Patient Zero & Patient Six: Zero-Value and Correlation Attacks on CSIDH and SIKE†

SAC 2022

25 August 2022

Fabio Campos[1,2] Michael Meyer[3] Krijn Reijnders[2] Marc Stöttinger[1]

[1]RheinMain University of Applied Sciences Wiesbaden, Germany

[2]Radboud University, Nijmegen, The Netherlands

[3]University of Regensburg, Germany

Outline

- Zero-value attacks
- Isogeny paths in CSIDH
- Vulnerable curves in CSIDH
- Attacking SQALE and CTIDH
- Countermeasures
- Applicability to SIKE

Motivation: Zero-value attacks

Zero-value attacks: Identify secret-dependent occurrences of zero-values in the power trace.

$\rightsquigarrow$ information on private key.

Motivation: Zero-value attacks

Zero-value attacks: Identify secret-dependent occurrences of zero-values in the power trace.
$\rightsquigarrow$ information on private key.

- Proposed for SIDH in [Koziel–Azarderakhsh–Jao-2017].

Motivation: Zero-value attacks

Zero-value attacks: Identify secret-dependent occurrences of zero-values in the power trace.
$\rightsquigarrow$ information on private key.

- Proposed for SIDH in [Koziel–Azarderakhsh–Jao-2017].
- Demonstrated for SIKE in [De Feo–El Mrabet–Genêt–Kaluđerović–de Guertechin–Pontié–Tasso-2022].

Motivation: Zero-value attacks

Zero-value attacks: Identify secret-dependent occurrences of zero-values in the power trace.
$\rightsquigarrow$ information on private key.

- Proposed for SIDH in [Koziel–Azarderakhsh–Jao-2017].
- Demonstrated for SIKE in [De Feo–El Mrabet–Genêt–Kaluđerović–de Guertechin–Pontié–Tasso-2022].
- Does this work in CSIDH too?

CSIDH

- Prime of the form $p = 4 \cdot \ell_1 \cdot \cdots \cdot \ell_n - 1$ with small distinct odd primes ℓ_i.

CSIDH

- Prime of the form $p = 4 \cdot \ell_1 \cdot \cdots \cdot \ell_n - 1$ with small distinct odd primes ℓ_i.
- Work with supersingular elliptic curves over $\mathbb{F}_p$.
  $\rightsquigarrow \#E(\mathbb{F}_p) = p + 1$ for all involved curves.

CSIDH

- Prime of the form $p = 4 \cdot \ell_1 \cdot \cdots \cdot \ell_n - 1$ with small distinct odd primes ℓ_i.
- Work with supersingular elliptic curves over $\mathbb{F}_p$.
  $\rightsquigarrow \#E(\mathbb{F}_p) = p + 1$ for all involved curves.
- We can efficiently compute isogenies of degrees ℓ_i.

CSIDH

- Prime of the form $p = 4 \cdot \ell_1 \cdot \cdots \cdot \ell_n - 1$ with small distinct odd primes ℓ_i.
- Work with supersingular elliptic curves over $\mathbb{F}_p$.
  $\rightsquigarrow \#E(\mathbb{F}_p) = p + 1$ for all involved curves.
- We can efficiently compute isogenies of degrees ℓ_i.
- Each of the ℓ_i-isogeny graphs consists of one or more cycles with identical vertex set.

CSIDH

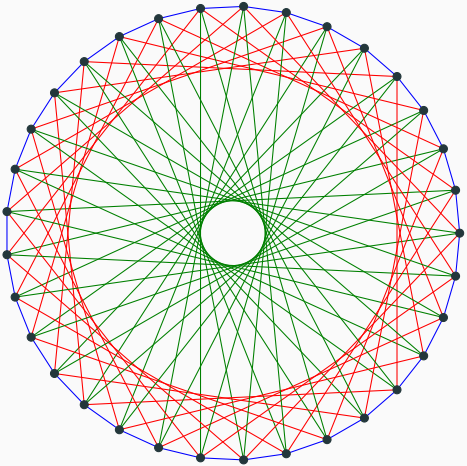
- Prime of the form $p = 4 \cdot \ell_1 \cdot \cdots \cdot \ell_n - 1$ with small distinct odd primes ℓ_i.
- Work with supersingular elliptic curves over $\mathbb{F}_p$.
  $\rightsquigarrow \#E(\mathbb{F}_p) = p + 1$ for all involved curves.
- We can efficiently compute isogenies of degrees ℓ_i.
- Each of the ℓ_i-isogeny graphs consists of one or more cycles with identical vertex set.
- CSIDH isogeny graph: union of these cycles

