An algorithm for efficient detection of (N, N)-splittings and its application to the isogeny problem in dimension 2

Maria Corte-Real Santos, Craig Costello, **Sam Frengley**

University College London, Microsoft Research, **University of Cambridge**

PKC 2024

Motivation: isogeny-based cryptography

Isogeny-based cryptography is a type of post-quantum cryptography that has been considered in NIST's standardisation process.

Motivation: isogeny-based cryptography

Isogeny-based cryptography is a type of post-quantum cryptography that has been considered in NIST's standardisation process.

The general isogeny problem (in dimension 1) underlies the security of many isogeny-based schemes (e.g., SQIsign).

Motivation: isogeny-based cryptography

Isogeny-based cryptography is a type of post-quantum cryptography that has been considered in NIST's standardisation process.

The general isogeny problem (in dimension 1) underlies the security of many isogeny-based schemes (e.g., SQIsign). Similarly, we consider the dimension 2 isogeny problem an upper bound for the security of dimension 2 isogeny-based protocols.

Motivation: isogeny-based cryptography

Isogeny-based cryptography is a type of post-quantum cryptography that has been considered in NIST's standardisation process.

The general isogeny problem (in dimension 1) underlies the security of many isogeny-based schemes (e.g., SQIsign). Similarly, we consider the dimension 2 isogeny problem an upper bound for the security of dimension 2 isogeny-based protocols.

The SIDH attacks also showed that understanding higher dimensional isogenies is needed to navigate the isogeny graph in dimension 1.

Motivation: isogeny-based cryptography

Isogeny-based cryptography is a type of post-quantum cryptography that has been considered in NIST's standardisation process.

The general isogeny problem (in dimension 1) underlies the security of many isogeny-based schemes (e.g., SQIsign). Similarly, we consider the dimension 2 isogeny problem an upper bound for the security of dimension 2 isogeny-based protocols.

The SIDH attacks also showed that understanding higher dimensional isogenies is needed to navigate the isogeny graph in dimension 1.

Much is conjectured, but little is known about the isogeny problem in dimension 2.

Motivation: isogeny-based cryptography

Isogeny-based cryptography is a type of post-quantum cryptography that has been considered in NIST's standardisation process.

The general isogeny problem (in dimension 1) underlies the security of many isogeny-based schemes (e.g., SQIsign). Similarly, we consider the dimension 2 isogeny problem an upper bound for the security of dimension 2 isogeny-based protocols.

The SIDH attacks also showed that understanding higher dimensional isogenies is needed to navigate the isogeny graph in dimension 1.

Much is conjectured, but little is known about the isogeny problem in dimension 2.

In this work we look at the problem in dimension 2 and decrease the concrete complexity of the best attack due to Costello–Smith.

Moving to dimension 2

To generalise supersingular elliptic curves over $\mathbb{F}_{p^2}$ to genus 2, we consider *superspecial (principally polarised) abelian surfaces* over $\mathbb{F}_{p^2}$.

Moving to dimension 2

To generalise supersingular elliptic curves over $\mathbb{F}_{p^2}$ to genus 2, we consider *superspecial (principally polarised) abelian surfaces* over $\mathbb{F}_{p^2}$.

There are two types:

Moving to dimension 2

To generalise supersingular elliptic curves over $\mathbb{F}_{p^2}$ to genus 2, we consider *superspecial (principally polarised) abelian surfaces* over $\mathbb{F}_{p^2}$.

There are two types:

1. Products of supersingular elliptic curves $E \times E'$

Moving to dimension 2

To generalise supersingular elliptic curves over $\mathbb{F}_{p^2}$ to genus 2, we consider *superspecial (principally polarised) abelian surfaces* over $\mathbb{F}_{p^2}$.

There are two types:

1. Products of supersingular elliptic curves $E \times E'$
2. Jacobians $\mathrm{Jac}(C)$ of genus 2 curves C

Moving to dimension 2

To generalise supersingular elliptic curves over $\mathbb{F}_{p^2}$ to genus 2, we consider *superspecial (principally polarised) abelian surfaces* over $\mathbb{F}_{p^2}$.

There are two types:

1. Products of supersingular elliptic curves $E \times E'$
2. Jacobians $\mathrm{Jac}(C)$ of genus 2 curves C

We study (N, N)-isogenies, which generalise N-isogenies to dimension 2 (kernel now generated by two points).

Moving to dimension 2

To generalise supersingular elliptic curves over $\mathbb{F}_{p^2}$ to genus 2, we consider *superspecial (principally polarised) abelian surfaces* over $\mathbb{F}_{p^2}$.

There are two types:

1. Products of supersingular elliptic curves $E \times E'$
2. Jacobians $\mathrm{Jac}(C)$ of genus 2 curves C

We study (N, N)-isogenies, which generalise N-isogenies to dimension 2 (kernel now generated by two points).

