Disorientation faults in CSIDH

28 November 2023
Séminaire de Théorie Algorithmique des Nombres
Institut de Mathématiques de Bordeaux

Gustavo Banegas

Juliane Krämer

Tanja Lange

Michael Meyer

Lorenz Panny

Krijn Reijnders

Jana Sotáková

Monika Trimoska


EINDHOVEN UNIVERSITY OF TECHNOLOGY

Fault attacks

Nixdorf, CC BY-SA 3.0

Physical attacks: trigger an error during the execution of sensitive computations; infer secret information from faulty outputs;

Takeaway:

- We propose lightweight countermeasures.
- The security of CSIDH is not compromised.

Key exchange

- Alice and Bob start on a common public node on a graph.
- They can not compute the whole graph, but they can *walk* on it $\rightarrow$ compute a step and see on which node we arrive.
- A path on the graph:

3	5	7	11	13
1	-1	0	2	0

$\longleftarrow$ "size" of step

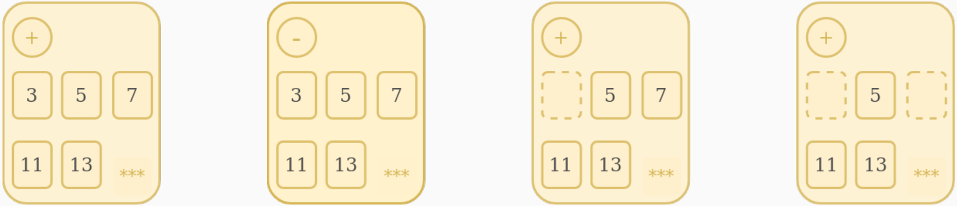
$\longleftarrow$ number of steps

- Goal: walk on the graph and end up on a common secret node.
- The catch: walking on the graph is commutative: the order in which the steps are taken does not matter, only the number of steps of each ~~size~~ degree.

Walking on the graph

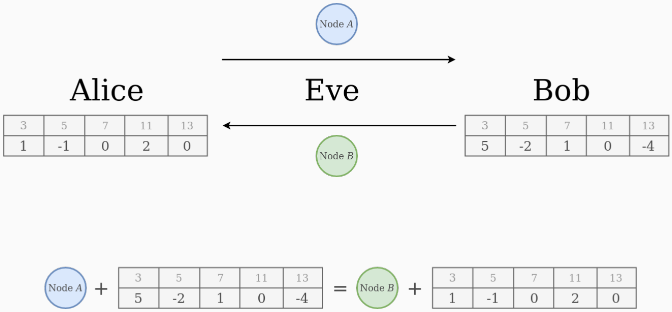
Magic box

Cards with instructions on how to compute steps.



- Some cards are for walking in the positive, and some are for walking in the negative direction.
- Some cards are missing instructions for certain steps (unlucky).

Key exchange



- Eve will relay the messages between Alice and Bob.

Node A →

Alice | Eve | Bob

Alice:

3	5	7	11	13
1	-1	0	2	0

← Node B

Bob:

3	5	7	11	13
5	-2	1	0	-4

Node A +

3	5	7	11	13
5	-2	1	0	-4

= Node B +

3	5	7	11	13
1	-1	0	2	0

- Eve will relay the messages between Alice and Bob.
- She brings the magic box.

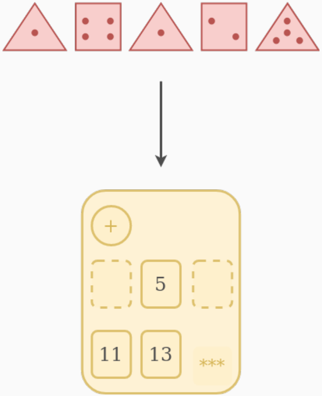
Computing the secret path

Alice gets a card with instructions.

Computing the secret path

Alice gets a card with instructions.

Computing the secret path



Cheat sheet

- Alice rolls 74 dice. Each dice has ℓ_i sides for $\ell_i \in \{3, 5, \ldots, 377, 587\}$.
- Getting a 'one' on the dice with ℓ_i sides : Alice gets a card *without* instructions for making ℓ_i-steps.
- Getting anything else: Alice gets a card *with* instructions for making ℓ_i-steps. Instructions are either for positive or negative steps, both with equal probability.
- Alice can compute all or some of the steps that she gets instructions for. Each step is computed at most once.
- **Round**: the process from rolling the dice to computing all possible steps.
- Alice performs as many rounds as she needs to compute all steps from the secret key.