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

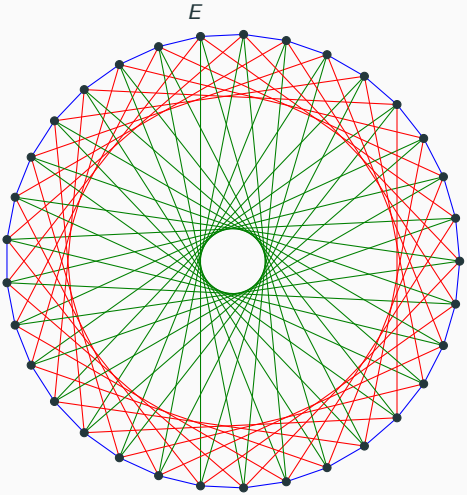
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

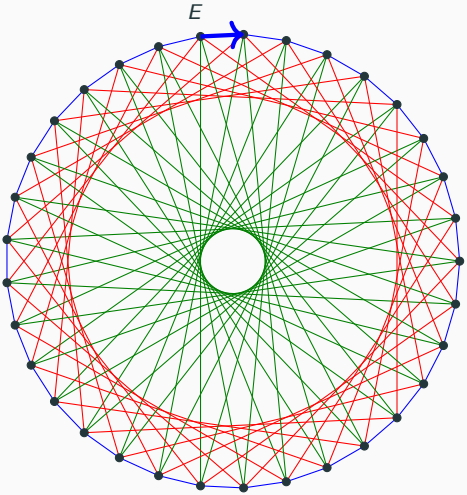
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

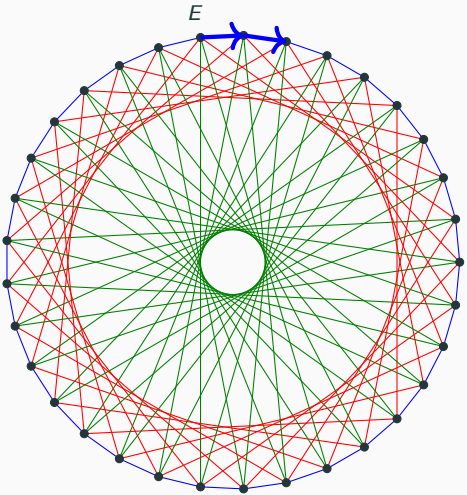
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

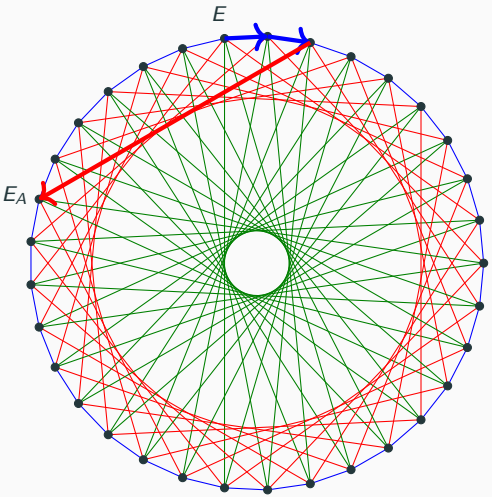
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

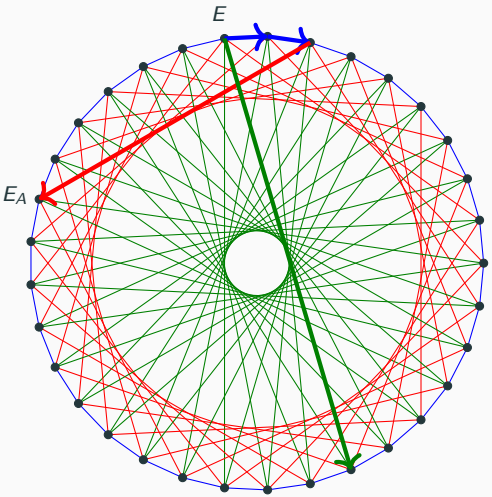
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

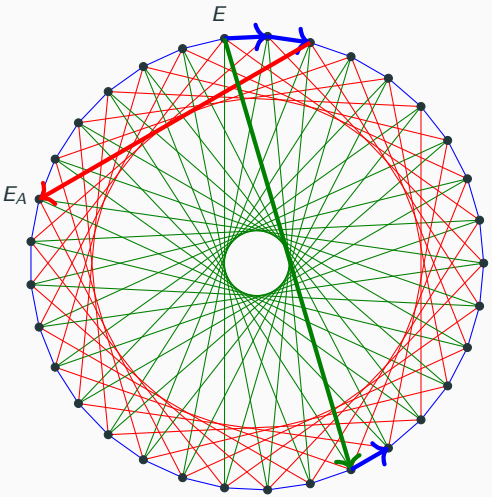
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

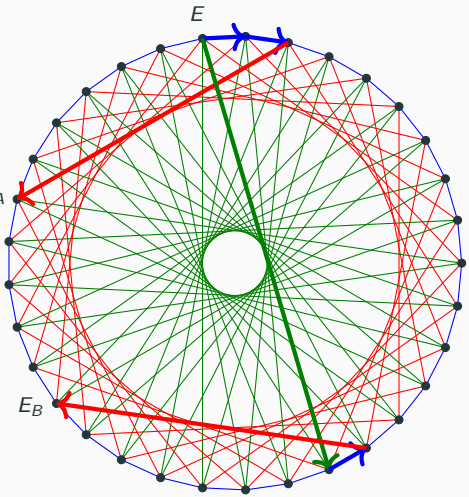
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

