Solving the Hidden Number Problem for CSIDH and CSURF via Automated Coppersmith

Jonas Meers

Julian Nowakowski

Ruhr University Bochum

December 7, 2023

Motivation

Definition

A **group action** is a map

$$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}, \qquad (g, x) \mapsto g \star x$$

for group $\mathcal{G}$ and set $\mathcal{X}$.

Motivation

Definition

A **group action** is a map

$$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}, \qquad (g, x) \mapsto g \star x$$

for group $\mathcal{G}$ and set $\mathcal{X}$.

Example

Exponentiation:	$(a, h) \mapsto$	h^a
Scalar multiplication:	$(m, P) \mapsto$	$[m]P$
Isogenies:	$([\mathfrak{a}], E) \mapsto$	$E/E[\mathfrak{a}]$

Motivation

Definition

A **group action** is a map

$$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}, \qquad (g, x) \mapsto g \star x$$

for group $\mathcal{G}$ and set $\mathcal{X}$.

Example

Exponentiation:	$(a, h) \mapsto$	h^a
Scalar multiplication:	$(m, P) \mapsto$	$[m]P$
Isogenies:	$([\mathfrak{a}], E) \mapsto$	$E/E[\mathfrak{a}]$

Alice

$a \xleftarrow{\$} \mathcal{G}$

$x_A = a \star x$

Public:

$\mathcal{G},\ \mathcal{X},\ x \in \mathcal{X}$

Bob

$b \xleftarrow{\$} \mathcal{G}$

$x_B = b \star x$

Motivation

Definition

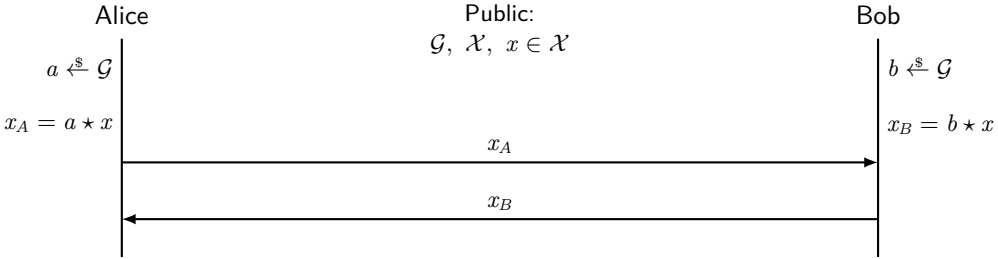
A **group action** is a map

$$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}, \qquad (g, x) \mapsto g \star x$$

for group $\mathcal{G}$ and set $\mathcal{X}$.

Example

Exponentiation:	$(a, h) \mapsto$	h^a
Scalar multiplication:	$(m, P) \mapsto$	$[m]P$
Isogenies:	$([\mathfrak{a}], E) \mapsto$	$E/E[\mathfrak{a}]$



Motivation

Definition

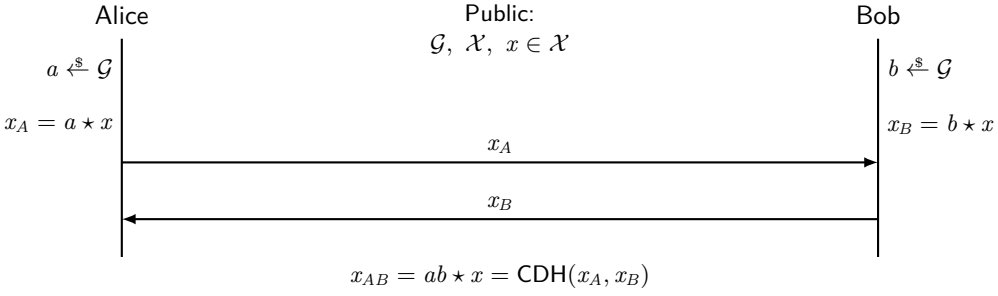
A **group action** is a map

$$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}, \qquad (g, x) \mapsto g \star x$$

for group $\mathcal{G}$ and set $\mathcal{X}$.