For the purposes of this talk, we only need to keep in mind that there are two types of surfaces: "reducible" and "non-reducible".

General Isogeny Problem in Two Dimensions

In its most general form, the superspecial isogeny problem in two dimensions asks to find an isogeny

$$\phi\colon A \longrightarrow A',$$

between two superspecial (p.p.) abelian surfaces $A/\mathbb{F}_{p^2}$ and $A'/\mathbb{F}_{p^2}$.

General Isogeny Problem in Two Dimensions

In its most general form, the superspecial isogeny problem in two dimensions asks to find an isogeny

$$\phi\colon A \longrightarrow A',$$

between two superspecial (p.p.) abelian surfaces $A/\mathbb{F}_{p^2}$ and $A'/\mathbb{F}_{p^2}$.

The general isogeny problem can be viewed as finding a path between two nodes in the superspecial isogeny graph.

The Superspecial Isogeny Graph $\Gamma(N; p)$

Let $p > N$ be a large prime.

The Superspecial Isogeny Graph $\Gamma(N; p)$

Let $p > N$ be a large prime. $\Gamma(N; p)$ is the graph with vertex set

$$\mathcal{S}(p) = \Big\{\text{Superspecial p.p. abelian surfaces over } \mathbb{F}_{p^2} \text{ (up to isomorphism)}\Big\},$$

and whose edges are (N, N)-isogenies (defined over $\bar{\mathbb{F}}_p$).

The Superspecial Isogeny Graph $\Gamma(N; p)$

Let $p > N$ be a large prime. $\Gamma(N; p)$ is the graph with vertex set

$$\mathcal{S}(p) = \Big\{ \text{Superspecial p.p. abelian surfaces over } \mathbb{F}_{p^2} \text{ (up to isomorphism)} \Big\},$$

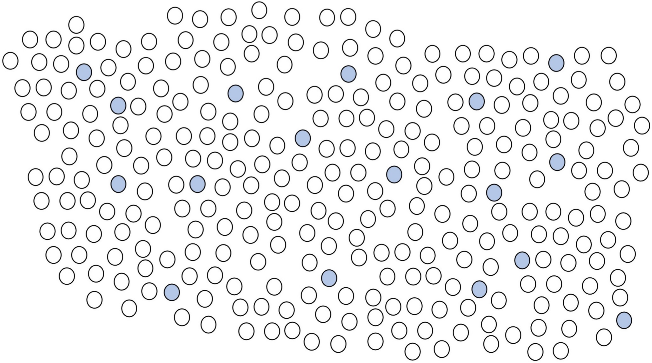
and whose edges are (N, N)-isogenies (defined over $\bar{\mathbb{F}}_p$).

$\mathcal{S}(p)$ is equal to the disjoint union of:

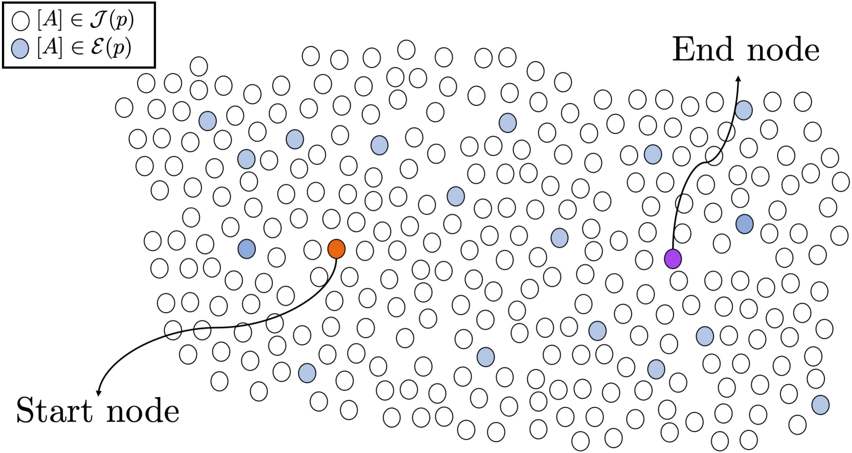
$$\begin{aligned}
\mathcal{E}(p) &:= \{A \in \mathcal{S}(p) \ : \ A \cong E \times E' \text{ with } E,\ E' \text{ supersingular ECs}\}. \\
\mathcal{J}(p) &:= \mathcal{S}(p) \setminus \mathcal{E}(p) \\
&= \{A \in \mathcal{S}(p) \ : \ A \cong \mathrm{Jac}(C)\}
\end{aligned}$$