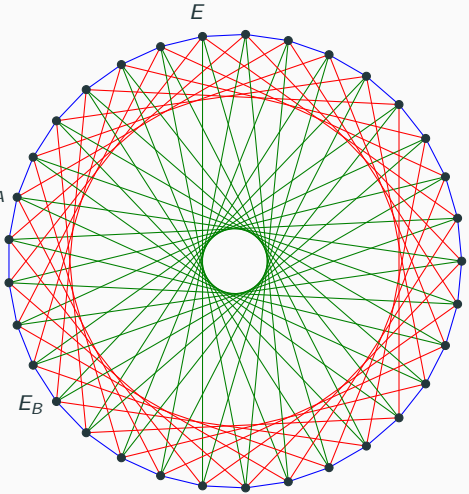
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

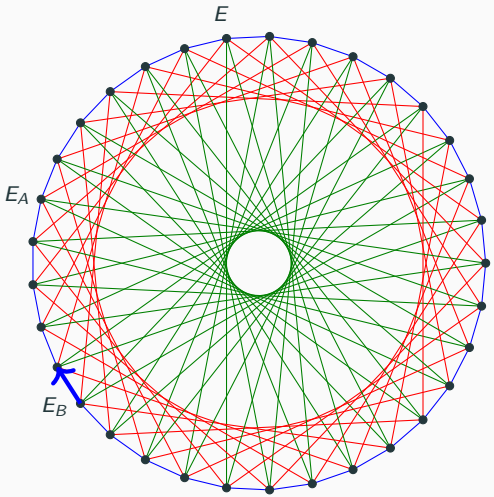
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

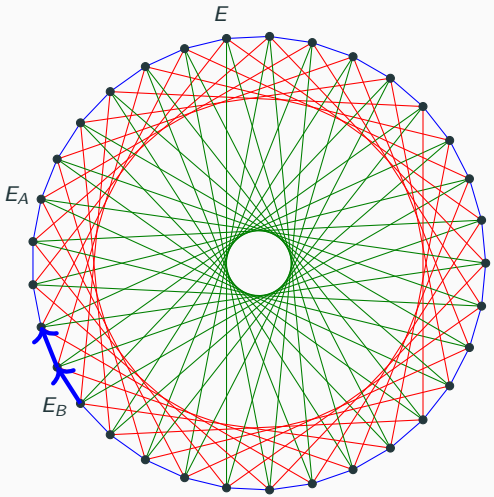
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

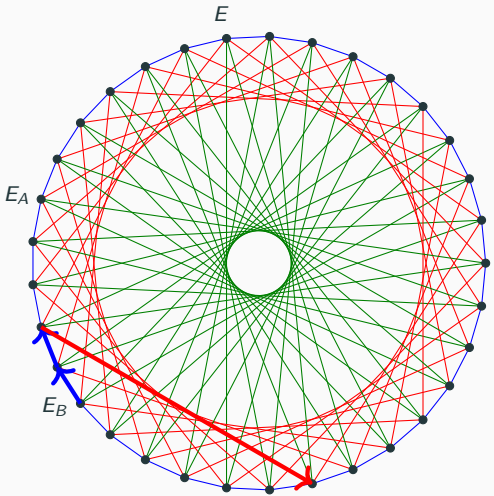
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

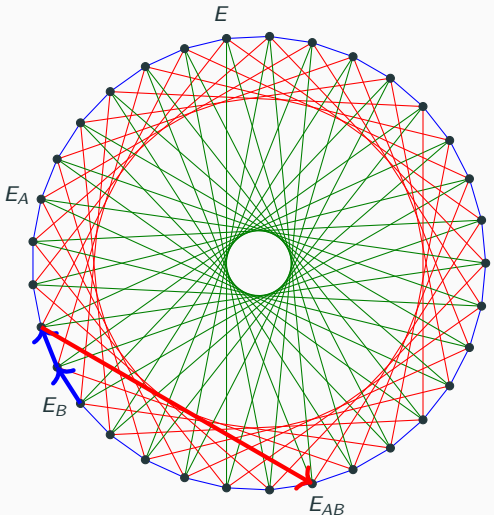
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