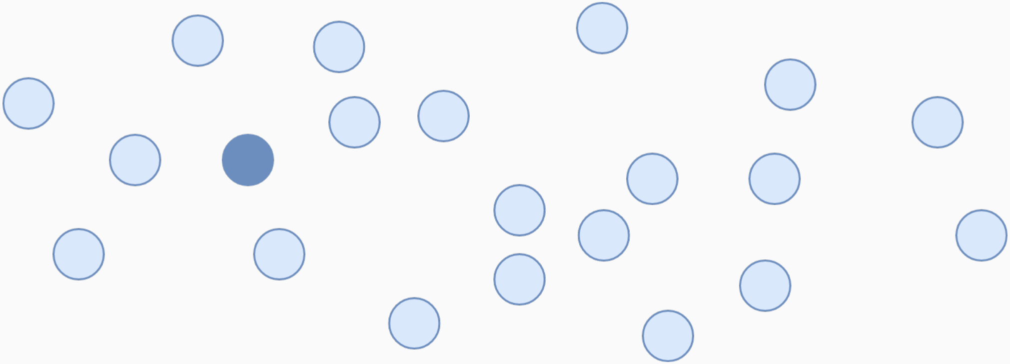
Computing the secret path (example)

Alice's secret key

3	5	7	11	13
1	-1	0	3	0

Left to compute

3	5	7	11	13
1	-1	0	3	0



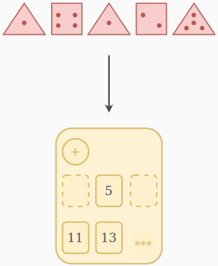
Computing the secret path (example)

Alice's secret key

3	5	7	11	13
1	-1	0	3	0

Left to compute

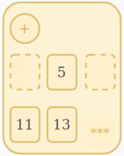
3	5	7	11	13
1	-1	0	3	0

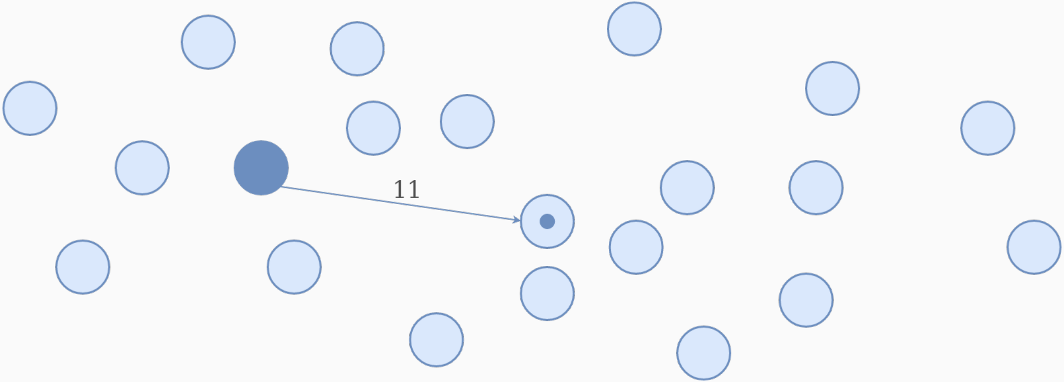


Computing the secret path (round 1)

Alice's secret key	3	5	7	11	13
	1	-1	0	3	0

Left to compute	3	5	7	11	13
	1	-1	0	2	0





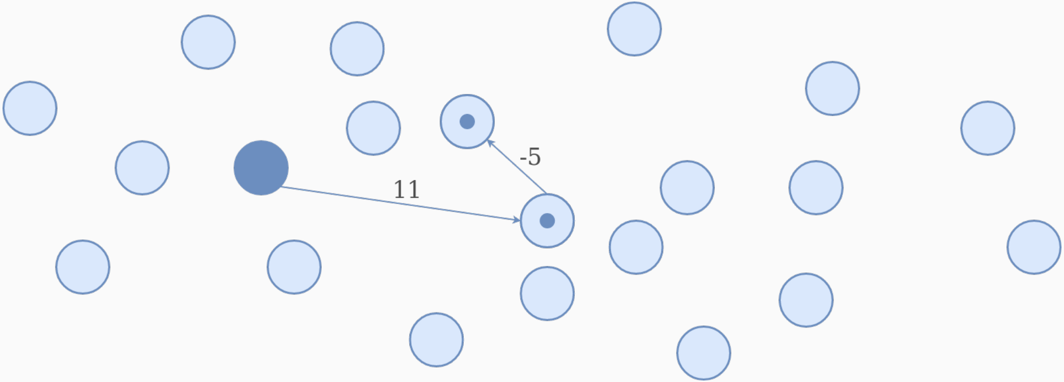
Computing the secret path (round 2)