The Superspecial Isogeny Graph $\Gamma(N; p)$

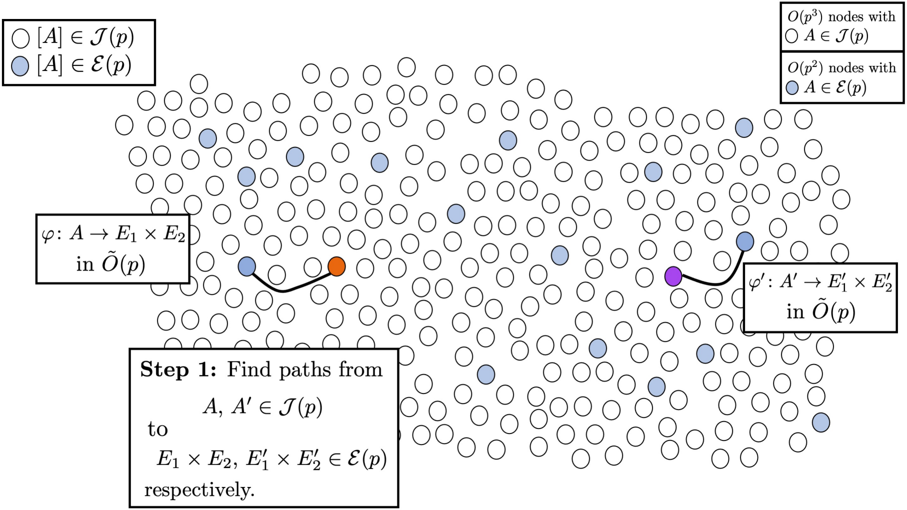
$O(p^3)$ nodes with ○ $A \in \mathcal{J}(p)$
$O(p^2)$ nodes with ● $A \in \mathcal{E}(p)$



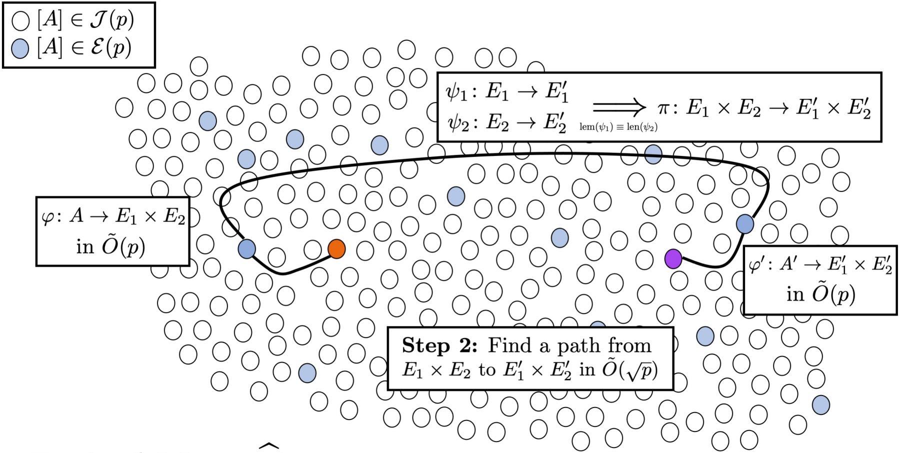
Attacking the General Isogeny Problem: Costello–Smith



Attacking the General Isogeny Problem: Costello–Smith



Attacking the General Isogeny Problem: Costello–Smith



Desired Map: $\widehat{\varphi'} \circ \pi \circ \varphi$

Attacking the General Isogeny Problem: First step

Summary: Using $(2,2)$-isogenies, Costello–Smith take walks in $\Gamma(2;p)$ and detect whether nodes are $(2,2)$-isogenous to a product.

Attacking the General Isogeny Problem: First step

Summary: Using $(2,2)$-isogenies, Costello–Smith take walks in $\Gamma(2;p)$ and detect whether nodes are $(2,2)$-isogenous to a product.

First step in more detail:

1. We start on a node $A_0 \in \mathcal{J}(p)$.

Attacking the General Isogeny Problem: First step

Summary: Using $(2,2)$-isogenies, Costello–Smith take walks in $\Gamma(2;p)$ and detect whether nodes are $(2,2)$-isogenous to a product.

First step in more detail:

1. We start on a node $A_0 \in \mathcal{J}(p)$.
2. Take a step in $\Gamma(2;p)$ via a $(2,2)$-isogeny $\phi_1 \colon A_0 \to A_1$.

Attacking the General Isogeny Problem: First step

Summary: Using $(2,2)$-isogenies, Costello–Smith take walks in $\Gamma(2;p)$ and detect whether nodes are $(2,2)$-isogenous to a product.

First step in more detail:

1. We start on a node $A_0 \in \mathcal{J}(p)$.
2. Take a step in $\Gamma(2;p)$ via a $(2,2)$-isogeny $\phi_1 \colon A_0 \to A_1$.
3. We can determine whether $A_1 \in \mathcal{E}(p)$. If not, take another step $\phi_2 \colon A_1 \to A_2$.