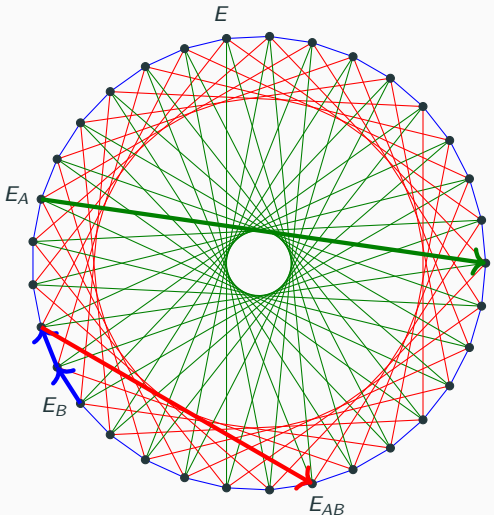
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

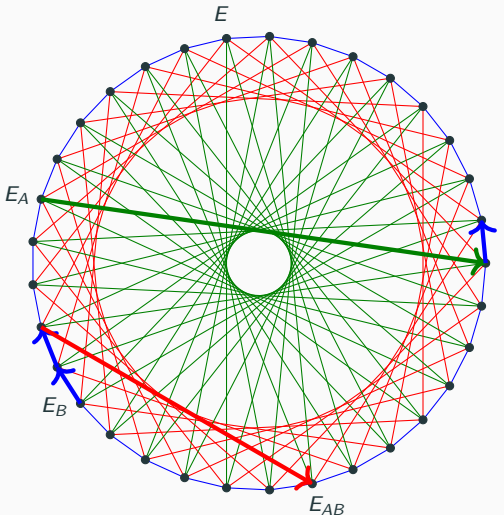
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

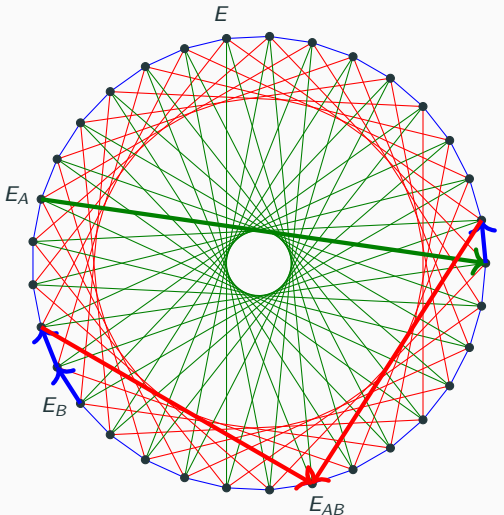
secret: $(+, -, +)$

CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$



Bob

secret: $(+, -, +)$

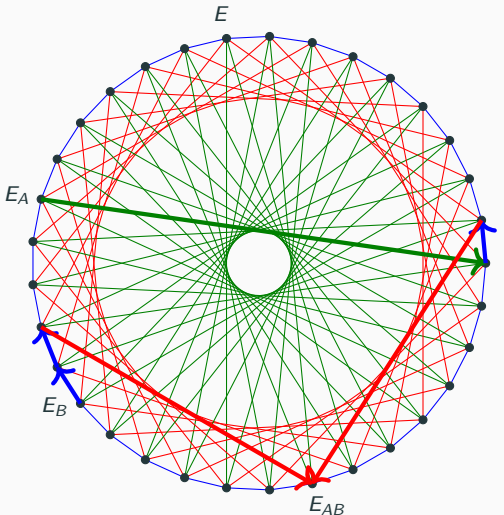
CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$