Alice's secret key

3	5	7	11	13
1	-1	0	3	0

Left to compute

3	5	7	11	13
1	0	0	2	0



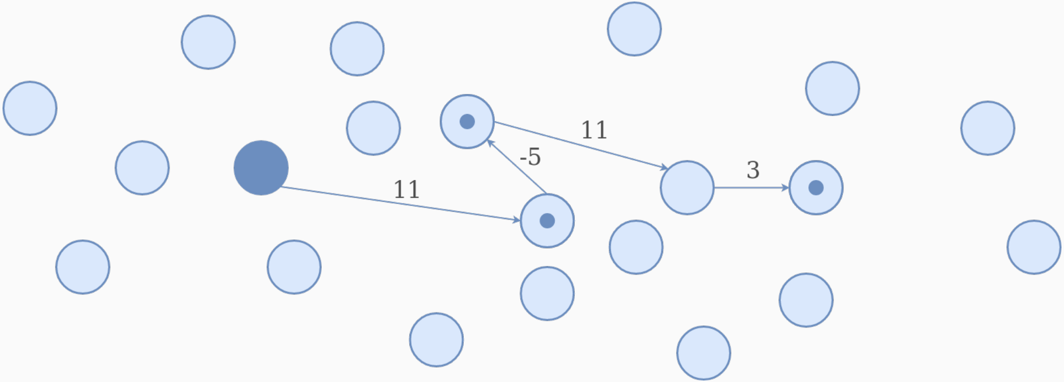
Computing the secret path (round 3)

Alice's secret key

3	5	7	11	13
1	-1	0	3	0

Left to compute

3	5	7	11	13
0	0	0	1	0



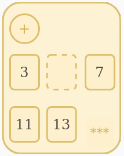
Computing the secret path (round 4)

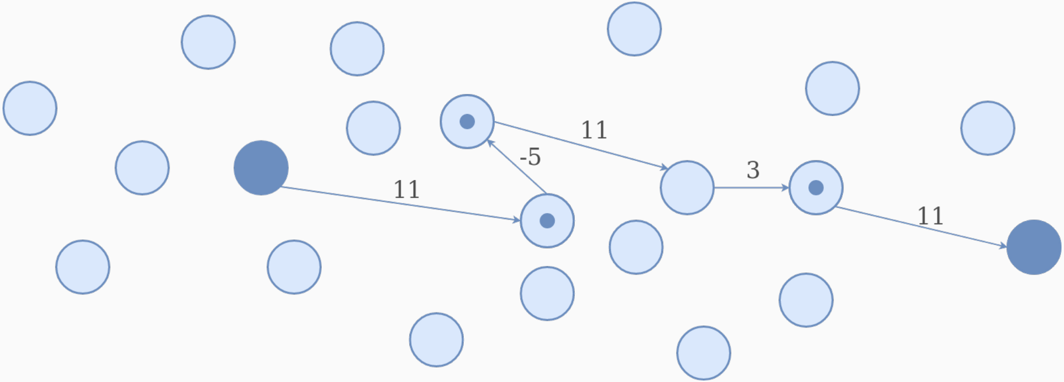
Alice's secret key

3	5	7	11	13
1	-1	0	3	0

Left to compute

3	5	7	11	13
0	0	0	0	0



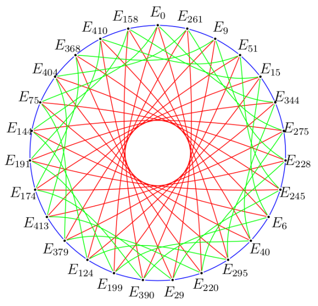


Cheat sheet

- Alice rolls 74 dice. Each dice has ℓ_i sides for $\ell_i \in \{3, 5, \ldots, 377, 587\}$.
- Getting a ‘one‘ on the dice with ℓ_i sides : Alice gets a card *without* instructions for making ℓ_i-steps.
- Getting anything else: Alice gets a card *with* instructions for making ℓ_i-steps. Instructions are either for positive or negative steps, both with equal probability.
- Alice can compute all or some of the steps that she gets instructions for. Each step is computed at most once.
- **Round**: the process from rolling the dice to computing all possible steps.
- Alice performs as many rounds as she needs to compute all steps from the secret key.

- An **isogeny** of elliptic curves is a non-zero map $E_1 \to E_2$
  - given by **rational functions**
  - that is a **group homomorphism**.
- **Degree** of a separable isogeny: the size of its kernel, aka number of points on E_1 mapping to the neutral element on E_2. Computing an isogeny of degree $\ell_i \to$ an ℓ_i-step.
- **CSIDH**: commutative group action suitable for non-interactive key exchange.

- Nodes $\to$ $\mathbb{F}_p$-isomorphism classes of supersingular elliptic curves.
  Edges $\to$ isogenies between them.
- CSIDH-512: $p = 4 \cdot \prod \ell_i - 1$, for $\ell_i \in \{3, 5, \ldots, 377, 587\}$ $\to$ we can compute ℓ_i-steps in the positive or in the negative direction, for all ℓ_i.
- Exponents $-5 \leq e_i \leq 5$ for all $1 \leq i \leq 74$.



Edges are 3, 5, and 7-isogenies. Image credit: Lorenz Panny.

Supersingular Isogeny Path problem

Given E_1 and E_2 two supersingular elliptic curves over $\mathbb{F}_p$, find and isogeny from E_1 to E_2.