Attacking the General Isogeny Problem: First step

Summary: Using $(2,2)$-isogenies, Costello–Smith take walks in $\Gamma(2;p)$ and detect whether nodes are $(2,2)$-isogenous to a product.

First step in more detail:

1. We start on a node $A_0 \in \mathcal{J}(p)$.
2. Take a step in $\Gamma(2;p)$ via a $(2,2)$-isogeny $\phi_1 \colon A_0 \to A_1$.
3. We can determine whether $A_1 \in \mathcal{E}(p)$. If not, take another step $\phi_2 \colon A_1 \to A_2$.
4. Repeat previous step until finding $A_i \in \mathcal{E}(p)$.

Attacking the General Isogeny Problem: First step

Summary: Using $(2,2)$-isogenies, Costello–Smith take walks in $\Gamma(2;p)$ and detect whether nodes are $(2,2)$-isogenous to a product.

First step in more detail:

1. We start on a node $A_0 \in \mathcal{J}(p)$.
2. Take a step in $\Gamma(2;p)$ via a $(2,2)$-isogeny $\phi_1 \colon A_0 \to A_1$.
3. We can determine whether $A_1 \in \mathcal{E}(p)$. If not, take another step $\phi_2 \colon A_1 \to A_2$.
4. Repeat previous step until finding $A_i \in \mathcal{E}(p)$.

Question: Taking steps in $\Gamma(2;p)$, can we detect whether the current node A_i is in (N,N)-split (i.e., (N,N)-isogenous to a product) for $N > 2$?

Attacking the General Isogeny Problem: First step

Summary: Using $(2,2)$-isogenies, Costello–Smith take walks in $\Gamma(2; p)$ and detect whether nodes are $(2,2)$-isogenous to a product.

First step in more detail:

1. We start on a node $A_0 \in \mathcal{J}(p)$.
2. Take a step in $\Gamma(2; p)$ via a $(2,2)$-isogeny $\phi_1 \colon A_0 \to A_1$.
3. We can determine whether $A_1 \in \mathcal{E}(p)$. If not, take another step $\phi_2 \colon A_1 \to A_2$.
4. Repeat previous step until finding $A_i \in \mathcal{E}(p)$.

Question: Taking steps in $\Gamma(2; p)$, can we detect whether the current node A_i is in (N,N)-split (i.e., (N,N)-isogenous to a product) for $N > 2$?

Naive Answer: Compute all (N,N)-isogenies from A_i, but this is not efficient. Can we make the detection efficient?

Detecting an (N, N)-splitting

There exist (easily computable) functions $\boldsymbol{\alpha}(A) = (\alpha_1(A), \alpha_2(A), \alpha_3(A))$ which assigns to A a triple of elements of $\mathbb{F}_{p^2}$ which uniquely determine $A^\dagger$.

†Up to isomorphism. These are (normalised) *Igusa invariants*.

Detecting an (N, N)-splitting

There exist (easily computable) functions $\boldsymbol{\alpha}(A) = (\alpha_1(A), \alpha_2(A), \alpha_3(A))$ which assigns to A a triple of elements of $\mathbb{F}_{p^2}$ which uniquely determine $A^\dagger$.

For $N \leq 11$, Kumar [Kum15] provides rational functions $i_1(r, s)$, $i_2(r, s)$, $i_3(r, s) \in \mathbb{F}_p(r, s)$, such that if there exists a simultaneous solution $r_0, s_0 \in \bar{\mathbb{F}}_p$ of

$$\begin{cases} i_1(r, s) = \alpha_1(A) \\ i_2(r, s) = \alpha_2(A) \\ i_3(r, s) = \alpha_3(A) \end{cases}$$

and the denominators do not vanish at (r_0, s_0), then A is (N, N)-split.

†Up to isomorphism. These are (normalised) *Igusa invariants*.

Detecting an (N, N)-splitting

Let $f_k(r, s) = i_k(r, s) - \alpha_k(A)$.

Detecting an (N, N)-splitting

Let $f_k(r, s) = i_k(r, s) - \alpha_k(A)$. We determine if there exist r_0, s_0 by:

Detecting an (N, N)-splitting

Let $f_k(r, s) = i_k(r, s) - \alpha_k(A)$. We determine if there exist r_0, s_0 by:

(1) Computing resultants of (the numerators of) $f_1(r, s)$, $f_2(r, s)$ and $f_2(r, s)$, $f_3(r, s)$ (with respect to r) to get $\mathrm{res}_1(s), \mathrm{res}_2(s)$.

Detecting an (N, N)-splitting

Let $f_k(r, s) = i_k(r, s) - \alpha_k(A)$. We determine if there exist r_0, s_0 by:

(1) Computing resultants of (the numerators of) $f_1(r, s)$, $f_2(r, s)$ and $f_2(r, s)$, $f_3(r, s)$ (with respect to r) to get $\mathrm{res}_1(s), \mathrm{res}_2(s)$.

