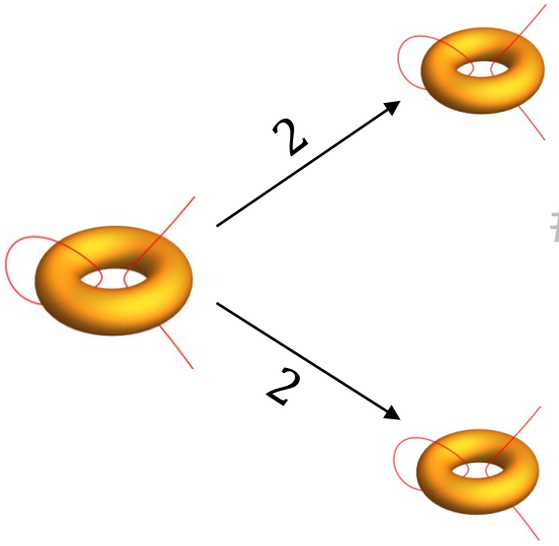
The supersingular isogeny problem in genus 2 and beyond

Craig Costello and Benjamin Smith

PQCrypto 2020

Why?

elliptic curve
2-isogenies



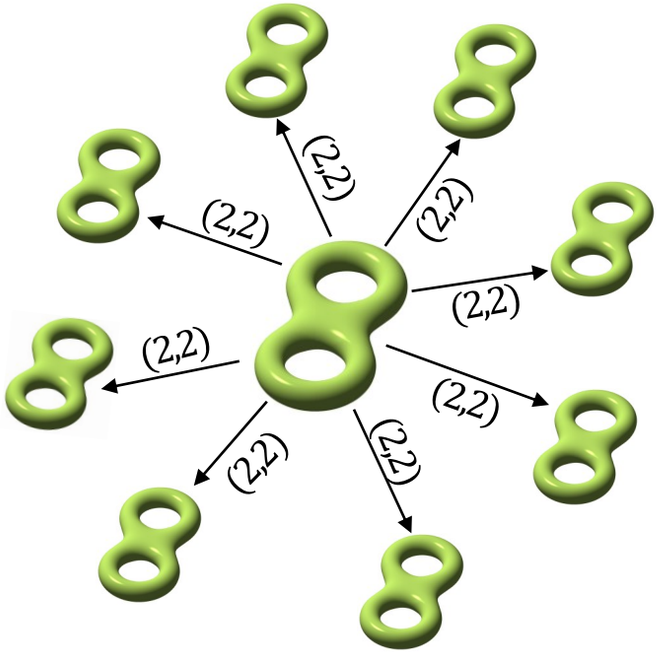
$\#E(\mathbb{F}_{p^2}) = O(p^2)$

$\#S_1(p) = O(p)$

$\#J(\mathbb{F}_{p^2}) = O(p^4)$

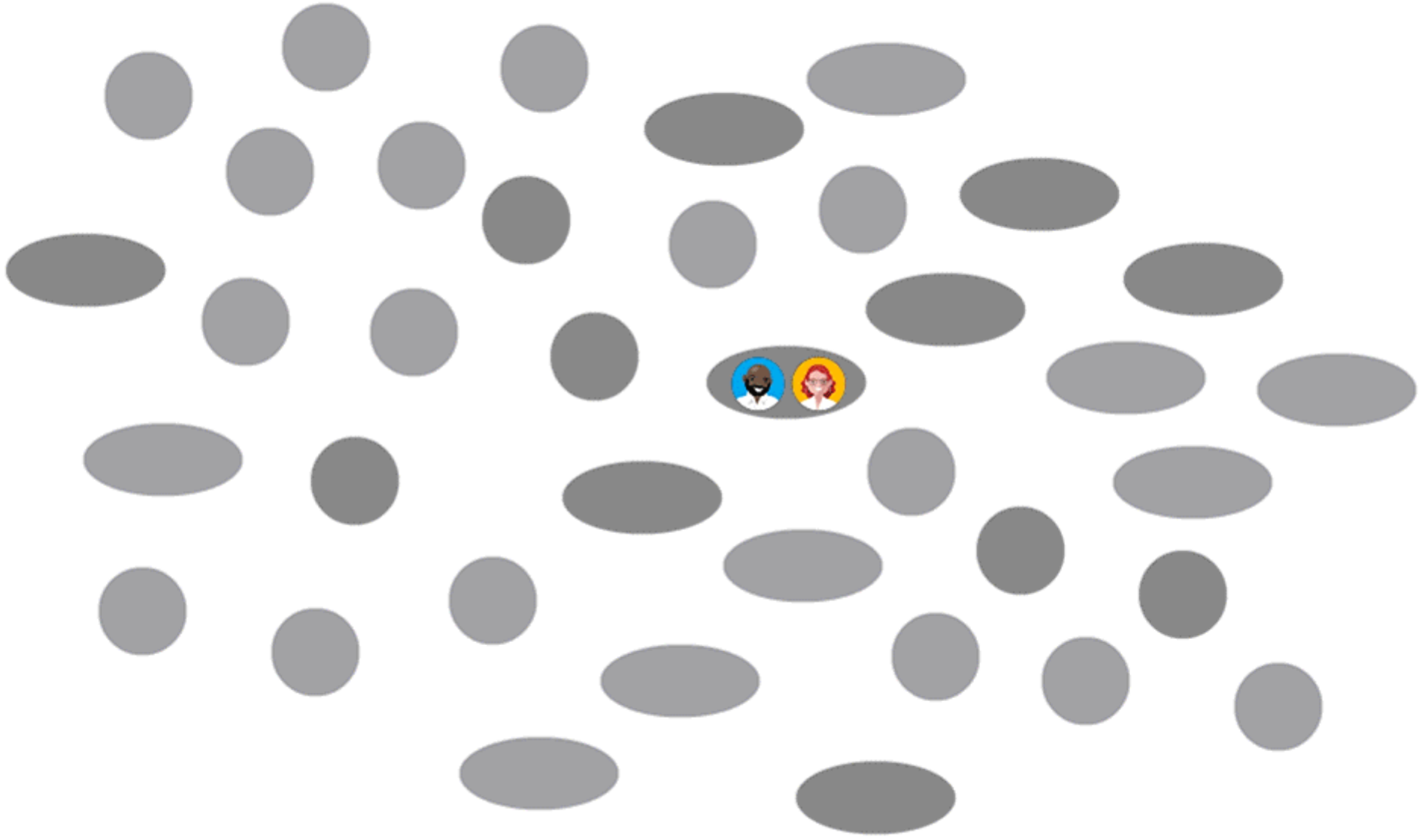