Taking a **positive** ℓ_i-step.

(1) Find a point $(x, y) \in E$ of **order** ℓ_i with $x, y \in \mathbb{F}_p$.
The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
Sample a new point if you get ∞ (probability $1/\ell_i$).

(2) Compute the **isogeny** with **kernel** $\langle(x, y)\rangle$ using Vélu's formulas.

Taking a **positive** ℓ_i-step.

(1) Find a point $(x, y) \in E$ of **order** ℓ_i with $x, y \in \mathbb{F}_p$.
The order of any $(x, y) \in E$ divides $p + 1$, so $[(p + 1)/\ell_i](x, y) = \infty$ or a point of order ℓ_i.
Sample a new point if you get ∞ (probability $1/\ell_i$).

(2) Compute the **isogeny** with **kernel** $\langle (x, y) \rangle$ using Vélu's formulas.

Taking a **negative** ℓ_i-step.

(1) Find a point $(x, y) \in E$ of **order** ℓ_i with $x \in \mathbb{F}_p$ but $y \notin \mathbb{F}_p$.
Same test as above to find such a point.

(2) Compute the **isogeny** with **kernel** $\langle (x, y) \rangle$ using Vélu's formulas.

Algorithm 2: Evaluating the class-group action.

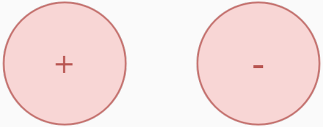
Input: $A \in \mathbb{F}_p$ and a list of integers $(e_1, \ldots, e_n)$.
Output: B such that $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]E_A = E_B$ (where $E_B \colon y^2 = x^3 + Bx^2 + x$).

While some $e_i \neq 0$ **do**
- Sample a random $x \in \mathbb{F}_p$.
- Set $s \leftarrow +1$ if $x^3 + Ax^2 + x$ is a square in $\mathbb{F}_p$, else $s \leftarrow -1$.
- Let $S = \{i \mid e_i \neq 0,\ \mathrm{sign}(e_i) = s\}$. **If** $S = \emptyset$ **then** start over with a new x.
- Let $k \leftarrow \prod_{i \in S} \ell_i$ and compute $Q \leftarrow [(p+1)/k]P$.
- **For each** $i \in S$ **do**
  - Compute $R \leftarrow [k/\ell_i]Q$. **If** $R = \infty$ **then** skip this i.
  - Compute an isogeny $\varphi \colon E_A \to E_B \colon y^2 = x^3 + Bx^2 + x$ with $\ker \varphi = R$.
  - Set $A \leftarrow B$, $Q \leftarrow \varphi(Q)$, $k \leftarrow k/\ell_i$, and finally $e_i \leftarrow e_i - s$.

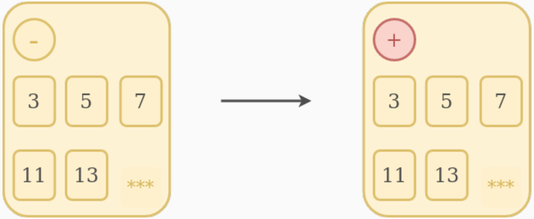
Return A.

- Alice rolls 74 dice. Each dice has ℓ_i sides for $\ell_i \in \{3, 5, \ldots, 377, 587\}$.
- Getting a 'one' on the dice with ℓ_i sides : Alice gets a card *without* instructions for making ℓ_i-steps.
- Getting anything else: Alice gets a card *with* instructions for making ℓ_i-steps. Instructions are either for positive or negative steps, both with equal probability.
- Alice can compute all or some of the steps that she gets instructions for. Each step is computed at most once.
- **Round**: the process from rolling the dice to computing all possible steps.
- Alice performs as many rounds as she needs to compute all steps from the secret key.

- You bring stickers to put over the direction sign on the cards.

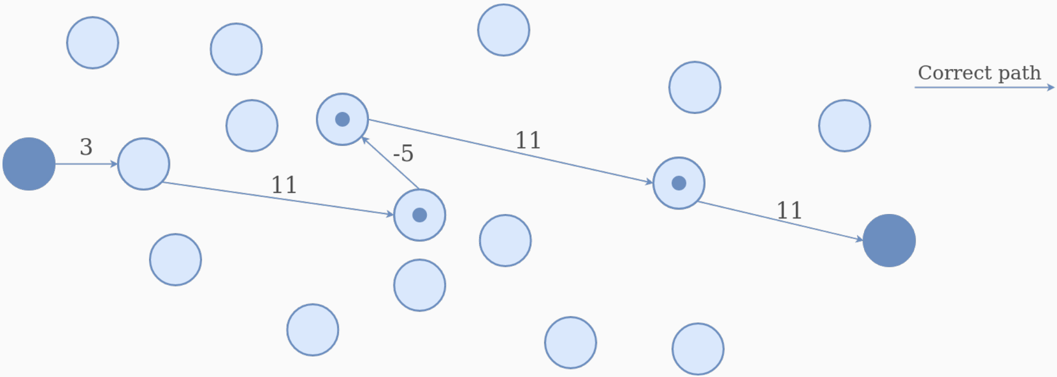


- Alice thinks she has a card with instructions for positive steps, but she has a card with instructions for negative steps.