(2) Compute $\gcd(\mathrm{res}_1(s), \mathrm{res}_2(s))$.

Detecting an (N, N)-splitting

Let $f_k(r, s) = i_k(r, s) - \alpha_k(A)$. We determine if there exist r_0, s_0 by:

(1) Computing resultants of (the numerators of) $f_1(r, s)$, $f_2(r, s)$ and $f_2(r, s)$, $f_3(r, s)$ (with respect to r) to get $\mathrm{res}_1(s), \mathrm{res}_2(s)$.

(2) Compute $\gcd(\mathrm{res}_1(s), \mathrm{res}_2(s))$.
   - If degree is 0, then A is not (N, N)-split.

Detecting an (N, N)-splitting

Let $f_k(r, s) = i_k(r, s) - \alpha_k(A)$. We determine if there exist r_0, s_0 by:

(1) Computing resultants of (the numerators of) $f_1(r, s)$, $f_2(r, s)$ and $f_2(r, s)$, $f_3(r, s)$ (with respect to r) to get $\mathrm{res}_1(s), \mathrm{res}_2(s)$.

(2) Compute $\gcd(\mathrm{res}_1(s), \mathrm{res}_2(s))$.

- If degree is 0, then A is not (N, N)-split.
- Otherwise, A is (N, N)-split.

Detecting an (N, N)-splitting

Let $f_k(r,s) = i_k(r,s) - \alpha_k(A)$. We determine if there exist r_0, s_0 by:

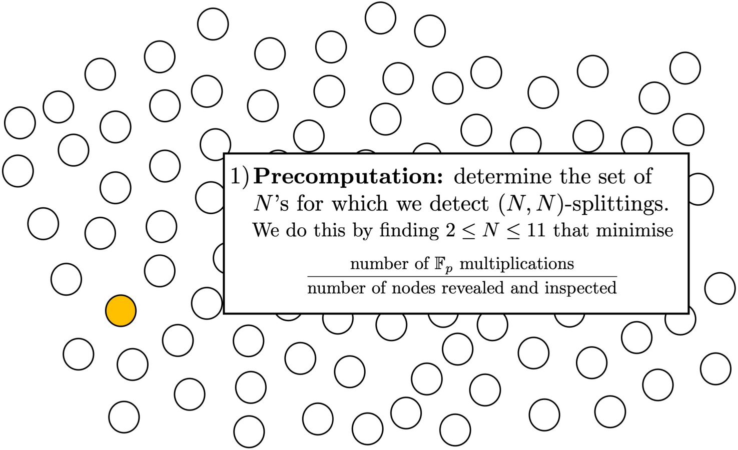
(1) Computing resultants of (the numerators of) $f_1(r,s)$, $f_2(r,s)$ and $f_2(r,s)$, $f_3(r,s)$ (with respect to r) to get $\mathrm{res}_1(s), \mathrm{res}_2(s)$.

(2) Compute $\gcd(\mathrm{res}_1(s), \mathrm{res}_2(s))$.
   - If degree is 0, then A is not (N, N)-split.
   - Otherwise, A is (N, N)-split.

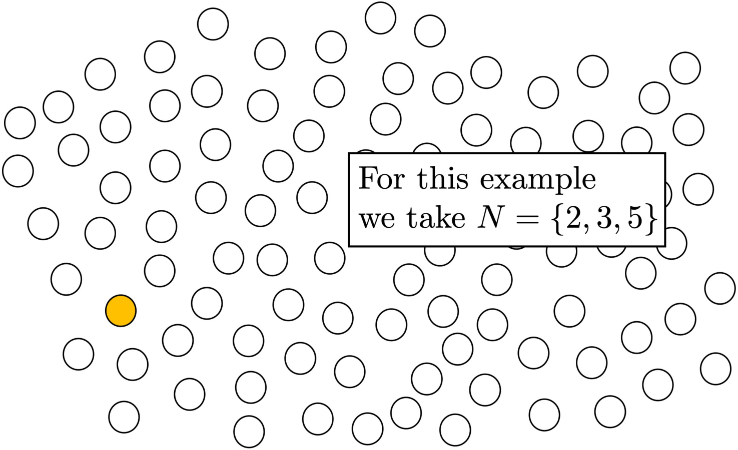
In fact, we obtain a more efficient method by precomputing the resultants generically.

Attacking the General Isogeny Problem: Revisted

We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.