$\#S_2(p) = O(p^3)$

genus 2 hyperelliptic
(2,2)-isogenies

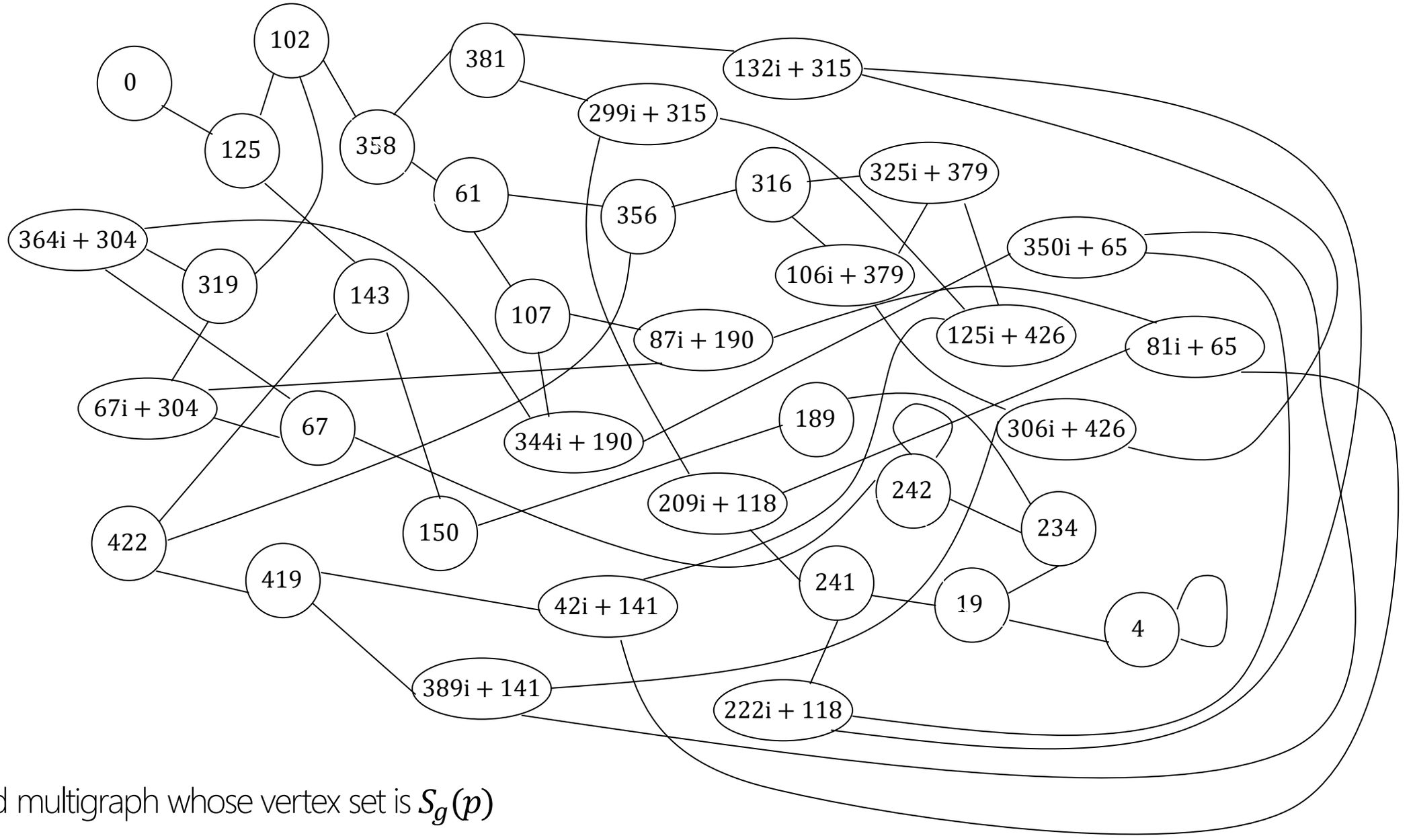


- genus 2: fields of one third the bitlength
- genus g: $\#S_g(p) = O\left(p^{g(g+1)/2}\right)$

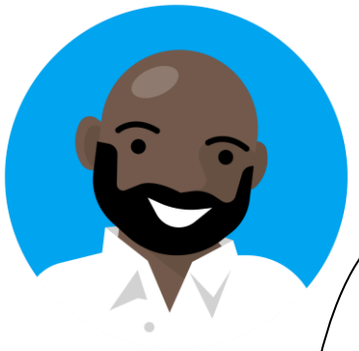
Example isogeny-based protocol: SIDH



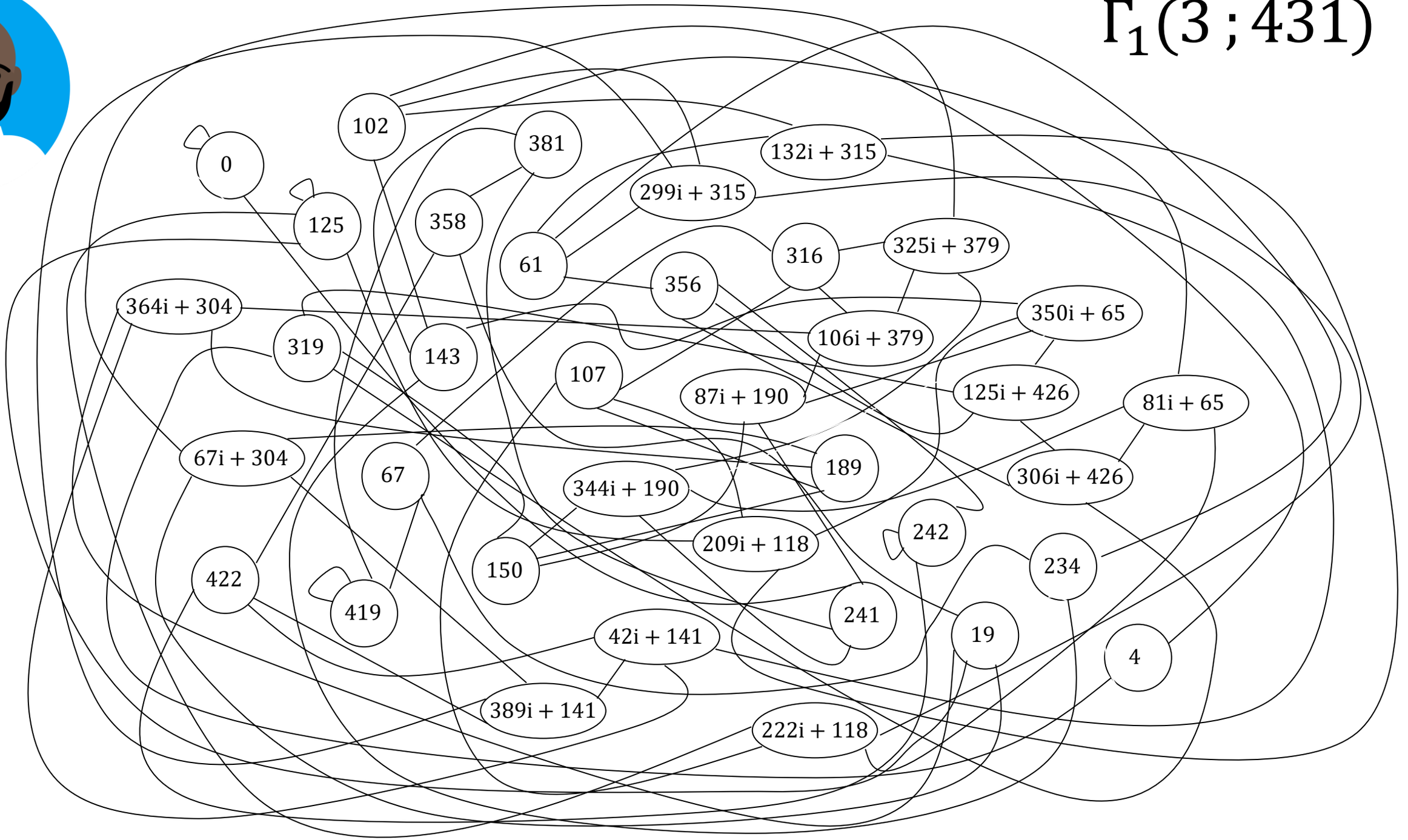
$$\Gamma_g(\ell\,;p) = \Gamma_1(2\,;431)$$



