal{E}'$ in $\Gamma_1(\ell; p)$
   Usual elliptic algorithm $\implies O(\sqrt{p}/P)$ *isogeny steps*
3. **Recurse** to compute a path $\psi_B : \mathcal{B} \to \cdots \to \mathcal{B}'$ in $\Gamma_{g-1}(\ell; p)$
   Expander hypothesis $\implies \widetilde{O}(p^{g-2}/P)$ *isogeny steps*
4. Apply the elliptic isogeny-glueing technique to get the final path.
   Probability of compatible lengths: $1/2^{g-1}$.

Total cost: $\widetilde{O}(p^{g-1}/P)$, dominated by the cost of walking into $T_g(p)$ in Step 1.
Much faster than $O(p^{g(g+1)/4})$.

Cryptographic implications

Isogeny-based hashing in $g > 1$ is **much less efficient** than the elliptic equivalent.

Question: what about SIDH analogues? The isogeny paths produced by our algorithms are **too long** to represent SIDH-type cryptosystem keys.

However, they allow us to connect target PPAVs with PPAVs with known endomorphism ring, and then KLPT-style techniques let us shorten the paths.

There is a lot of detail to work out here (good thing we have ANR CIAO).

Conclusion: supersingular isogeny-based cryptosystems in dimension $g > 1$ are **likely to be uncompetitive** with elliptic equivalents.