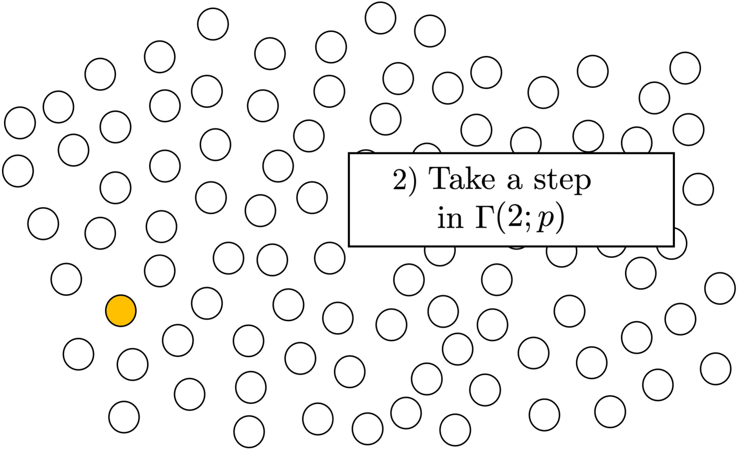


We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.



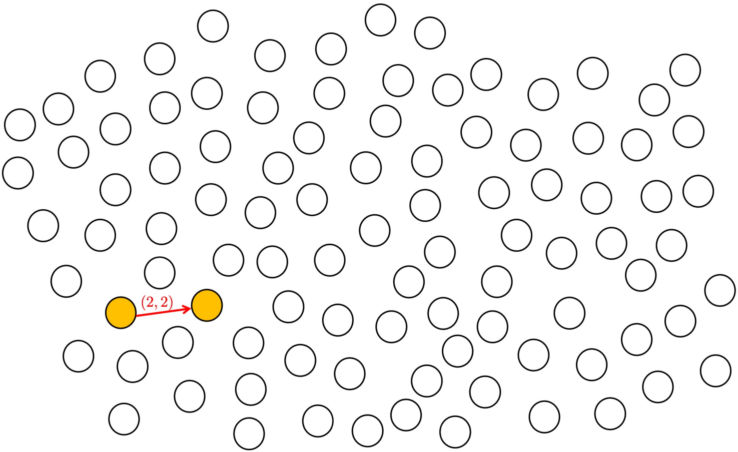
Attacking the General Isogeny Problem: Revisted

We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.



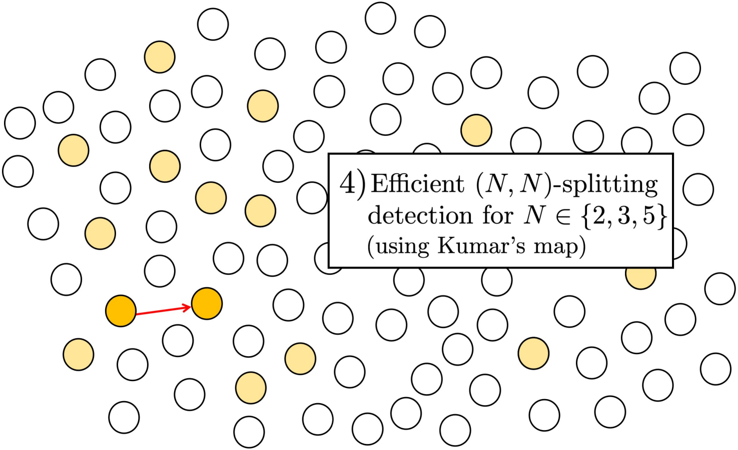
Attacking the General Isogeny Problem: Revisted

We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.



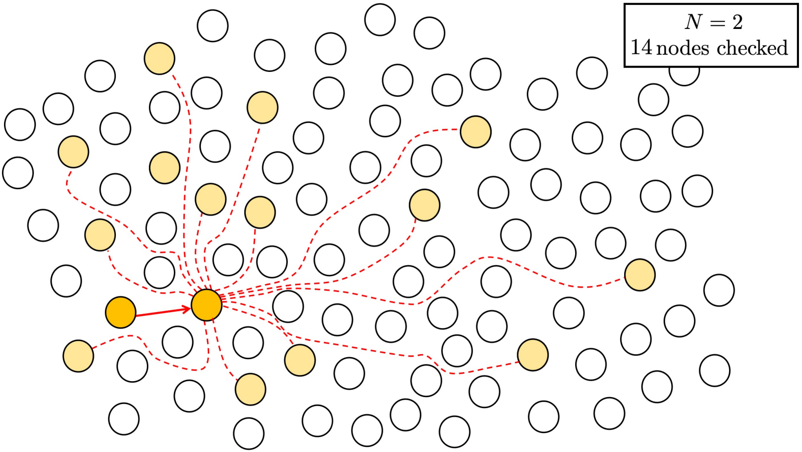
Attacking the General Isogeny Problem: Revisted

We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.