Alice's secret key

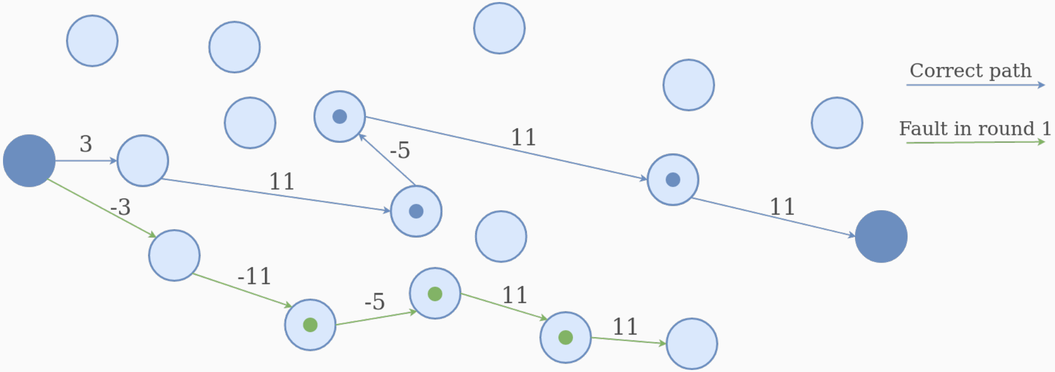
3	5	7	11	13
1	-1	0	3	0



Faulted paths

Alice's secret key

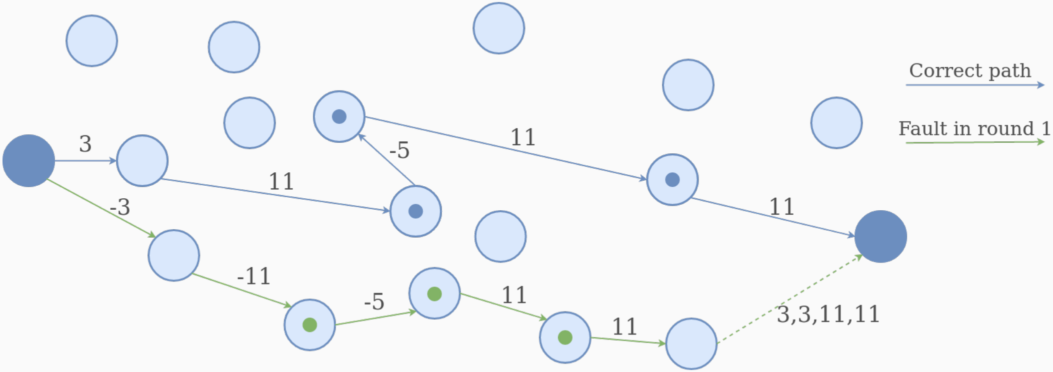
3	5	7	11	13
1	-1	0	3	0



Faulted paths

Alice's secret key

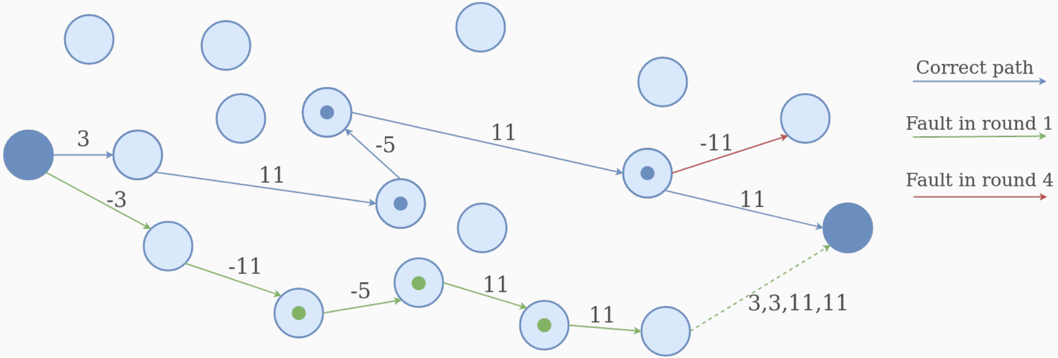
3	5	7	11	13
1	-1	0	3	0



Faulted paths

Alice's secret key

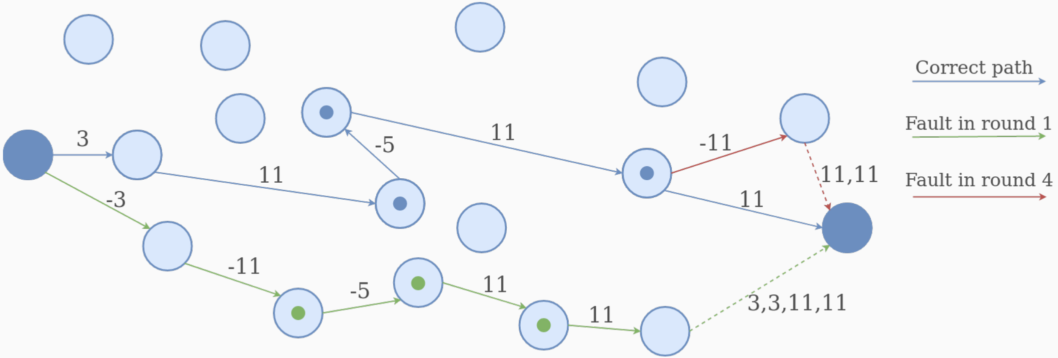
3	5	7	11	13
1	-1	0	3	0



Faulted paths

Alice's secret key

3	5	7	11	13
1	-1	0	3	0



Algorithm 2: Evaluating the class-group action.

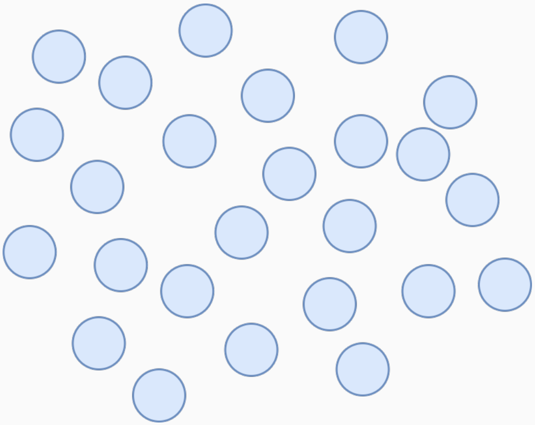
Input: $A \in \mathbb{F}_p$ and a list of integers $(e_1, \ldots, e_n)$.
Output: B such that $[\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}]E_A = E_B$ (where $E_B\colon y^2 = x^3 + Bx^2 + x$).

While some $e_i \neq 0$ **do**
- Sample a random $x \in \mathbb{F}_p$.
- Set $s \leftarrow +1$ if $x^3 + Ax^2 + x$ is a square in $\mathbb{F}_p$, else $s \leftarrow -1$.
- Let $S = \{i \mid e_i \neq 0,\ \mathrm{sign}(e_i) = s\}$. **If** $S = \emptyset$ **then** start over with a new x.