$\Gamma_g(\ell\,;p)$: directed multigraph whose vertex set is $S_g(p)$



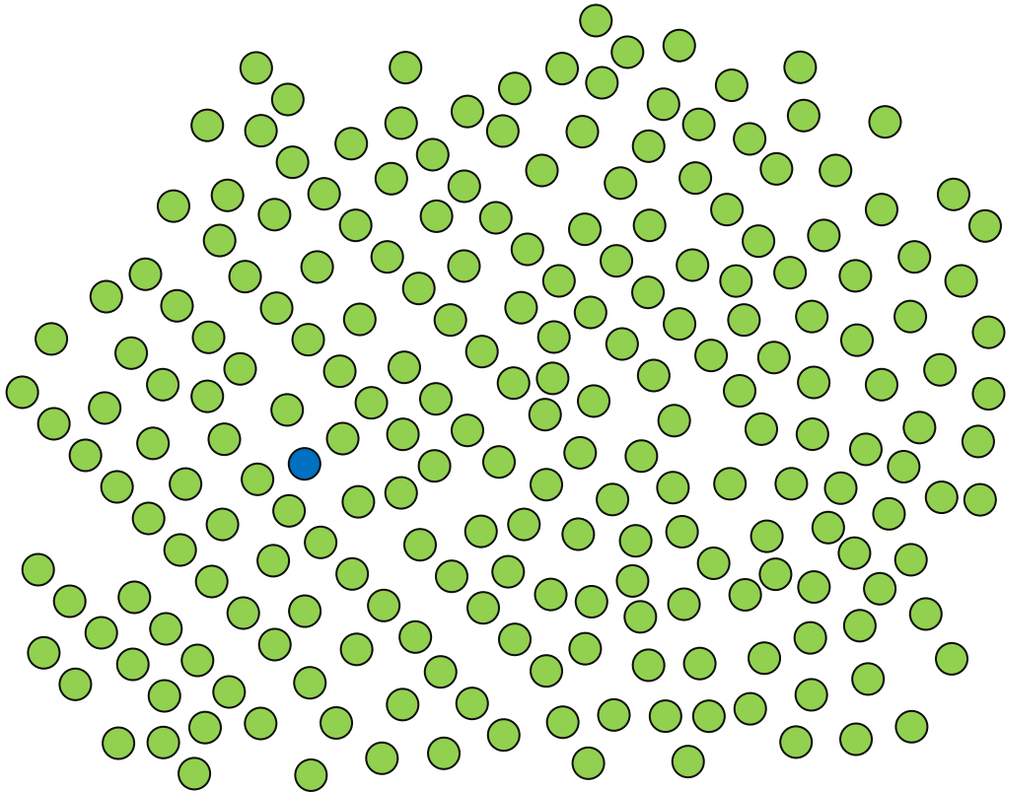
$\Gamma_1(3;431)$


0
102
381
132i + 315
299i + 315
125
358
61
316
325i + 379
356
364i + 304
350i + 65
106i + 379
319
143
107
87i + 190
125i + 426
81i + 65
67i + 304
67
189
306i + 426
344i + 190
209i + 118
242
150
234
422
419
241
19
42i + 141
4
389i + 141
222i + 118