Example

Exponentiation:	$(a, h) \mapsto$	h^a
Scalar multiplication:	$(m, P) \mapsto$	$[m]P$
Isogenies:	$([\mathfrak{a}], E) \mapsto$	$E/E[\mathfrak{a}]$



Motivation

Definition

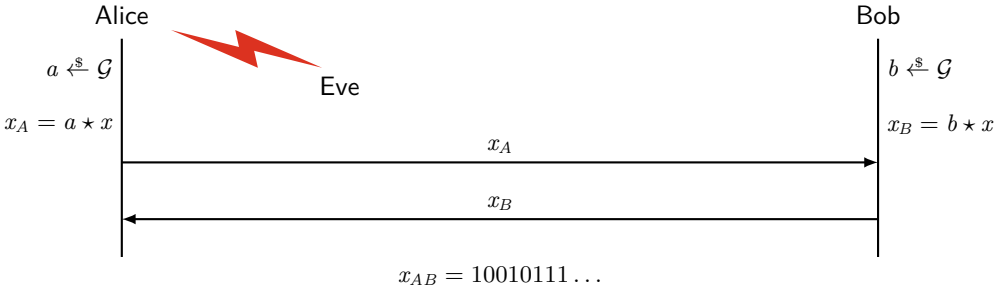
A **group action** is a map

$$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}, \qquad (g, x) \mapsto g \star x$$

for group $\mathcal{G}$ and set $\mathcal{X}$.

Example

Exponentiation:	(a, h)	$\mapsto h^a$
Scalar multiplication:	(m, P)	$\mapsto [m]P$
Isogenies:	$([\mathfrak{a}], E)$	$\mapsto E/E[\mathfrak{a}]$



Motivation

Definition

A **group action** is a map

$$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}, \qquad (g, x) \mapsto g \star x$$

for group $\mathcal{G}$ and set $\mathcal{X}$.

Example

Exponentiation:	$(a, h) \mapsto$	h^a
Scalar multiplication:	$(m, P) \mapsto$	$[m]P$
Isogenies:	$([\mathfrak{a}], E) \mapsto$	$E/E[\mathfrak{a}]$

Hidden Number Problem

Given: Public elements x_A, x_B and access to a CDH-like oracle with fixed input x_A

$$\mathcal{O}_k(y) \coloneqq \mathsf{MSB}_k(a \star y)$$

Task: Recover $x_{AB} = \mathsf{CDH}(x_A, x_B)$

Motivation

Definition

A **group action** is a map

$$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}, \qquad (g, x) \mapsto g \star x$$

for group $\mathcal{G}$ and set $\mathcal{X}$.

Example

Exponentiation:	$(a, h) \mapsto$	h^a
Scalar multiplication:	$(m, P) \mapsto$	$[m]P$
Isogenies:	$([\mathfrak{a}], E) \mapsto$	$E/E[\mathfrak{a}]$

Hidden Number Problem

Given: Public elements x_A, x_B and access to a CDH-like oracle with fixed input x_A

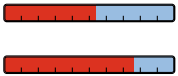
$$\mathcal{O}_k(y) \coloneqq \mathsf{MSB}_k(a \star y)$$

Task: Recover $x_{AB} = \mathsf{CDH}(x_A, x_B)$

- Solutions for DH [BV96], EC-DH [BHH01] and SIDH [GPST16]
- CSIDH [CLM+18] and CSURF [CD20] are popular post-quantum secure group actions, but **no results** on HNP

Contributions

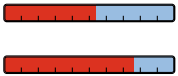
1. Efficient heuristic PPT algorithm solving HNP for
   - CSIDH: $k \approx 0.54 \log p$
   - CSURF: $k \approx 0.76 \log p$



   - Recovery works on any continuous block of missing bits
   - Only 3 queries to the oracle

Contributions

1. Efficient heuristic PPT algorithm solving HNP for
   - CSIDH: $k \approx 0.54 \log p$
   - CSURF: $k \approx 0.76 \log p$



   - Recovery works on any continuous block of missing bits
   - Only 3 queries to the oracle
2. Complete automation of Coppersmith's method
   - Prior results required a lot of manual work
   - Lattice optimization can be seen as a combinatorial problem that can be solved efficiently
   - Implementation in Sage

CSIDH and CSURF

Group Action

$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$

CSIDH and CSURF

- Set elements are elliptic curves in **Montgomery form**

$$E_M : y^2 = x^3 + Mx^2 + x, \qquad M \in \mathbb{F}_p^* \setminus \{\pm 2\}$$

Group Action

$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$

CSIDH and CSURF

- Set elements are elliptic curves in **Montgomery form**

$$E_M : y^2 = x^3 + Mx^2 + x, \qquad M \in \mathbb{F}_p^* \setminus \{\pm 2\}$$

- There is a group $\mathcal{G} = \langle g_1, \ldots, g_n \rangle$ that acts on these curves via **isogenies**