$\rightsquigarrow (2, 0, -1)$



Bob

secret: $(+, -, +)$

$\rightsquigarrow (-1, 1, 1)$

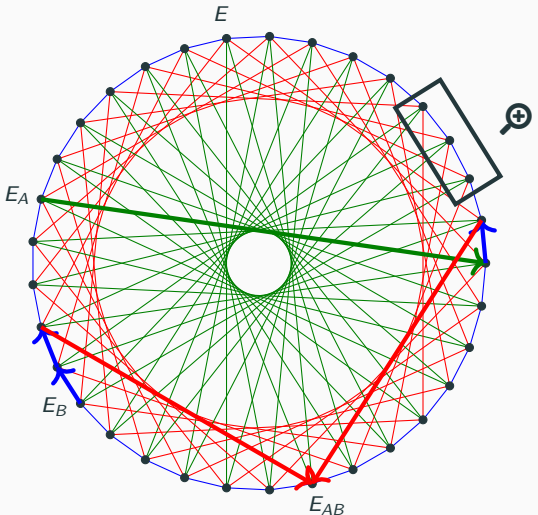
CSIDH

Toy example: $p = 659 = 4 \cdot 3 \cdot 5 \cdot 11 - 1$

Alice

secret: $(+, +, -)$

$\rightsquigarrow (2, 0, -1)$



Bob

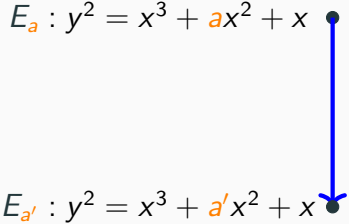
secret: $(+, -, +)$

$\rightsquigarrow (-1, 1, 1)$

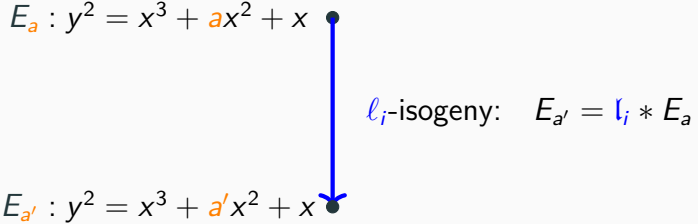
CSIDH

$$E_a : y^2 = x^3 + ax^2 + x \bullet$$

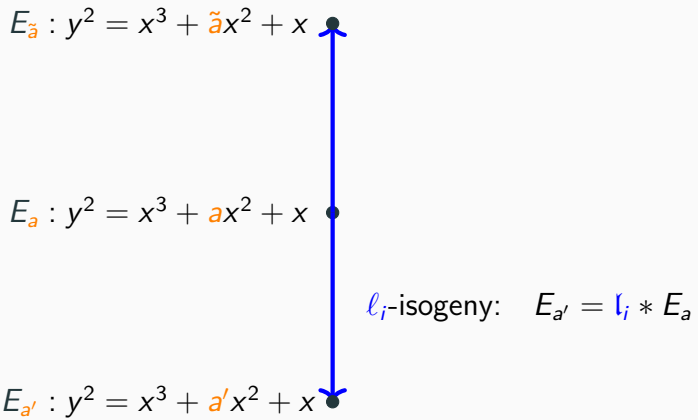
CSIDH



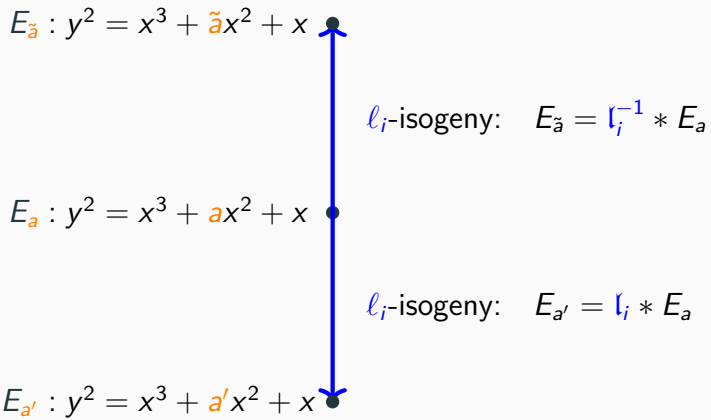
CSIDH



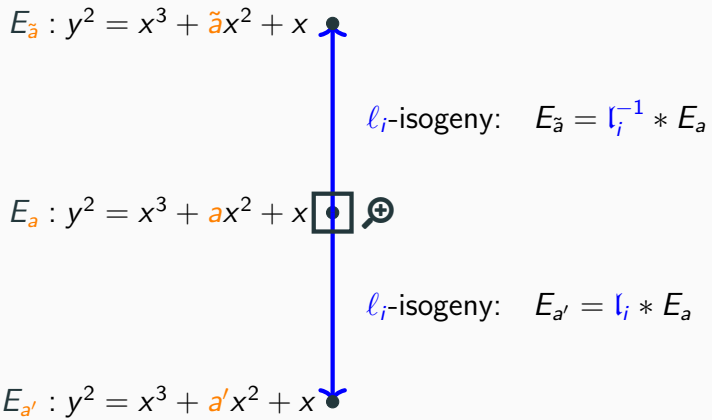
CSIDH


$E_{\tilde{a}} : y^2 = x^3 + \tilde{a}x^2 + x$
$E_a : y^2 = x^3 + ax^2 + x$
ℓ_i-isogeny: $E_{a'} = \mathfrak{l}_i * E_a$
$E_{a'} : y^2 = x^3 + a'x^2 + x$

CSIDH


$E_{\tilde{a}} : y^2 = x^3 + \tilde{a}x^2 + x$
ℓ_i-isogeny: $E_{\tilde{a}} = \mathfrak{l}_i^{-1} * E_a$
$E_a : y^2 = x^3 + ax^2 + x$
ℓ_i-isogeny: $E_{a'} = \mathfrak{l}_i * E_a$
$E_{a'} : y^2 = x^3 + a'x^2 + x$

CSIDH


$E_{\tilde{a}} : y^2 = x^3 + \tilde{a}x^2 + x$
ℓ_i-isogeny: $E_{\tilde{a}} = \mathfrak{l}_i^{-1} * E_a$
$E_a : y^2 = x^3 + ax^2 + x$
ℓ_i-isogeny: $E_{a'} = \mathfrak{l}_i * E_a$
$E_{a'} : y^2 = x^3 + a'x^2 + x$

Usual representations of curves E_a in projective coefficients:

- Montgomery form $(A : C)$ with $a = A/C$ and C non-zero
- alternative Montgomery form $(A + 2C : 4C)$ with $a = A/C$ and C non-zero