Two types of supersingular problems in $\Gamma_g(\ell\,;p)$

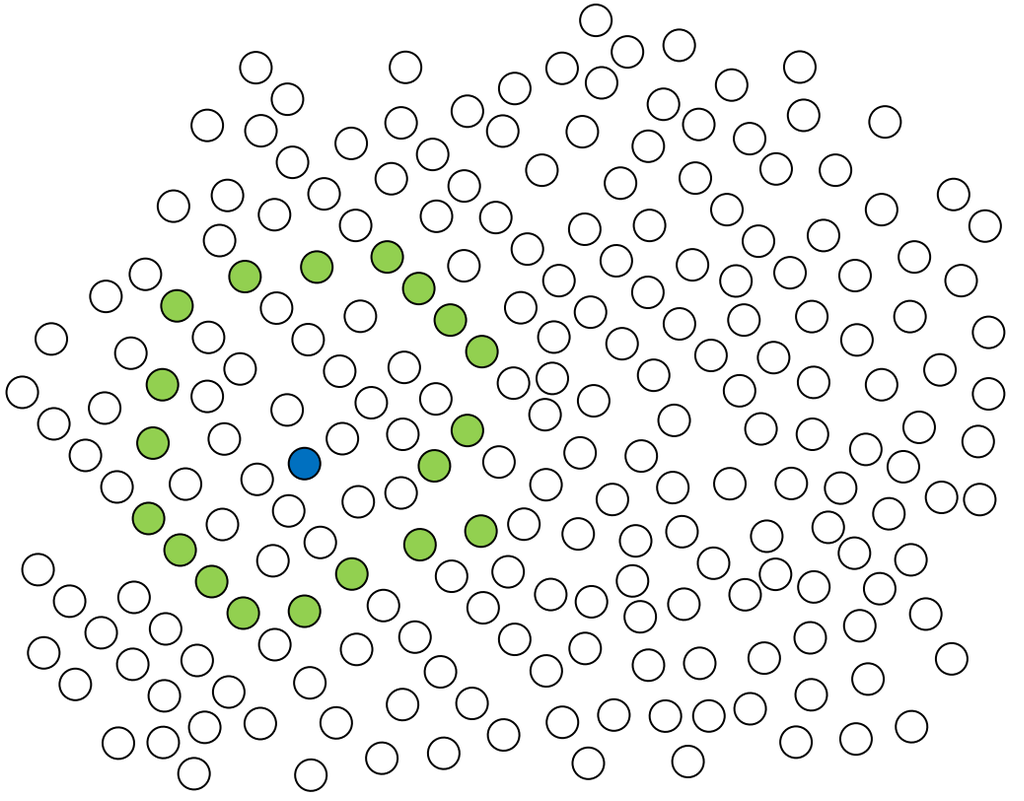
$$\phi: \quad E \to E'$$

General: E' can be anywhere



e.g. hash functions, signatures…

Short walks: E' can be in $\sqrt{}$-sized subset

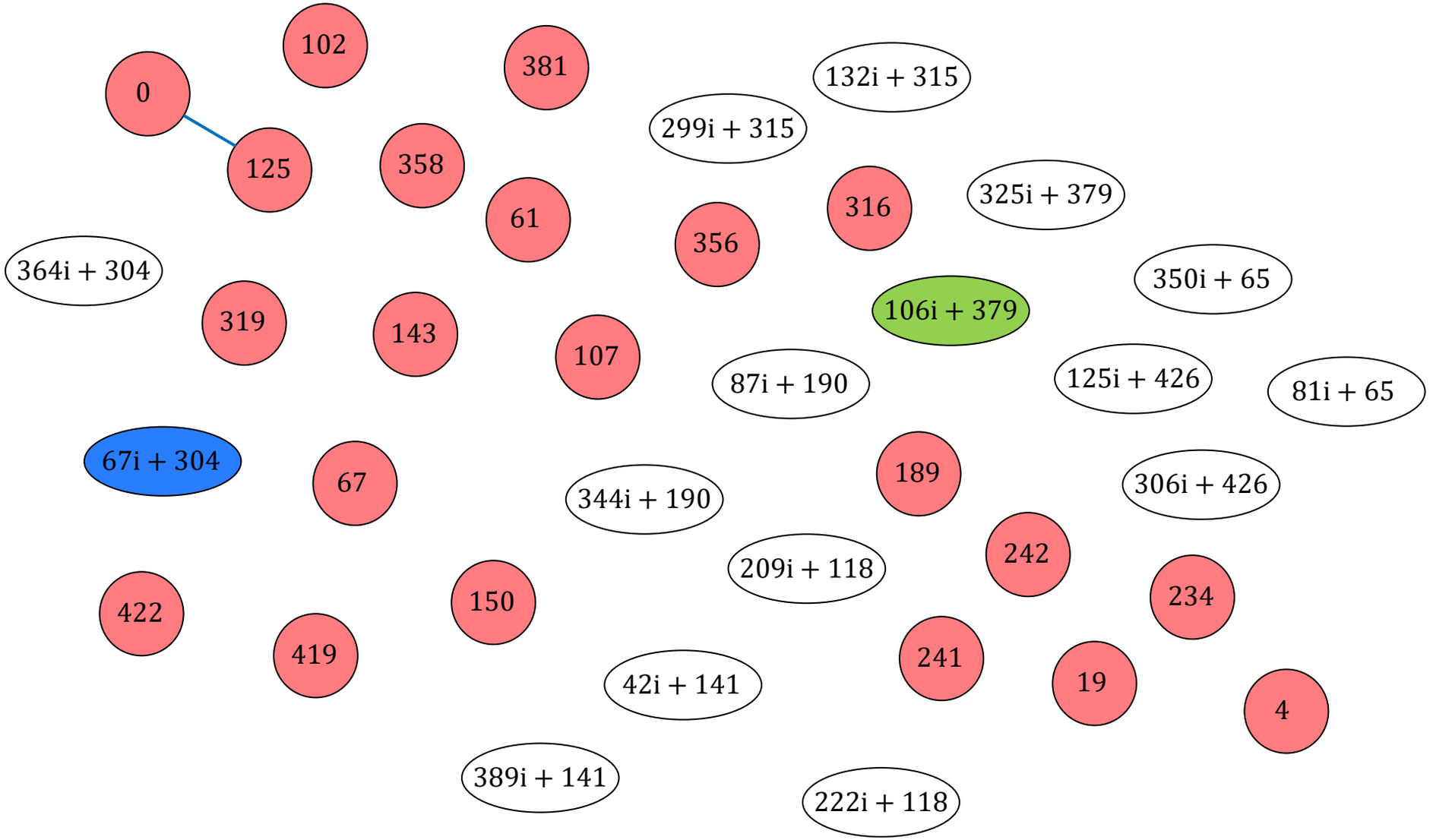


e.g. SIDH, some signatures…

The Delfs-Galbraith algorithm for the general problem

$\phi: E \to E'$

More than half the nodes here are in $\mathbb{F}_{p^2}$, but as $p \to \infty$, there are $O(p)$ nodes and only $O(\sqrt{p})$ lie in $\mathbb{F}_p$.