Group Action

$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$

CSIDH and CSURF

- Set elements are elliptic curves in **Montgomery form**

$$E_M : y^2 = x^3 + Mx^2 + x, \qquad M \in \mathbb{F}_p^* \setminus \{\pm 2\}$$

Group Action

$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$

- There is a group $\mathcal{G} = \langle g_1, \ldots, g_n \rangle$ that acts on these curves via **isogenies**
- For the generators g_i there exist efficient formulas that compute the group action

	CSIDH	CSURF
g_1	not available	$M' = (M+6)/(2\sqrt{M+2})$
g_2	$M' = -6x_P^3 + Mx_P^2 + 6x_P$	
g_3	$M' = x_P^2 x_{2P}^2 (M - 6(x_P + x_{2P} - x_P^{-1} - x_{2P}^{-1}))$	
$\vdots$	$\vdots$	

CSIDH and CSURF

- Set elements are elliptic curves in **Montgomery form**

$$E_M : y^2 = x^3 + Mx^2 + x, \qquad M \in \mathbb{F}_p^* \setminus \{\pm 2\}$$

Group Action

$\star : \mathcal{G} \times \mathcal{X} \to \mathcal{X}$

- There is a group $\mathcal{G} = \langle g_1, \ldots, g_n \rangle$ that acts on these curves via **isogenies**

- For the generators g_i there exist efficient formulas that compute the group action

	CSIDH	CSURF
g_1	not available	$M' = (M+6)/(2\sqrt{M+2})$
g_2	$M' = -6x_P^3 + Mx_P^2 + 6x_P$	
g_3	$M' = x_P^2 x_{2P}^2 (M - 6(x_P + x_{2P} - x_P^{-1} - x_{2P}^{-1}))$	
$\vdots$	$\vdots$	

- In practice: iteratively apply the action of the generators to get $E_{M'} = \left(\prod_{i=1}^{n} g_i^{e_i}\right) \star E_M$

Strategy

Hidden Number Problem

Given: Public curves E_A, E_B and oracle

$$\mathcal{O}_k(E) \coloneqq \mathsf{MSB}_k(a \star E)$$

Task: Recover E_{AB}

Strategy

Hidden Number Problem

Given: Public curves E_A, E_B and oracle

$$\mathcal{O}_k(E) \coloneqq \mathsf{MSB}_k(a \star E)$$

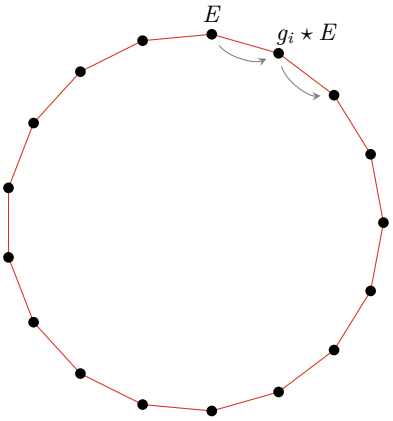
Task: Recover E_{AB}

Observation

For a fixed generator g_i the repeated action

$$g_i \star E, \quad g_i^2 \star E, \quad g_i^3 \star E, \ \ldots$$

forms a **cycle**[1].



[1] More precisely, there could be *multiple* disjoint cycles.

Strategy

Hidden Number Problem

Given: Public curves E_A, E_B and oracle

$$\mathcal{O}_k(E) \coloneqq \mathsf{MSB}_k(a \star E)$$

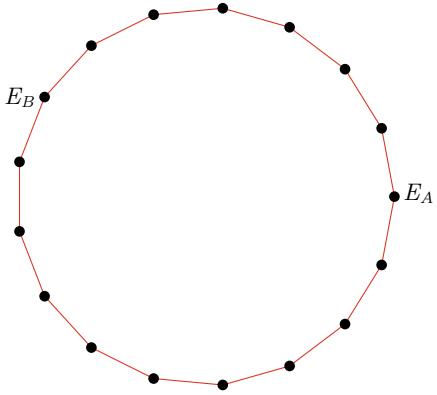
Task: Recover E_{AB}

Observation

For a fixed generator g_i the repeated action

$$g_i \star E, \quad g_i^2 \star E, \quad g_i^3 \star E, \ \ldots$$

forms a **cycle**[1].