Usual representations of curves E_a in projective coefficients:

- Montgomery form $(A : C)$ with $a = A/C$ and C non-zero
- alternative Montgomery form $(A + 2C : 4C)$ with $a = A/C$ and C non-zero

Our attack aims at two types of vulnerable representations:

Usual representations of curves E_a in projective coefficients:

- Montgomery form $(A : C)$ with $a = A/C$ and C non-zero
- alternative Montgomery form $(A + 2C : 4C)$ with $a = A/C$ and C non-zero

Our attack aims at two types of vulnerable representations:

- Zero-value representation: Represents the Montgomery coefficient $a \in \mathbb{F}_p$ in projective coordinates $(\alpha : \beta)$ such that $\alpha = 0$ or $\beta = 0$.

CSIDH

Usual representations of curves E_a in projective coefficients:

- Montgomery form $(A : C)$ with $a = A/C$ and C non-zero
- alternative Montgomery form $(A + 2C : 4C)$ with $a = A/C$ and C non-zero

Our attack aims at two types of vulnerable representations:

- Zero-value representation: Represents the Montgomery coefficient $a \in \mathbb{F}_p$ in projective coordinates $(\alpha : \beta)$ such that $\alpha = 0$ or $\beta = 0$.
- Strongly-correlated representation: Represents the Montgomery coefficient $a \in \mathbb{F}_p$ in projective coordinates $(\alpha : \beta)$ such that the bit representations of α and β are bit shifts.

Vulnerable Curves

E_0 is a valid supersingular curve in the usual CSIDH setting.

Vulnerable Curves

E_0 is a valid supersingular curve in the usual CSIDH setting.

- E_0 in Montgomery form: $(0 : C)$ with $C \in \mathbb{F}_p \backslash \{0\}$

Vulnerable Curves

E_0 is a valid supersingular curve in the usual CSIDH setting.

- E_0 in Montgomery form: $(0 : C)$ with $C \in \mathbb{F}_p \backslash \{0\}$

  $\rightsquigarrow$ zero-value representation

Vulnerable Curves

E_0 is a valid supersingular curve in the usual CSIDH setting.

- E_0 in Montgomery form: $(0 : C)$ with $C \in \mathbb{F}_p \backslash \{0\}$

  $\rightsquigarrow$ zero-value representation
- E_0 in alternative Montgomery form: $(2C : 4C)$ with $C \in \mathbb{F}_p \backslash \{0\}$

Vulnerable Curves

E_0 is a valid supersingular curve in the usual CSIDH setting.

- E_0 in Montgomery form: $(0 : C)$ with $C \in \mathbb{F}_p \backslash \{0\}$

  $\rightsquigarrow$ zero-value representation
- E_0 in alternative Montgomery form: $(2C : 4C)$ with $C \in \mathbb{F}_p \backslash \{0\}$

  $\rightsquigarrow$ strongly-correlated representation if $2C < p/2$

Vulnerable Curves

E_0 is a valid supersingular curve in the usual CSIDH setting.

- E_0 in Montgomery form: $(0 : C)$ with $C \in \mathbb{F}_p \backslash \{0\}$

  $\rightsquigarrow$ zero-value representation
- E_0 in alternative Montgomery form: $(2C : 4C)$ with $C \in \mathbb{F}_p \backslash \{0\}$

  $\rightsquigarrow$ strongly-correlated representation if $2C < p/2$

$\rightsquigarrow$ Both are detectable via side-channel analysis!

Vulnerable Curves

E_0 is a valid supersingular curve in the usual CSIDH setting.

- E_0 in Montgomery form: $(0 : C)$ with $C \in \mathbb{F}_p \backslash \{0\}$

  $\rightsquigarrow$ zero-value representation
- E_0 in alternative Montgomery form: $(2C : 4C)$ with $C \in \mathbb{F}_p \backslash \{0\}$

  $\rightsquigarrow$ strongly-correlated representation if $2C < p/2$

$\rightsquigarrow$ Both are detectable via side-channel analysis!

Also works for E_6 in alternative Montgomery form: $(8C : 4C)$ with $C \in \mathbb{F}_p$ is strongly-correlated if $4C < p/2$.

Attack idea

Idea: Guess a secret key bit, and let the target's isogeny path pass over E_0 or E_6 if the guess was correct.

$\rightsquigarrow$ Correct guess can be confirmed by side-channel analysis.

Attack idea

Idea: Guess a secret key bit, and let the target's isogeny path pass over E_0 or E_6 if the guess was correct.

$\rightsquigarrow$ Correct guess can be confirmed by side-channel analysis.

- Constant-time CSIDH usually has an ordered evaluation of isogenies: $\mathfrak{l}^{(n)} * \cdots * \mathfrak{l}^{(1)} * E$ (modulo point rejections).

Attack idea

Idea: Guess a secret key bit, and let the target's isogeny path pass over E_0 or E_6 if the guess was correct.