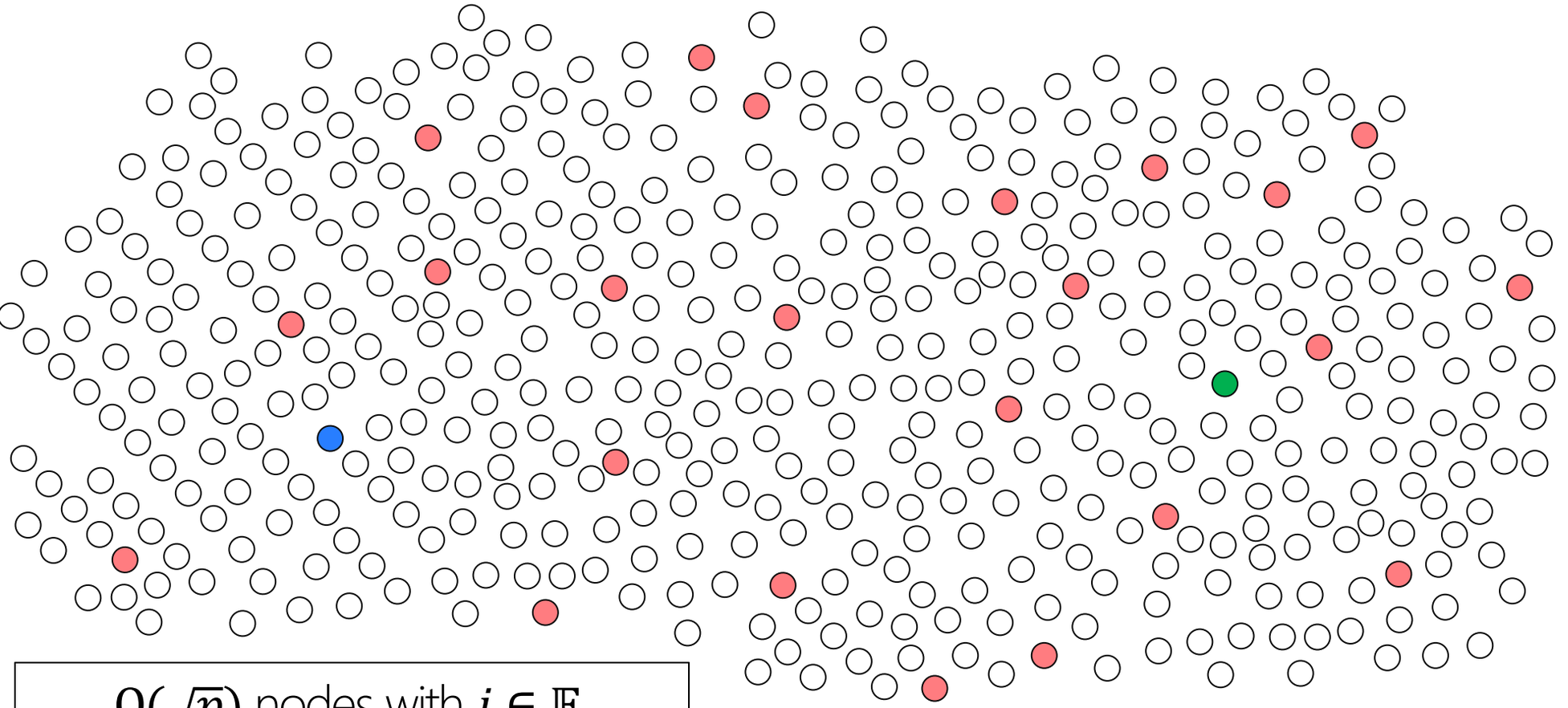


$\phi_{\text{easy}}: \tilde{E} \to \tilde{E}'$

Delfs-Galbraith algorithm

$\phi: \quad E \to E'$

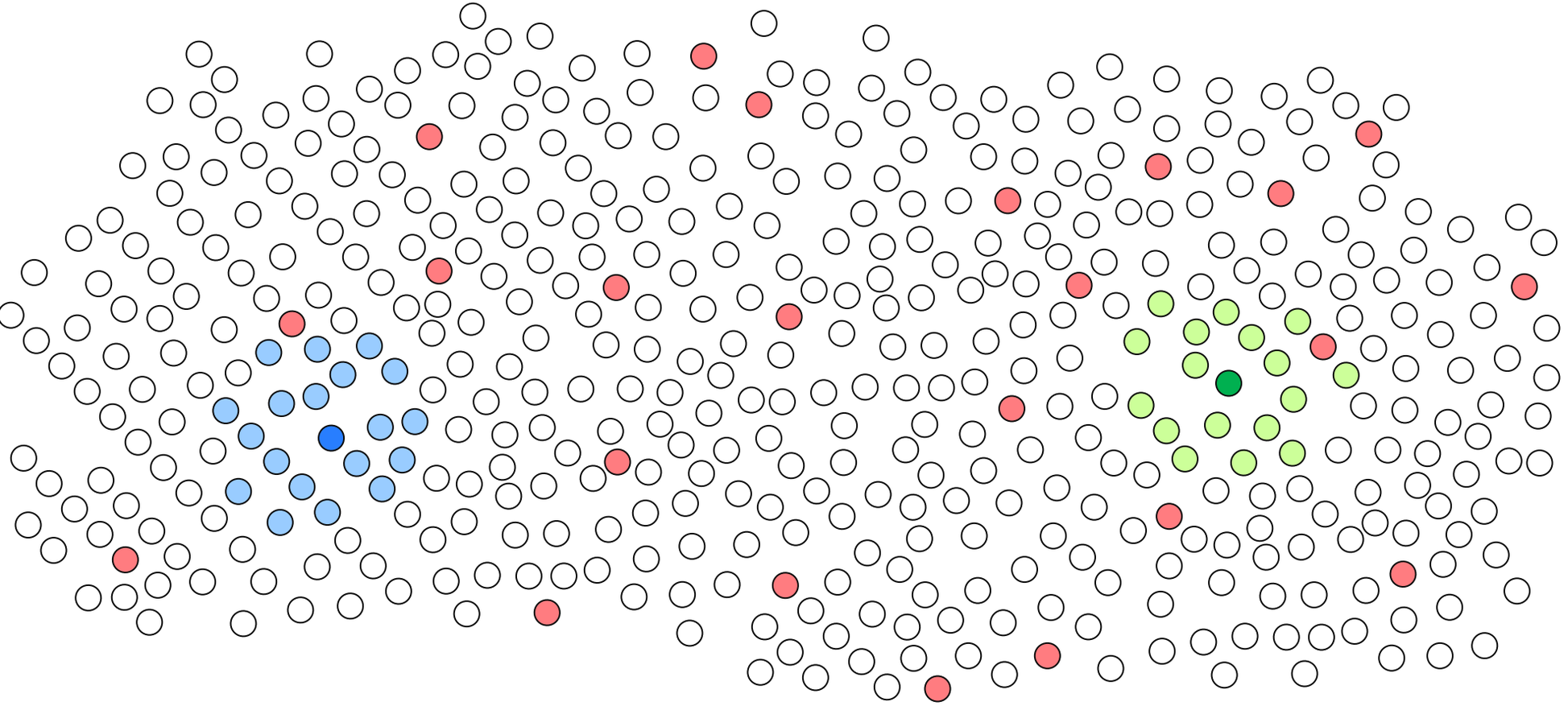
$O(p)$ nodes with $j \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$



$O(\sqrt{p})$ nodes with $j \in \mathbb{F}_p$

Delfs-Galbraith algorithm

$\phi : \quad E \rightarrow E'$



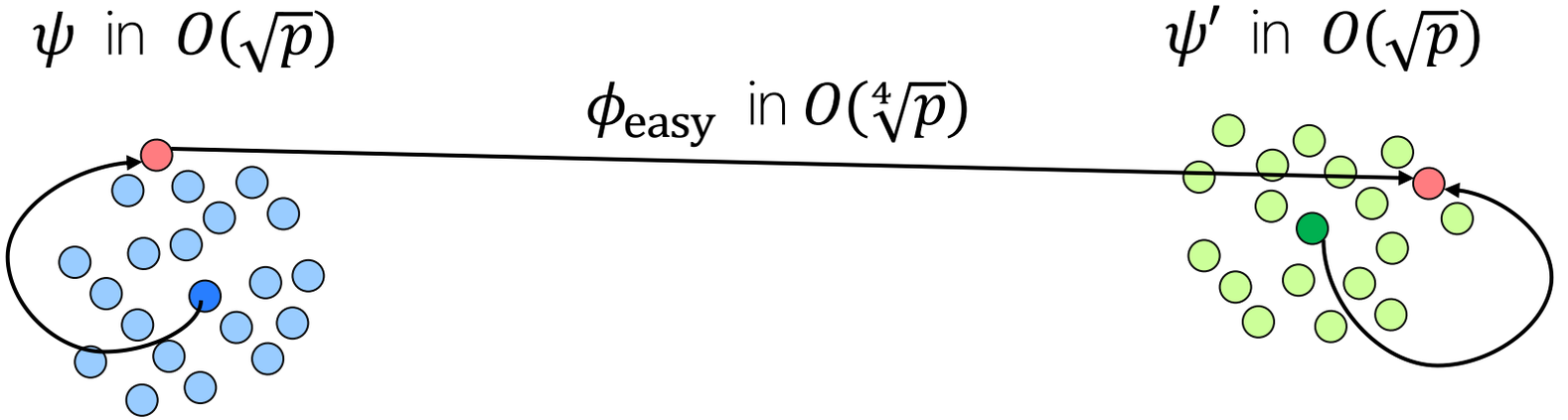
Delfs-Galbraith algorithm

$$\phi : \quad E \to E'$$

$$\psi \downarrow \qquad \downarrow \psi'$$

$$\phi_{\text{easy}}: \quad \tilde{E} \to \tilde{E}'$$

$$\phi = \widehat{\psi}' \circ \phi_{\text{easy}} \circ \psi$$



ϕ in $O(\sqrt{p})$

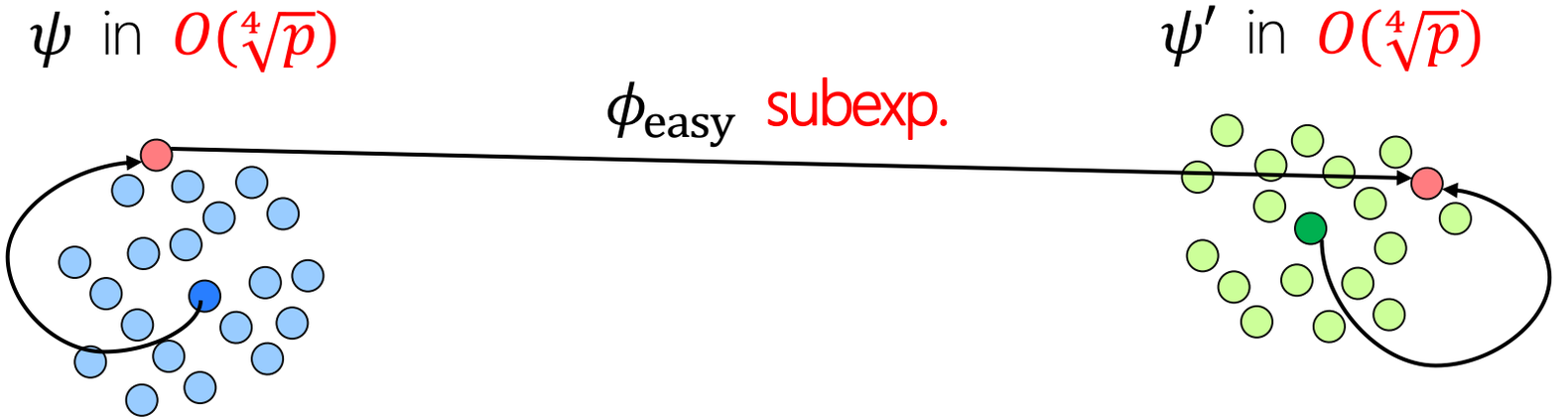
Grovering Delfs-Galbraith
aka the Biasse-Jao-Sankar algorithm

$$\phi : \; E \to E'$$

$$\psi \qquad\qquad \psi'$$

$$\phi_{\text{easy}}: \; \tilde{E} \to \tilde{E}'$$

$$\phi = \widehat{\psi}' \circ \phi_{\text{easy}} \circ \psi$$



ϕ in $O(\sqrt[4]{p})$

Supersingular isogeny crypto in higher dimensions

- $S_g(p)$: superspecial g-dimensional PP abelian varieties A_g (up to isom.)