- Let $k \leftarrow \prod_{i \in S} \ell_i$ and compute $Q \leftarrow [(p+1)/k]P$.
- **For each** $i \in S$ **do**
  - Compute $R \leftarrow [k/\ell_i]Q$. **If** $R = \infty$ **then** skip this i.
  - Compute an isogeny $\varphi\colon E_A \to E_B\colon y^2 = x^3 + Bx^2 + x$ with $\ker \varphi = R$.
  - Set $A \leftarrow B$, $Q \leftarrow \varphi(Q)$, $k \leftarrow k/\ell_i$, and finally $e_i \leftarrow e_i - s$.

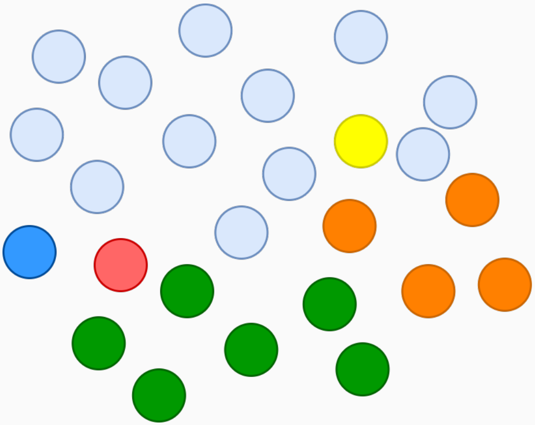
Return A.

Eve's graph (an exercice)

- **Strategy:** Collecting faulty output nodes from the first round, both from negative and positive steps.



- **Strategy:** Collecting faulty output nodes from the first round, both from negative and positive steps.

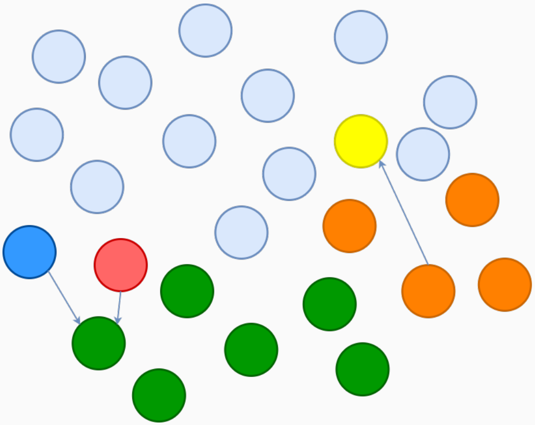


Eve's graph (an exercice)

`pubcrawl`

Eve's graph (an exercice)

- **Strategy:** Collecting faulty output nodes from the first round, both from negative and positive steps.