$\rightsquigarrow$ Correct guess can be confirmed by side-channel analysis.

- Constant-time CSIDH usually has an ordered evaluation of isogenies: $\mathfrak{l}^{(n)} * \cdots * \mathfrak{l}^{(1)} * E$ (modulo point rejections).
- Task: Find out if $\mathfrak{l}^{(k)} = \mathfrak{l}_i^e$ for $e = 1$ or $e = -1$ (or $e = 0$ if the implementation uses dummy isogenies).

Generic attack for the k-th bit

- Assume we know the first $k-1$ isogeny steps $\alpha_{k-1} = \mathfrak{l}^{(k-1)} * \cdots * \mathfrak{l}^{(1)}$.

Generic attack for the k-th bit

- Assume we know the first $k-1$ isogeny steps $\mathfrak{a}_{k-1} = \mathfrak{l}^{(k-1)} * \cdots * \mathfrak{l}^{(1)}$.
- Guess e and set $\mathfrak{a}_{k,e} = \mathfrak{l}_i^e * \mathfrak{a}_{k-1}$.

Generic attack for the k-th bit

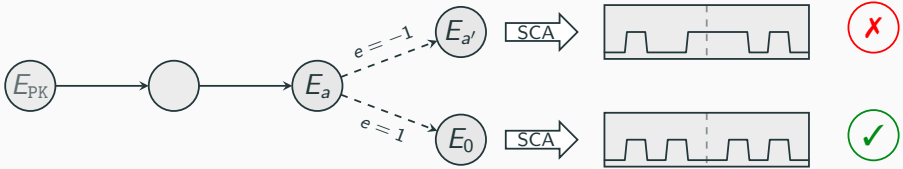
- Assume we know the first $k-1$ isogeny steps $\mathfrak{a}_{k-1} = \mathfrak{l}^{(k-1)} * \cdots * \mathfrak{l}^{(1)}$.
- Guess e and set $\mathfrak{a}_{k,e} = \mathfrak{l}_i^e * \mathfrak{a}_{k-1}$.
- Pass $E_{\mathrm{PK}} = \mathfrak{a}_{k,e}^{-1} * E_0$ as public key.

Generic attack for the k-th bit

- Assume we know the first $k-1$ isogeny steps $\mathfrak{a}_{k-1} = \mathfrak{l}^{(k-1)} * \cdots * \mathfrak{l}^{(1)}$.
- Guess e and set $\mathfrak{a}_{k,e} = \mathfrak{l}_i^e * \mathfrak{a}_{k-1}$.
- Pass $E_{\mathrm{PK}} = \mathfrak{a}_{k,e}^{-1} * E_0$ as public key.
- If the k-th step passes over E_0, the guess was correct;
  otherwise, guess a different e and repeat.

Generic attack for the k-th bit

- Assume we know the first $k-1$ isogeny steps $\mathfrak{a}_{k-1} = \mathfrak{l}^{(k-1)} * \cdots * \mathfrak{l}^{(1)}$.
- Guess e and set $\mathfrak{a}_{k,e} = \mathfrak{l}_i^e * \mathfrak{a}_{k-1}$.
- Pass $E_{\mathrm{PK}} = \mathfrak{a}_{k,e}^{-1} * E_0$ as public key.
- If the k-th step passes over E_0, the guess was correct;
  otherwise, guess a different e and repeat.



SQALE [Chávez-Saab–Chi-Domínguez–Jaques–Rodríguez-Henríquez-2020]

- SQALE uses the alternative Montgomery form $(A + 2C : 4C)$.
  $\rightsquigarrow$ we can detect E_0.

SQALE [Chávez-Saab–Chi-Domínguez–Jaques–Rodríguez-Henríquez-2020]

- SQALE uses the alternative Montgomery form $(A + 2C : 4C)$.
  $\rightsquigarrow$ we can detect E_0.
- SQALE uses large parameters (2048-bit to 9216-bit primes) and secret keys from $\{-1, 1\}^n$.

SQALE [Chávez-Saab–Chi-Domínguez–Jaques–Rodríguez-Henríquez-2020]

- SQALE uses the alternative Montgomery form $(A + 2C : 4C)$.
  $\rightsquigarrow$ we can detect E_0.
- SQALE uses large parameters (2048-bit to 9216-bit primes) and secret keys from $\{-1, 1\}^n$.
- Ordered evaluation: $\mathfrak{l}_n * \cdots * \mathfrak{l}_1 * E$

SQALE [Chávez-Saab–Chi-Domínguez–Jaques–Rodríguez-Henríquez-2020]

- SQALE uses the alternative Montgomery form $(A + 2C : 4C)$.
  $\rightsquigarrow$ we can detect E_0.
- SQALE uses large parameters (2048-bit to 9216-bit primes) and secret keys from $\{-1, 1\}^n$.
- Ordered evaluation: $\mathfrak{l}_n * \cdots * \mathfrak{l}_1 * E$
- Adaptively recover key bits e_i with the generic approach.