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

$$\#S_1(p) = O(p),\ \#S_2(p) = O(p^3),\ \#S_3(p) = O(p^6),\ \#S_4(p) = O(p^{10}),\ \ldots$$

- Compare to old-school (H)ECC trade-off

$$\#A_g(\mathbb{F}_p) = O(p^g)$$

$$\#A_1(p) = O(p),\ \#A_2(p) = O(p^2),\ \#A_3(p) = O(p^3),\ \#A_4(p) = O(p^4),\ \ldots.$$

- On one hand, seemingly more potential for higher dimensions (than HECC)
- On the other, we know very little about isogeny graphs beyond $g = 1$

(and this is to say nothing of *how* to compute kernel isogenies efficiently at runtime)

- *Nevertheless……* Hypothesis: $\Gamma_g(\ell\,;p)$ is Ramanujan

Not technically true
(see Jordan-Zaytman) but
still a reasonable assumption

https://arxiv.org/pdf/2005.09031.pdf

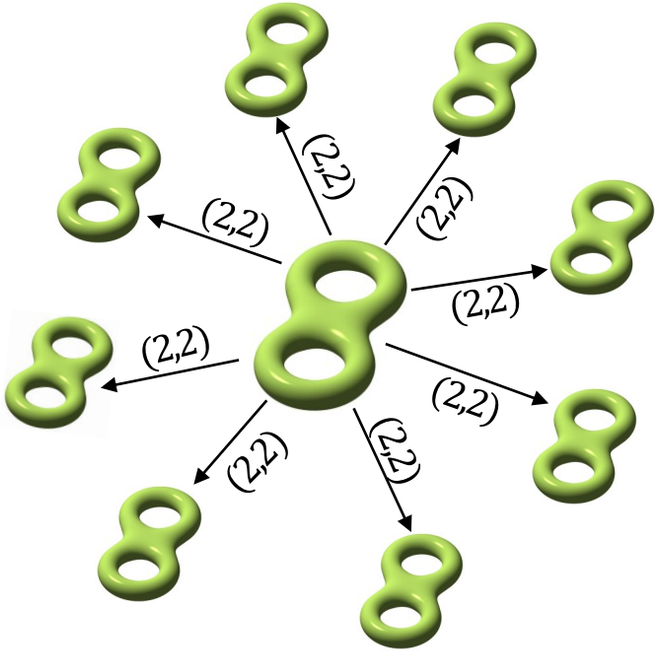
Genus 2

- Flynn-Ti (DH) and Takashima (hash) both had problems that were fixed by Castryck-Decru-Smith (hash):

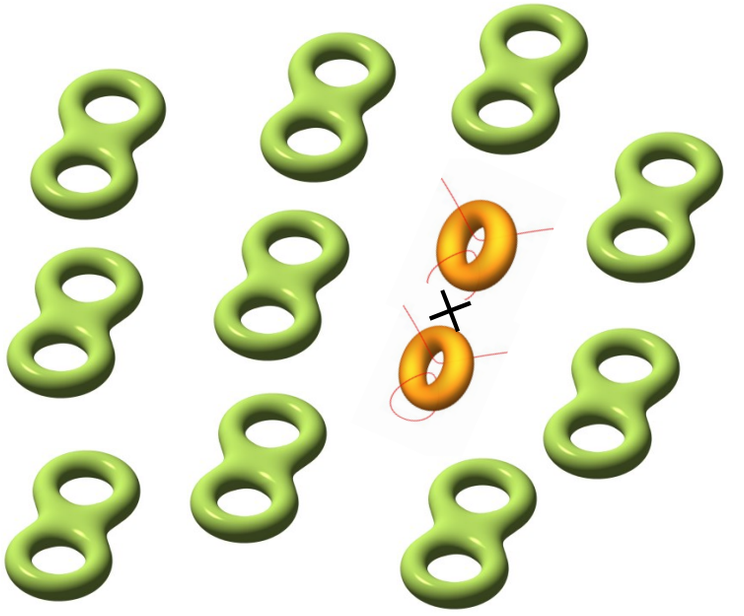
  - Superspecial is the correct notion for $g \geq 1$ (coincides with supersingular for $g = 1$)

  - Of the 15 $(2,2)$-isogenies emanating from $J(\mathbb{F}_{p^2})$, 8 are "good" (avoid trivial cycles)

  - Also show how to constructively avoid special cases…



Special cases?



- Supersingular A_g isogenous to supersingular elliptic product $E_1 \times \ldots \times E_g$

- e.g. $S_2(p) = S_2(p)^J \sqcup S_2(p)^E$, where

$$S_2(p)^J = \{A : A \cong J_C \text{ with } g(C) = 2\}$$

$$S_2(p)^E = \{A : A \cong E_1 \times E_2 \text{ with } E_1, E_2 \in S_1(p)\}$$

$$\#S_2(p)^J = O(p^3) \quad \text{and} \quad \#S_2(p)^E = O(p^2)$$

- CDS show how to avoid $S_2(p)^E$ in the constructive sense, but attacker would still prefer to solve problems in $S_2(p)^E$, i.e. isogeny problems in $S_1(p)$ where $\#S_1(p) = O(p)$

- Crux of paper (e.g. genus 2):