[1] More precisely, there could be *multiple* disjoint cycles.

Strategy

Hidden Number Problem

Given: Public curves E_A, E_B and oracle

$$\mathcal{O}_k(E) \coloneqq \mathsf{MSB}_k(a \star E)$$

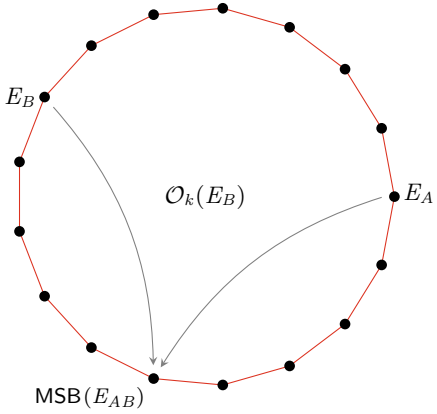
Task: Recover E_{AB}

Observation

For a fixed generator g_i the repeated action

$$g_i \star E, \quad g_i^2 \star E, \quad g_i^3 \star E, \ \ldots$$

forms a **cycle**[1].



[1] More precisely, there could be *multiple* disjoint cycles.

Strategy

Hidden Number Problem

Given: Public curves E_A, E_B and oracle

$$\mathcal{O}_k(E) \coloneqq \mathsf{MSB}_k(a \star E)$$

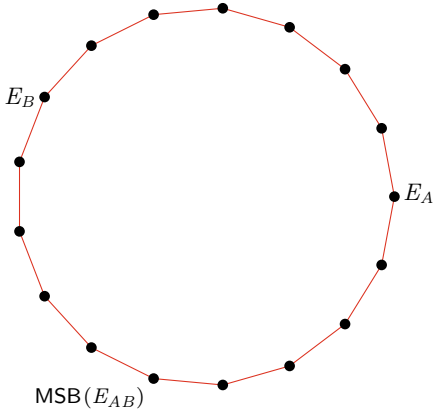
Task: Recover E_{AB}

Observation

For a fixed generator g_i the repeated action

$$g_i \star E, \quad g_i^2 \star E, \quad g_i^3 \star E, \ \ldots$$

forms a **cycle**[1].



[1] More precisely, there could be *multiple* disjoint cycles.

Strategy

Hidden Number Problem

Given: Public curves E_A, E_B and oracle

$$\mathcal{O}_k(E) \coloneqq \mathsf{MSB}_k(a \star E)$$

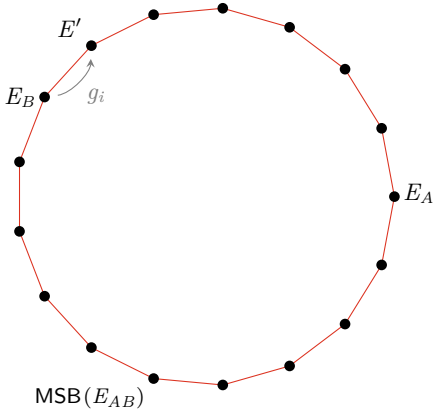
Task: Recover E_{AB}

Observation

For a fixed generator g_i the repeated action

$$g_i \star E, \quad g_i^2 \star E, \quad g_i^3 \star E, \ \ldots$$

forms a **cycle**[1].



[1] More precisely, there could be *multiple* disjoint cycles.

Strategy

Hidden Number Problem

Given: Public curves E_A, E_B and oracle

$$\mathcal{O}_k(E) \coloneqq \mathsf{MSB}_k(a \star E)$$

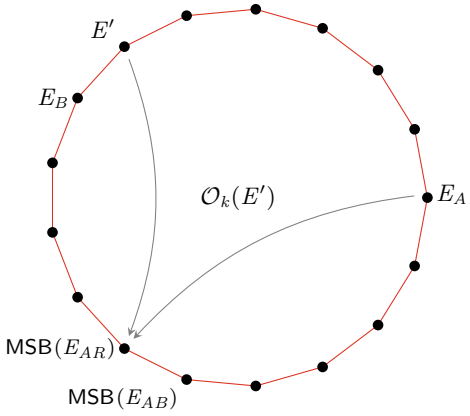
Task: Recover E_{AB}

Observation

For a fixed generator g_i the repeated action

$$g_i \star E, \quad g_i^2 \star E, \quad g_i^3 \star E, \ \ldots$$

forms a **cycle**[1].