Takeaway

- Probability of having a missing ℓ_i torsion.
- **Incoming** arrow: missing torsion in **negative** coefficients
- **Outgoing** arrow: missing torsion in **positive** coefficients

Recovering the entire secret key

- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.

Recovering the entire secret key

- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.
- How?

- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.
- How?
  $\hookrightarrow$ perform fault injection always in round 5.

$$
\begin{array}{l}
- - - - (-) \\
+ + + + (-) \\
+ + - - (+) \\
- + - - (-)
\end{array}
$$

- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.
- How?
  $\hookrightarrow$ perform fault injection always in round 5.

$$\begin{array}{l} - - - - (-) \\ + + + + (-) \\ + + - - (+) \\ - + - - (-) \end{array} \qquad \longrightarrow \text{effective} - \text{round-5 curve}$$

Recovering the entire secret key

- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.
- How?
  $\hookrightarrow$ perform fault injection always in round 5.

$$
\begin{array}{ll}
- - - - (-) & \longrightarrow \text{effective} - \text{round-5 curve} \\
+ + + + (-) & \longrightarrow \text{effective} - \text{round-1 curve} \\
+ + - - (+) & \\
- + - - (-) &
\end{array}
$$

Recovering the entire secret key

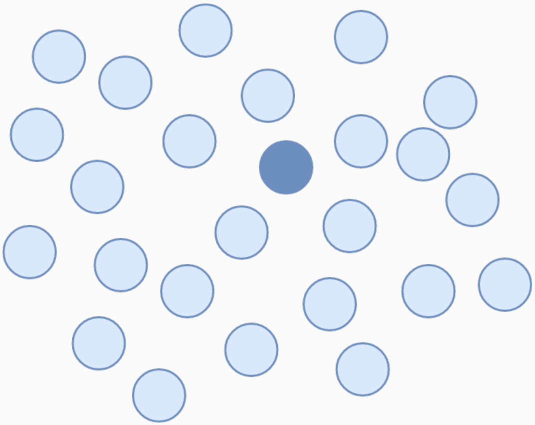
- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.
- How?
  $\hookrightarrow$ perform fault injection always in round 5.

$$
\begin{array}{ll}
- - - - (-) & \longrightarrow \text{effective } - \text{ round-5 curve} \\
+ + + + (-) & \longrightarrow \text{effective } - \text{ round-1 curve} \\
+ + - - (+) & \longrightarrow \text{effective } + \text{ round-3 curve} \\
- + - - (-) &
\end{array}
$$

- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.
- How?
  $\hookrightarrow$ perform fault injection always in round 5.

$- - - - (-)$	$\longrightarrow$ effective $-$ round-5 curve
$+ + + + (-)$	$\longrightarrow$ effective $-$ round-1 curve
$+ + - - (+)$	$\longrightarrow$ effective $+$ round-3 curve
$- + - - (-)$	$\longrightarrow$ effective $-$ round-4 curve

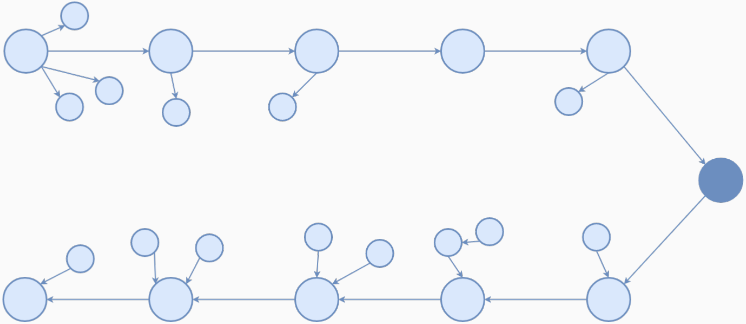
- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.



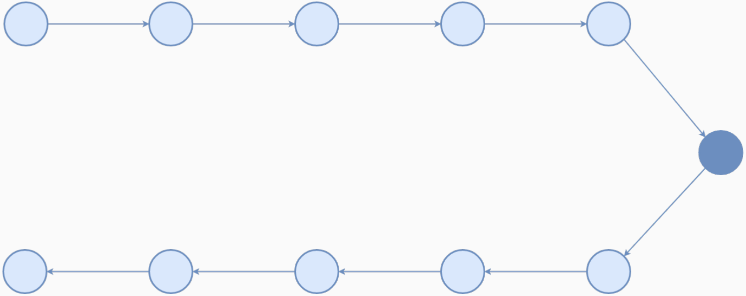
- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.

`pubcrawl`

- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.