to find ϕ: $A \to A'$ with $A, A' \in S_2(p)^J$

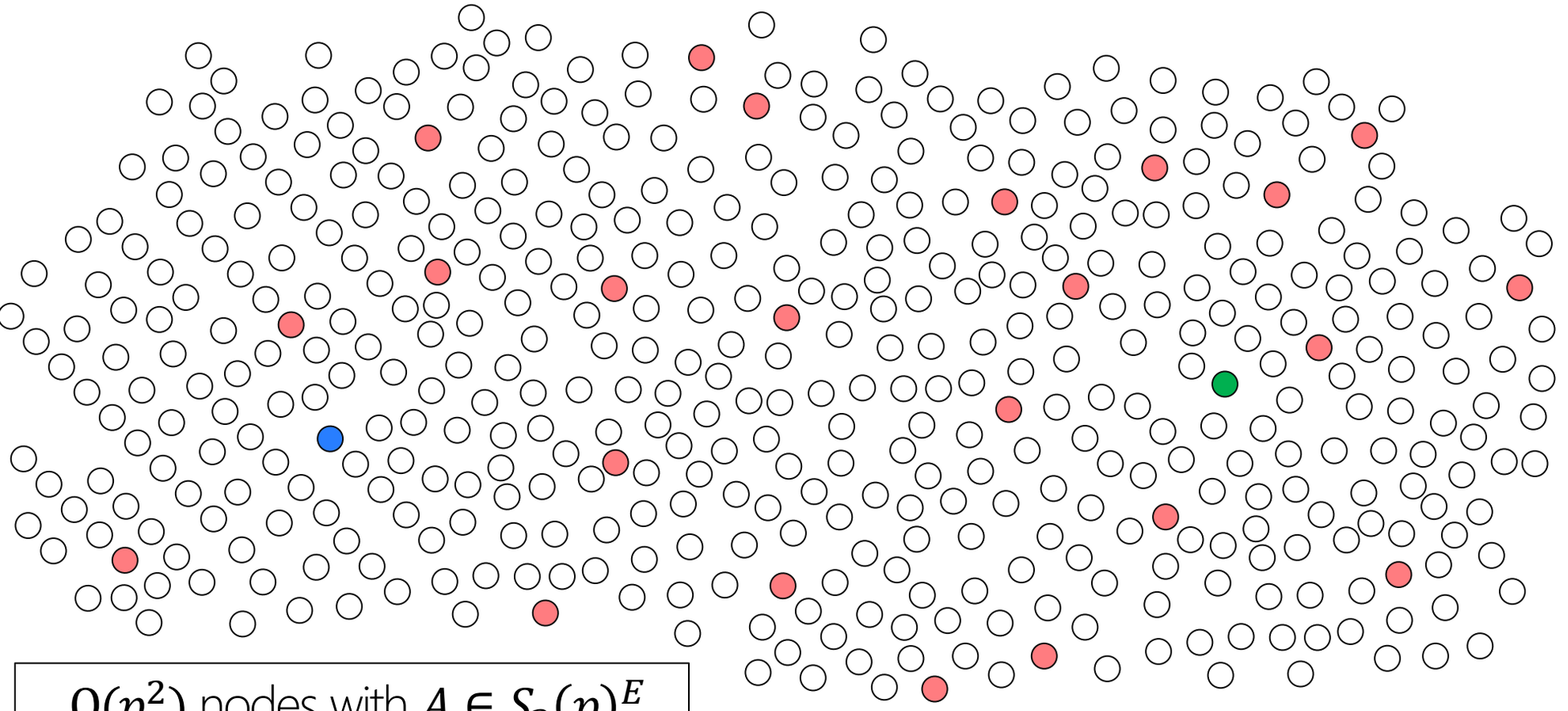
(Step 1) find paths from A to $\tilde{A} \in S_2(p)^E$ and from A' to $\tilde{A'} \in S_2(p)^E$ (mimic Delfs-Galbraith walks)

(Step 2) find path from $\tilde{A} \cong E_1 \times E_2$ to $\tilde{A}' \cong E_1' \times E_2'$ by finding elliptic paths (use actual Delfs-Galbraith)

~~Delfs-Galbraith~~ Algorithm 1 (in genus 2)

$\phi: A \to A'$

$O(p^3)$ nodes with $A \in S_2(p)$



$O(p^2)$ nodes with $A \in S_2(p)^E$

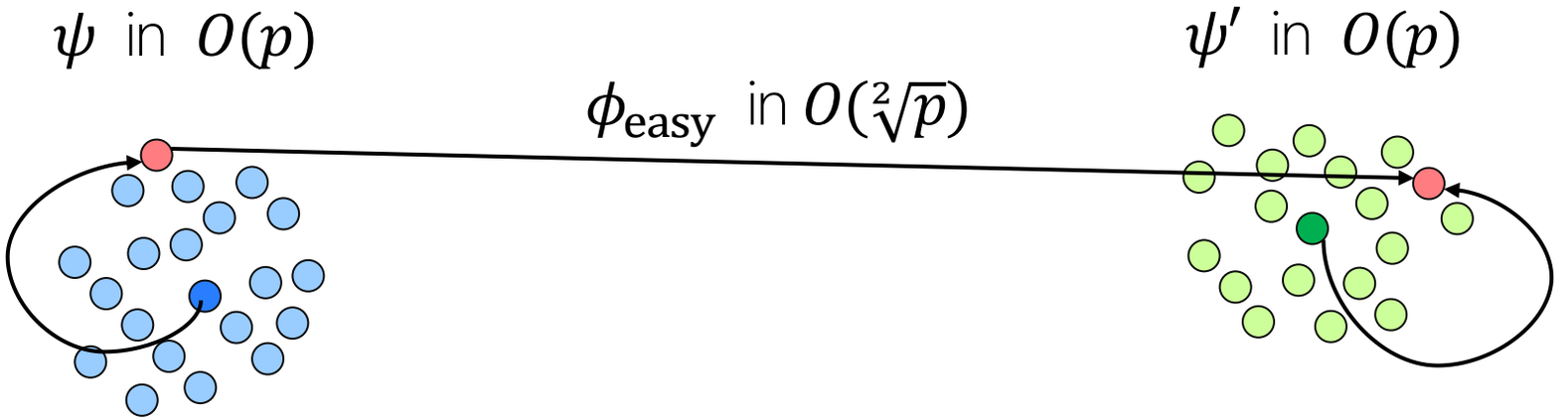
~~Delfs-Galbraith~~ Algorithm 1 (in genus 2)

$$\phi : A \to A'$$

ψ $\qquad$ ψ'

$$\phi = \widehat{\psi}' \circ \phi_{\text{easy}} \circ \psi$$

$$\phi_{\text{easy}}: \ (E_1 \times E_2) \to \widetilde{E_1} \times \widetilde{E_2}$$

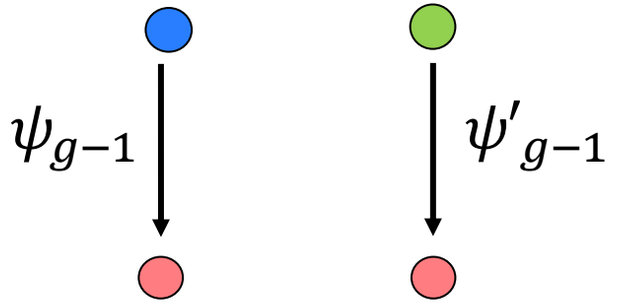


ϕ in $O(p)$

Compare to Pollard rho $O(p^{3/2})$

Algorithm 1 in genus g

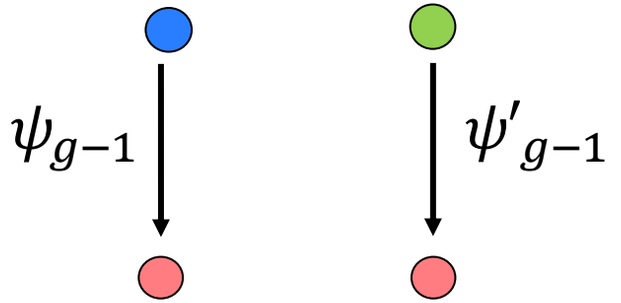
$$\phi_g : A_g \rightarrow A_g{}'$$



$$\phi_{\text{easier}}: \left(E_1 \times A_{g-1}\right) \rightarrow \left(\tilde{E}_1 \times \tilde{A}_{g-1}\right)$$