[1] More precisely, there could be *multiple* disjoint cycles.

Strategy

Hidden Number Problem

Given: Public curves E_A, E_B and oracle

$$\mathcal{O}_k(E) \coloneqq \mathsf{MSB}_k(a \star E)$$

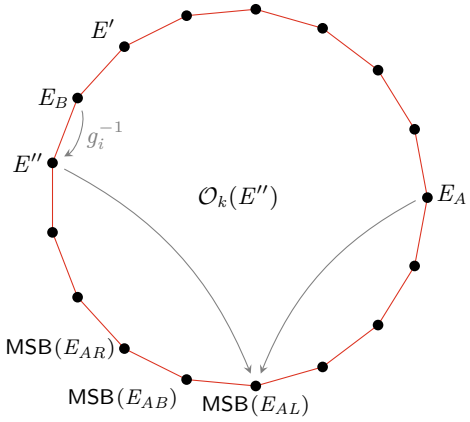
Task: Recover E_{AB}

Observation

For a fixed generator g_i the repeated action

$$g_i \star E, \quad g_i^2 \star E, \quad g_i^3 \star E, \ \ldots$$

forms a **cycle**[1].



[1] More precisely, there could be *multiple* disjoint cycles.

Strategy

Hidden Number Problem

Given: Public curves E_A, E_B and oracle

$$\mathcal{O}_k(E) \coloneqq \mathsf{MSB}_k(a \star E)$$

Task: Recover E_{AB}

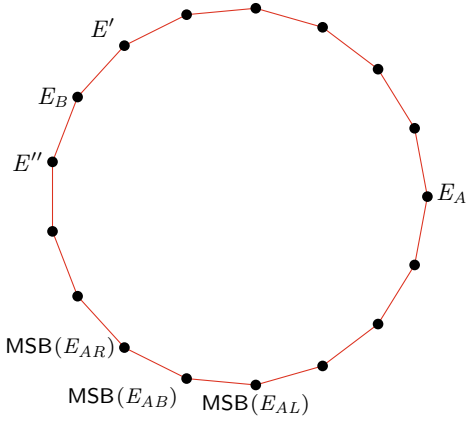
Observation

For a fixed generator g_i the repeated action

$$g_i \star E, \quad g_i^2 \star E, \quad g_i^3 \star E, \ \ldots$$

forms a **cycle**[1].

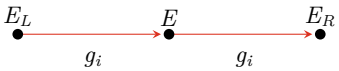
☞ By abusing **commutativity** we can explore the neighborhood of E_{AB}



[1] More precisely, there could be *multiple* disjoint cycles.

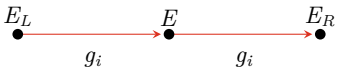
Neighborhood Relation

CSURF



Neighborhood Relation

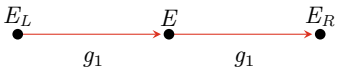
CSURF



	CSIDH	CSURF
g_1	not available	$M' = (M+6)/(2\sqrt{M+2})$
g_2	$M' = -6x_P^3 + Mx_P^2 + 6x_P$	
g_3	$M' = x_P^2 x_{2P}^2 (M - 6(x_P + x_{2P} - x_P^{-1} - x_{2P}^{-1}))$	
$\vdots$	$\vdots$	

Neighborhood Relation

CSURF

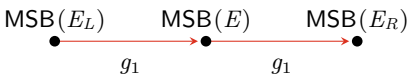


From the group element g_1 we get a **system of equations**:

$$M_L^2 + 12M_L - 4M^2M_L - 8M^2 + 36 \equiv 0 \bmod p$$
$$M^2 + 12M - 4M_R^2M - 8M_R^2 + 36 \equiv 0 \bmod p$$

Neighborhood Relation

CSURF



From the group element g_1 we get a **system of equations**:

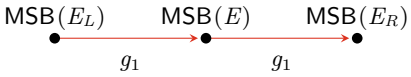
$$M_L^2 + 12M_L - 4M^2M_L - 8M^2 + 36 \equiv 0 \bmod p$$
$$M^2 + 12M - 4M_R^2M - 8M_R^2 + 36 \equiv 0 \bmod p$$

Observation

$$M = \underbrace{M_{\text{TOP}} \cdot 2^{n-k}}_{\text{MSBs, shifted}} + \underbrace{M_{\text{BOT}}}_{\text{Unknown LSBs}} \quad \text{over } \mathbb{F}_p$$