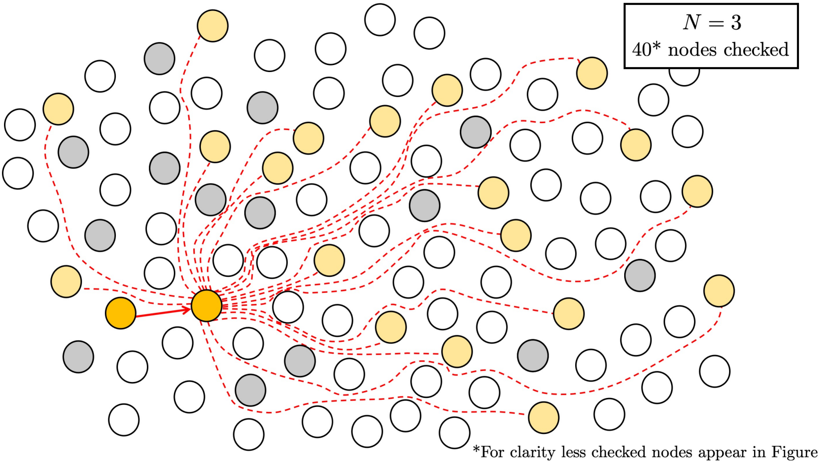
Attacking the General Isogeny Problem: Revisted

We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.



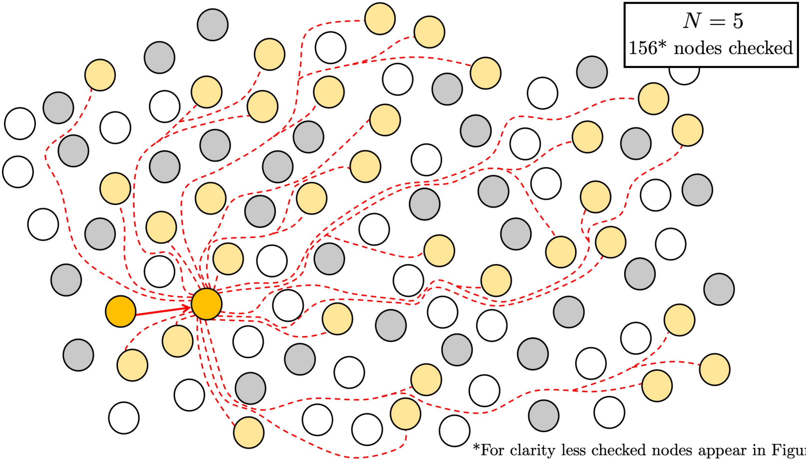
Attacking the General Isogeny Problem: Revisted

We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.



Attacking the General Isogeny Problem: Revisted

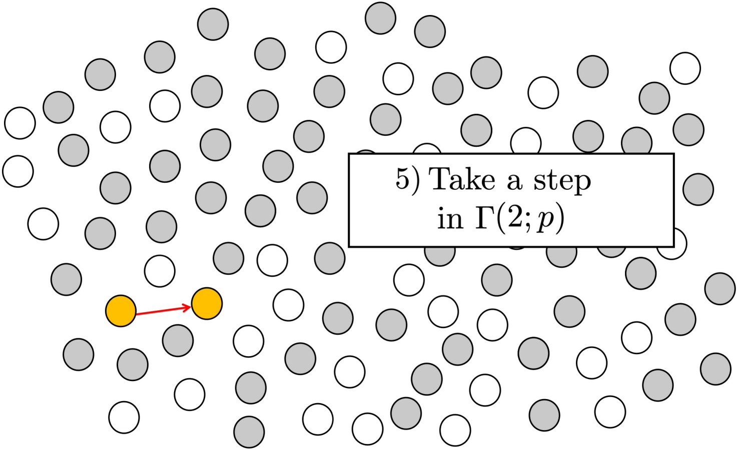
We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.



*For clarity less checked nodes appear in Figure

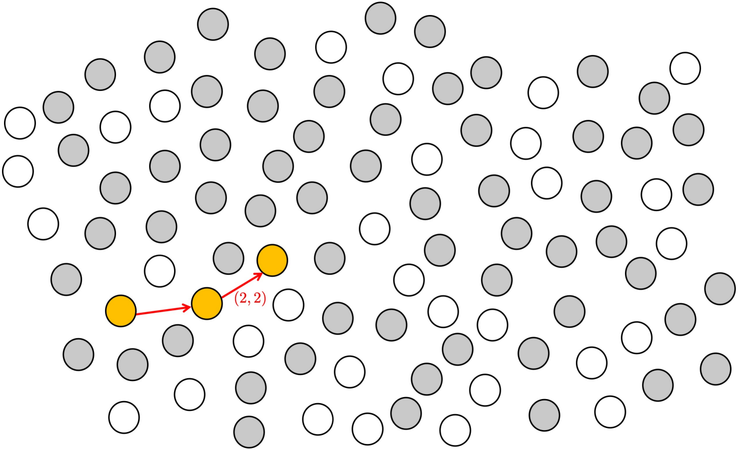
Attacking the General Isogeny Problem: Revisted

We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.