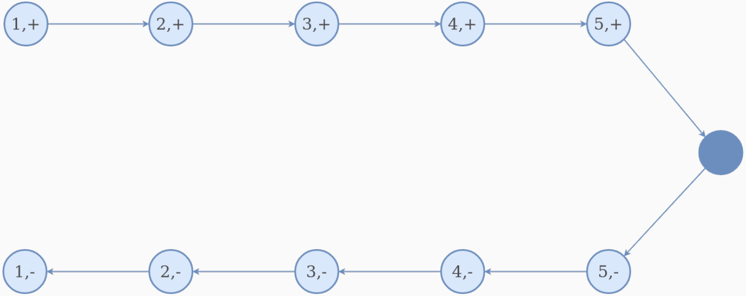


- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.

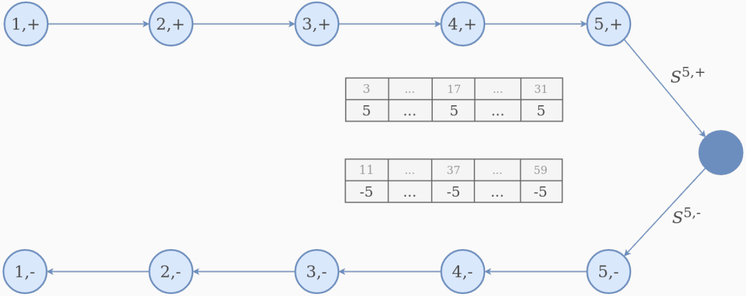


Eve's graph

- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.

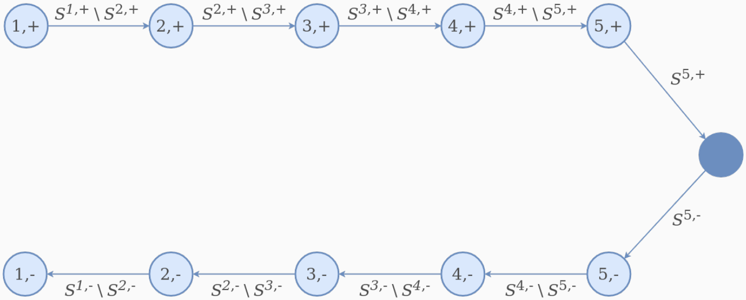


- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.



Eve's graph

- **Strategy:** Collecting faulty output nodes from the first 5 rounds, both from negative and positive steps.



How practical is it to obtain a fully connected graph?

- We have 74 primes, 10 gaps. Pigeon principle: at least one of the gaps is of distance at most 7.

How practical is it to obtain a fully connected graph?

- We have 74 primes, 10 gaps. Pigeon principle: at least one of the gaps is of distance at most 7.
- Reducing the search space I: when we find the orientation of some primes.

How practical is it to obtain a fully connected graph?

- We have 74 primes, 10 gaps. Pigeon principle: at least one of the gaps is of distance at most 7.
- Reducing the search space I: when we find the orientation of some primes.
- Reducing the search space II: when we find the coefficient of some primes.

Strategy I:

Minimum spanning tree search algorithm

Strategy I:

Minimum spanning tree search algorithm

(not very reliable)

Strategy I:

Minimum spanning tree search algorithm

(not very reliable)

Strategy II:

An isogenist with a pen

Strategy I:

Minimum spanning tree search algorithm

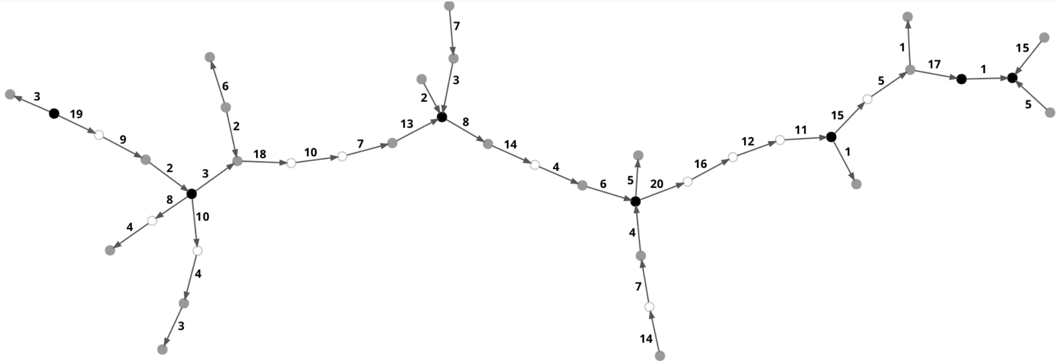
(not very reliable)

Strategy II:

An isogenist with a pen

$\hookrightarrow$ Demo

Toy example (CSIDH-103)



Actual example (CSIDH-512)

In the full paper

- Attack on CSIDH
- Attack on CTIDH
- Exploiting the twist
- Lightweight countermeasures

eprint: 2022/1202