Neighborhood Relation

CSURF



From the group element g_1 we get a **system of equations**:

$$M_L^2 + 12M_L - 4M^2M_L - 8M^2 + 36 \equiv 0 \bmod p$$
$$M^2 + 12M - 4M_R^2M - 8M_R^2 + 36 \equiv 0 \bmod p$$

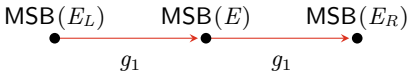
☞ The system has a **small** solution in the missing bits of E_L, E, E_R

Observation

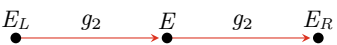
$$M = \underbrace{M_{\text{TOP}} \cdot 2^{n-k}}_{\text{MSBs, shifted}} + \underbrace{M_{\text{BOT}}}_{\text{Unknown LSBs}} \quad \text{over } \mathbb{F}_p$$

Neighborhood Relation

CSURF



CSIDH



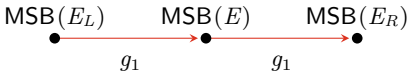
From the group element g_1 we get a **system of equations**:

$$M_L^2 + 12M_L - 4M^2M_L - 8M^2 + 36 \equiv 0 \bmod p$$
$$M^2 + 12M - 4M_R^2M - 8M_R^2 + 36 \equiv 0 \bmod p$$

☞ The system has a **small** solution in the missing bits of E_L, E, E_R

Neighborhood Relation

CSURF

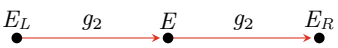


From the group element g_1 we get a **system of equations**:

$$M_L^2 + 12M_L - 4M^2M_L - 8M^2 + 36 \equiv 0 \bmod p$$
$$M^2 + 12M - 4M_R^2M - 8M_R^2 + 36 \equiv 0 \bmod p$$

☞ The system has a **small** solution in the missing bits of E_L, E, E_R

CSIDH

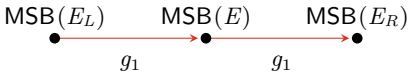


There exists a group element g' of **order 3**

$$g' = \prod g_i$$

Neighborhood Relation

CSURF

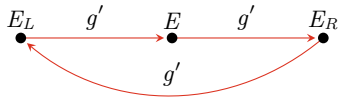


From the group element g_1 we get a **system of equations**:

$$M_L^2 + 12M_L - 4M^2M_L - 8M^2 + 36 \equiv 0 \bmod p$$
$$M^2 + 12M - 4M_R^2M - 8M_R^2 + 36 \equiv 0 \bmod p$$

☞ The system has a **small** solution in the missing bits of E_L, E, E_R

CSIDH

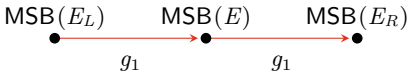


There exists a group element g' of **order 3**

$$g' = \prod g_i$$

Neighborhood Relation

CSURF

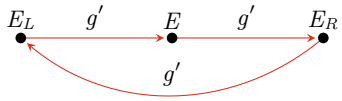


From the group element g_1 we get a **system of equations**:

$$M_L^2 + 12M_L - 4M^2M_L - 8M^2 + 36 \equiv 0 \bmod p$$
$$M^2 + 12M - 4M_R^2M - 8M_R^2 + 36 \equiv 0 \bmod p$$

☞ The system has a **small** solution in the missing bits of E_L, E, E_R

CSIDH



There exists a group element g' of **order 3**

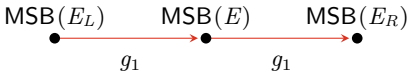
$$g' = \prod g_i$$

Action of g' [OT20]

$$M' = 2\frac{M-6}{M+2}$$

Neighborhood Relation

CSURF

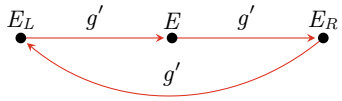


From the group element g_1 we get a **system of equations**:

$$M_L^2 + 12M_L - 4M^2M_L - 8M^2 + 36 \equiv 0 \bmod p$$
$$M^2 + 12M - 4M_R^2M - 8M_R^2 + 36 \equiv 0 \bmod p$$

☞ The system has a **small** solution in the missing bits of E_L, E, E_R

CSIDH



There exists a group element g' of **order 3**

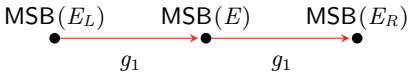
$$g' = \prod g_i$$

We get the following **system of equations**:

$$2M - MM_L - 2M_L - 12 \equiv 0 \bmod p$$
$$2M_R - MM_R - 2M - 12 \equiv 0 \bmod p$$
$$2M_L - M_LM_R - 2M_R - 12 \equiv 0 \bmod p$$