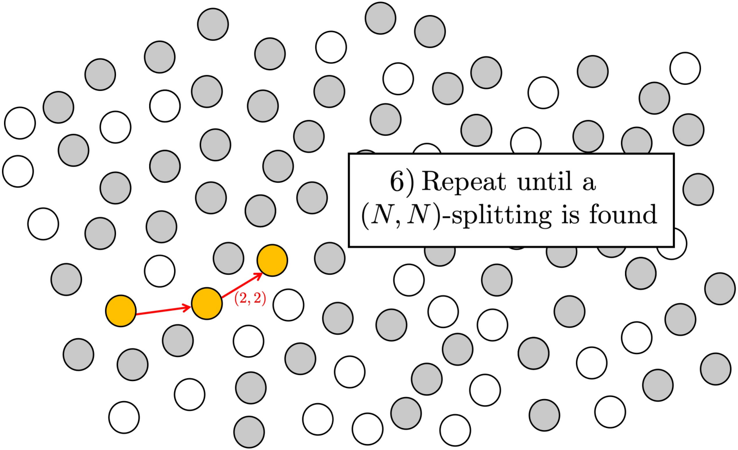
Attacking the General Isogeny Problem: Revisted

We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.



Attacking the General Isogeny Problem: Revisted

We now apply efficient splitting detection to the Costello–Smith algorithm and decreasing its concrete complexity.



Results

We implemented and optimised the first step of Costello–Smith attack with *and* without detection of (N, N)-splitting. We ran these (for primes p of bitsizes $50 - 1000$) until reaching 10^8 $\mathbb{F}_p$ multiplications.

Results

We implemented and optimised the first step of Costello–Smith attack with *and* without detection of (N, N)-splitting. We ran these (for primes p of bitsizes $50 - 1000$) until reaching 10^8 $\mathbb{F}_p$ multiplications. We counted the number of nodes revealed and $\mathbb{F}_p$ multiplications per node revealed.

Results

We implemented and optimised the first step of Costello–Smith attack with *and* without detection of (N, N)-splitting. We ran these (for primes p of bitsizes $50 - 1000$) until reaching 10^8 $\mathbb{F}_p$ multiplications. We counted the number of nodes revealed and $\mathbb{F}_p$ multiplications per node revealed.

	Walks in $\Gamma_2(2; p)$ without additional searching [CS20]			Walks in $\Gamma_2(2; p)$ w. split searching in $\Gamma_2(N; p)$ **This work**			
prime p	bits p	nodes per 10^8 muls	muls per node	set $N \in \{\dots\}$	nodes per 10^8 muls	muls per node	**imprv. factor**
$2^{11} \cdot 3^{24} - 1$	50	172712	579	$\{2, 3\}$	2830951	35	**16.5**
$2^{27} \cdot 3^{77} - 1$	150	63492	1575	$\{3, 4\}$	1858912	54	**29.2**
$2^{181} \cdot 3^{43} - 1$	250	34083	2934	$\{4, 6\}$	1771608	56	**52.4**
$2^{113} \cdot 3^{244} - 1$	500	20239	4941	$\{4, 6\}$	1667360	60	**82.4**
$2^{107} \cdot 3^{437} - 1$	800	13228	7560	$\{4, 6\}$	1548504	65	**116.3**
$2^{721} \cdot 3^{176} - 1$	1000	8814	11346	$\{4, 6\}$	1403752	71	**159.8**

Results

We implemented and optimised the first step of Costello–Smith attack with *and* without detection of (N, N)-splitting. We ran these (for primes p of bitsizes $50 - 1000$) until reaching 10^8 $\mathbb{F}_p$ multiplications. We counted the number of nodes revealed and $\mathbb{F}_p$ multiplications per node revealed.

	Walks in $\Gamma_2(2; p)$ without additional searching [CS20]			Walks in $\Gamma_2(2; p)$ w. split searching in $\Gamma_2(N; p)$ **This work**			
prime p	bits p	nodes per 10^8 muls	muls per node	set $N \in \{\dots\}$	nodes per 10^8 muls	muls per node	**imprv. factor**
$2^{11} \cdot 3^{24} - 1$	50	172712	579	$\{2, 3\}$	2830951	35	**16.5**
$2^{27} \cdot 3^{77} - 1$	150	63492	1575	$\{3, 4\}$	1858912	54	**29.2**
$2^{181} \cdot 3^{43} - 1$	250	34083	2934	$\{4, 6\}$	1771608	56	**52.4**
$2^{113} \cdot 3^{244} - 1$	500	20239	4941	$\{4, 6\}$	1667360	60	**82.4**
$2^{107} \cdot 3^{437} - 1$	800	13228	7560	$\{4, 6\}$	1548504	65	**116.3**
$2^{721} \cdot 3^{176} - 1$	1000	8814	11346	$\{4, 6\}$	1403752	71	**159.8**

Any questions?

`eprint.iacr.org/2022/1736`