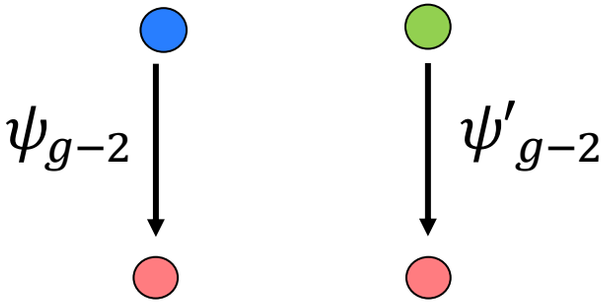
Algorithm 1 in genus g

$$\phi_g : A_g \rightarrow {A_g}'$$



$$\phi_{\text{easier}}: \left(E_1 \times A_{g-1}\right) \rightarrow (\tilde{E}_1 \times \tilde{A}_{g-1})$$

$$\phi_{g-1} : A_{g-1} \rightarrow {A_{g-1}}'$$



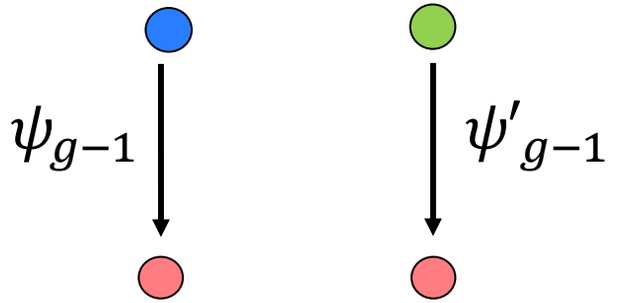
$$\phi_{\text{even easier}}: \left(E_2 \times A_{g-2}\right) \rightarrow (\tilde{E}_2 \times \tilde{A}_{g-2})$$

$\vdots$

$\vdots$

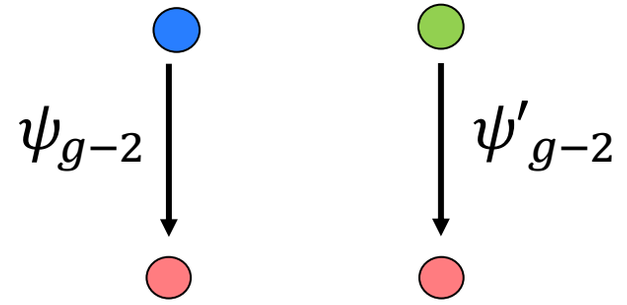
Algorithm 1 in genus g

$\phi_g : A_g \to A_g'$



ϕ_{easier}: $(E_1 \times A_{g-1}) \to (\tilde{E}_1 \times \tilde{A}_{g-1})$

$\phi_{g-1} : A_{g-1} \to A_{g-1}'$



$\phi_{\text{even easier}}$: $(E_2 \times A_{g-2}) \to (\tilde{E}_2 \times \tilde{A}_{g-2})$

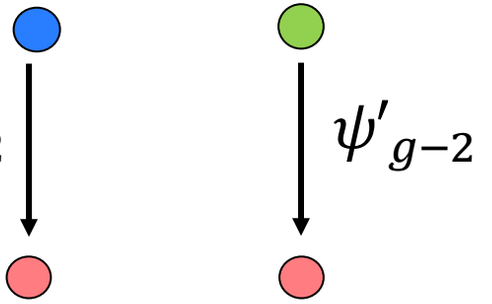
⋮

⋮

⋮

⋮

$\phi_{g-1} : A_2 \to A_2'$

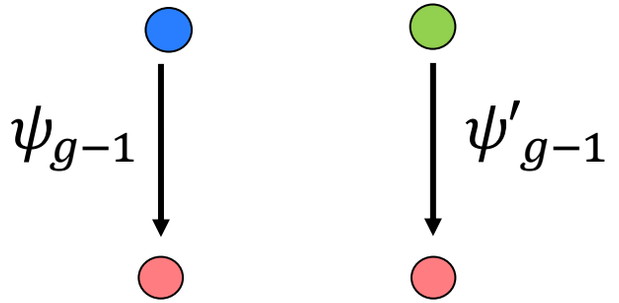


ϕ_{easy}: $(E_{g-1} \times E_g) \to (\tilde{E}_{g-1} \times \tilde{E}_g)$

⋮

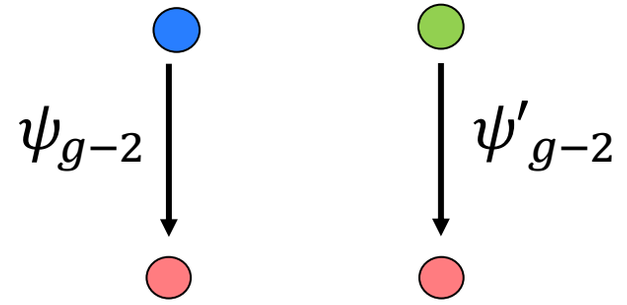
Algorithm 1 in genus g

$\phi_g : A_g \to A_g'$



$\phi_{\text{easier}}: (E_1 \times A_{g-1}) \to (\tilde{E}_1 \times \tilde{A}_{g-1})$

$\phi_{g-1} : A_{g-1} \to A_{g-1}'$



$\phi_{\text{even easier}}: (E_2 \times A_{g-2}) \to (\tilde{E}_2 \times \tilde{A}_{g-2})$

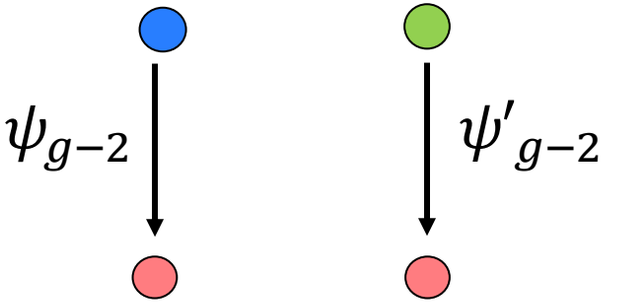
⋮

⋮

⋮

⋮

$\phi_{g-1} : A_2 \to A_2'$



$\phi_{\text{easy}}: (E_{g-1} \times E_g) \to (\tilde{E}_{g-1} \times \tilde{E}_g)$

⋮

- connect paths with $\phi_{\text{easiest},i} : E_i \to \tilde{E}_i$ for $1 \leq i \leq g$ via Delfs-Galbraith
- compose and output ϕ_g

Algorithm 1 complexity

- Finding the two top level paths (e.g. $A_g \to E_1 \times A_{g-1}$) dominates complexity
- Number of nodes to sample before finding product

$$\frac{\#S_g(p)}{\#(\text{image of } S_{g-1}(p) \times S_1(p))} = O(p^{g-1})$$

- Thus, Algorithm 1 runs in

  $$\tilde{O}(p^{g-1})$$

  isogeny operations
- Parallelises perfectly, i.e. $\tilde{O}(p^{g-1}/P)$ isogeny operations on P processes
- Groverises *perfectly*, i.e. $\tilde{O}(\sqrt{p^{g-1}})$ operations on a quantum computer

Complexity of Algorithm 1

	Dimension g	1	2	3	4	5	6
Classical	Delfs-Galbraith	0.5	1.5	3	5	7.5	10.5
	this work		1	2	3	4	5
Quantum	Biasse-Jao-Sankar	0.25	0.75	1.5	2.5	3.75	4.25
	this work		0.5	1	1.5	2	2.5

Table. Asymptotic complexities (base-p logarithms) of algorithms for solving the isogeny problem in $\Gamma_g(\ell\,;p)$

- Above complexities are for the general isogeny problem
- For shorter walks (e.g. SIDH), Alg. 1 still wins for $g \geq 6$, but vOW may win for $g \leq 5$
- Unlike vOW, Alg. 1 is memory-free, parallelises/Groverises perfectly

Summary

- General isogeny problem in dimension g is now

$$\tilde{O}(p^{g-1}) \text{ classical}$$
and
$$\tilde{O}\left(\sqrt{p^{g-1}}\right) \text{ quantum}$$

- This should not be seen as a deterrent for research in high-dimension isogeny crypto, especially not a deterrent for DH-like applications in lower dimensions.

- Many things we still don't know about dimensions $g > 1$

- Still very attractive potential for $g > 1$ warranting more research in these directions