Neighborhood Relation

CSURF



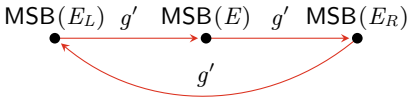
From the group element g_1 we get a **system of equations**:

$$M_L^2 + 12M_L - 4M^2M_L - 8M^2 + 36 \equiv 0 \bmod p$$
$$M^2 + 12M - 4M_R^2M - 8M_R^2 + 36 \equiv 0 \bmod p$$

☞ The system has a **small** solution in the missing bits of E_L, E, E_R

☞ Apply the same trick to get a system with a small solution

CSIDH



There exists a group element g' of **order 3**

$$g' = \prod g_i$$

We get the following **system of equations**:

$$2M - MM_L - 2M_L - 12 \equiv 0 \bmod p$$
$$2M_R - MM_R - 2M - 12 \equiv 0 \bmod p$$
$$2M_L - M_LM_R - 2M_R - 12 \equiv 0 \bmod p$$

Coppersmith's Method - Automated

Coppersmith-type Problem

Given:

- Polynomials $f_1, \ldots, f_n \in \mathbb{Z}[x_1, \ldots, x_k]$,
- bounds $X_1, \ldots, X_k \in \mathbb{N}$,
- modulus $p \in \mathbb{N}$.

Find:

- All **small roots** $r = (r_1, \ldots, r_k) \in \mathbb{Z}^k$ with
  - $f_i(r_1, \ldots, r_k) \equiv 0 \bmod p$, and
  - $|r_i| \leq X_i$.

Coppersmith's Method - Automated

Coppersmith-type Problem

Given:

- Polynomials $f_1, \ldots, f_n \in \mathbb{Z}[x_1, \ldots, x_k]$,
- bounds $X_1, \ldots, X_k \in \mathbb{N}$,
- modulus $p \in \mathbb{N}$.

Find:

- All **small roots** $r = (r_1, \ldots, r_k) \in \mathbb{Z}^k$ with
  - $f_i(r_1, \ldots, r_k) \equiv 0 \bmod p$, and
  - $|r_i| \leq X_i$.

Strategy:

- Fix $m \in \mathbb{N}$ and define **shift-polynomials**

$$f_1^{i_1} \cdot \ldots \cdot f_n^{i_n} \cdot x_1^{j_1} \cdot \ldots \cdot x_k^{j_k} \cdot p^{m \cdot n - (i_1 + \ldots + i_n)}$$

- Select **suitable** subset $\mathcal{F}$ of shift-polynomials and use it to construct a lattice
- Use **lattice reduction** (LLL) to find $(r_1, \ldots, r_k)$

☞ The performance **strongly** depends on the choice for $\mathcal{F}$

Coppersmith's Method - Automated

Coppersmith-type Problem

Given:

- Polynomials $f_1, \ldots, f_n \in \mathbb{Z}[x_1, \ldots, x_k]$,
- bounds $X_1, \ldots, X_k \in \mathbb{N}$,
- modulus $p \in \mathbb{N}$.

Find:

- All **small roots** $r = (r_1, \ldots, r_k) \in \mathbb{Z}^k$ with
  - $f_i(r_1, \ldots, r_k) \equiv 0 \bmod p$, and
  - $|r_i| \leq X_i$.

Strategy:

- Fix $m \in \mathbb{N}$ and define **shift-polynomials**

$$f_1^{i_1} \cdot \ldots \cdot f_n^{i_n} \cdot x_1^{j_1} \cdot \ldots \cdot x_k^{j_k} \cdot p^{m \cdot n - (i_1 + \ldots + i_n)}$$

- Select **suitable** subset $\mathcal{F}$ of shift-polynomials and use it to construct a lattice
- Use **lattice reduction** (LLL) to find $(r_1, \ldots, r_k)$

☞ The performance **strongly** depends on the choice for $\mathcal{F}$

Problem

How to select the shift-polynomials in $\mathcal{F}$?

Problem

How large can the bounds $X_1, \ldots, X_k$ be?

Coppersmith's Method - Automated

Coppersmith-type Problem

Given:

- Polynomials $f_1, \ldots, f_n \in \mathbb{Z}[x_1, \ldots, x_k]$,
- bounds $X_1, \ldots, X_k \in \mathbb{N}$,
- modulus $p \in \mathbb{N}$.

Find:

- All **small roots** $r = (r_1, \ldots, r_k) \in \mathbb{Z}^k$ with
  - $f_i(r_1, \ldots, r_k) \equiv 0 \bmod p$, and
  - $|r_i| \leq X_i$.

Strategy:

- Fix $m \in \mathbb{N}$ and define **shift-polynomials**

$$f_1^{i_1} \cdot \ldots \cdot f_n^{i_n} \cdot x_1^{j_1} \cdot \ldots \cdot x_k^{j_k} \cdot p^{m \cdot n - (i_1 + \ldots + i_n)}$$

- Select **suitable** subset $\mathcal{F}$ of shift-polynomials and use it to construct a lattice
- Use **lattice reduction** (LLL) to find $(r_1, \ldots, r_k)$

☞ The performance **strongly** depends on the choice for $\mathcal{F}$

Observation 1

$\mathcal{F}$ only depends on a **single** shift-polynomial (for which there even exists a canonical candidate).

☞ Given a single shift-polynomial, a locally optimal $\mathcal{F}$ can be constructed **automatically**

Problem

How large can the bounds $X_1, \ldots, X_k$ be?

Coppersmith's Method - Automated

Coppersmith-type Problem

Given:

- Polynomials $f_1, \ldots, f_n \in \mathbb{Z}[x_1, \ldots, x_k]$,
- bounds $X_1, \ldots, X_k \in \mathbb{N}$,
- modulus $p \in \mathbb{N}$.

Find:

- All **small roots** $r = (r_1, \ldots, r_k) \in \mathbb{Z}^k$ with
  - $f_i(r_1, \ldots, r_k) \equiv 0 \bmod p$, and
  - $|r_i| \leq X_i$.

Strategy:

- Fix $m \in \mathbb{N}$ and define **shift-polynomials**

$$f_1^{i_1} \cdot \ldots \cdot f_n^{i_n} \cdot x_1^{j_1} \cdot \ldots \cdot x_k^{j_k} \cdot p^{m \cdot n - (i_1 + \ldots + i_n)}$$

- Select **suitable** subset $\mathcal{F}$ of shift-polynomials and use it to construct a lattice
- Use **lattice reduction** (LLL) to find $(r_1, \ldots, r_k)$

☞ The performance **strongly** depends on the choice for $\mathcal{F}$

Observation 1

$\mathcal{F}$ only depends on a **single** shift-polynomial (for which there even exists a canonical candidate).

☞ Given a single shift-polynomial, a locally optimal $\mathcal{F}$ can be constructed **automatically**

Observation 2

We can use **polynomial interpolation** to derive bounds $X_1, \ldots, X_k$ automatically.

Summary

- We solved the **Hidden Number Problem** for CSIDH and CSURF
  - CSIDH: Given 54%, recover remaining 46% 
  - CSURF: Given 76%, recover remaining 24% 
- Recovery strongly benefits from a **small order subgroup** in the CSIDH setting
- **Combinatorial reformulation** of Coppersmith that allows for complete automation
- Open source **Sage implementation** available at `github.com/juliannowakowski/automated-coppersmith`

`https://ia.cr/2023/1409`

References I

[BHH01] Dan Boneh, Shai Halevi, and Nick Howgrave-Graham. The modular inversion hidden number problem. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 36–51. Springer, Heidelberg, December 2001.

[BV96] Dan Boneh and Ramarathnam Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In Neal Koblitz, editor, *CRYPTO'96*, volume 1109 of *LNCS*, pages 129–142. Springer, Heidelberg, August 1996.

[CD20] Wouter Castryck and Thomas Decru. CSIDH on the surface. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 111–129. Springer, Heidelberg, 2020.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An efficient post-quantum commutative group action. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part III*, volume 11274 of *LNCS*, pages 395–427. Springer, Heidelberg, December 2018.

[GPST16] Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 63–91. Springer, Heidelberg, December 2016.

References II

[OT20] Hiroshi Onuki and Tsuyoshi Takagi. On collisions related to an ideal class of order 3 in CSIDH. In Kazumaro Aoki and Akira Kanaoka, editors, *IWSEC 20*, volume 12231 of *LNCS*, pages 131–148. Springer, Heidelberg, September 2020.