SQALE [Chávez-Saab–Chi-Domínguez–Jaques–Rodríguez-Henríquez-2020]

- SQALE uses the alternative Montgomery form $(A + 2C : 4C)$.
  $\rightsquigarrow$ we can detect E_0.
- SQALE uses large parameters (2048-bit to 9216-bit primes) and secret keys from $\{-1, 1\}^n$.
- Ordered evaluation: $\mathfrak{l}_n * \cdots * \mathfrak{l}_1 * E$
- Adaptively recover key bits e_i with the generic approach.
- Each step can fail with a probability of $1/\ell_i$
  $\rightsquigarrow$ increases the number of measurements.

CTIDH [Banegas–Bernstein–Campos–Chou–Lange–Meyer–Smith–Sotáková-2021]

- CTIDH switches between Montgomery and alternative Montgomery form.
  $\rightsquigarrow$ we can detect E_0.

CTIDH [Banegas–Bernstein–Campos–Chou–Lange–Meyer–Smith–Sotáková-2021]

- CTIDH switches between Montgomery and alternative Montgomery form.
  $\rightsquigarrow$ we can detect E_0.
- CTIDH uses a more complicated key space and hides the actual isogeny degrees in use.

CTIDH [Banegas–Bernstein–Campos–Chou–Lange–Meyer–Smith–Sotáková-2021]

- CTIDH switches between Montgomery and alternative Montgomery form.
  $\rightsquigarrow$ we can detect E_0.
- CTIDH uses a more complicated key space and hides the actual isogeny degrees in use.
- CTIDH uses an ordered evaluation, but we have to guess the direction and the degree of each isogeny.

CTIDH [Banegas–Bernstein–Campos–Chou–Lange–Meyer–Smith–Sotáková-2021]

- CTIDH switches between Montgomery and alternative Montgomery form.
  $\rightsquigarrow$ we can detect E_0.
- CTIDH uses a more complicated key space and hides the actual isogeny degrees in use.
- CTIDH uses an ordered evaluation, but we have to guess the direction and the degree of each isogeny.
- Each step can fail with a probability of $\approx 1/\ell_i$

CTIDH [Banegas–Bernstein–Campos–Chou–Lange–Meyer–Smith–Sotáková-2021]

- CTIDH switches between Montgomery and alternative Montgomery form.
  $\rightsquigarrow$ we can detect E_0.
- CTIDH uses a more complicated key space and hides the actual isogeny degrees in use.
- CTIDH uses an ordered evaluation, but we have to guess the direction and the degree of each isogeny.
- Each step can fail with a probability of $\approx 1/\ell_i$
- This increases the number of measurements.

Simulation

Simulation of our attacks exploiting strong correlation:

- We require ordered evaluations
  $\rightsquigarrow$ exact positions of computations involving A and C resp. $A + 2C$ and $4C$ are known!

Simulation

Simulation of our attacks exploiting strong correlation:

- We require ordered evaluations
  $\rightsquigarrow$ exact positions of computations involving A and C resp. $A + 2C$ and $4C$ are known!
- Simulation gets Hamming weights of all limbs and adds noise.

Simulation

Simulation of our attacks exploiting strong correlation:

- We require ordered evaluations
  $\rightsquigarrow$ exact positions of computations involving A and C resp. $A + 2C$ and $4C$ are known!
- Simulation gets Hamming weights of all limbs and adds noise.
- Checks for strong correlation.

Simulation

Simulation of our attacks exploiting strong correlation:

- We require ordered evaluations
  $\rightsquigarrow$ exact positions of computations involving A and C resp. $A + 2C$ and $4C$ are known!
- Simulation gets Hamming weights of all limbs and adds noise.
- Checks for strong correlation.
- average #measurements in SQALE-2048: 8,273

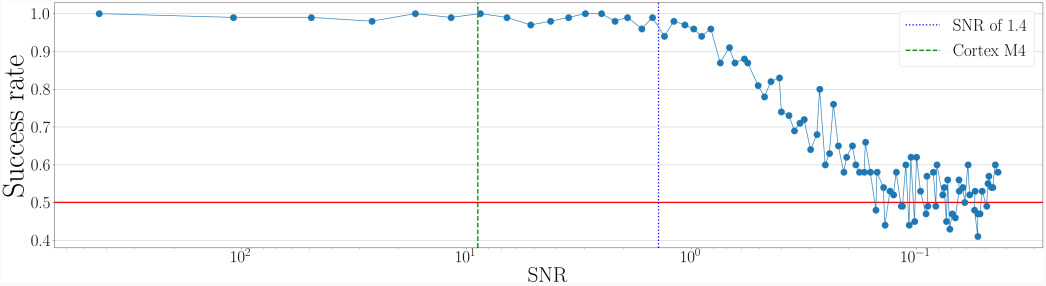
Simulation

Simulation of our attacks exploiting strong correlation:

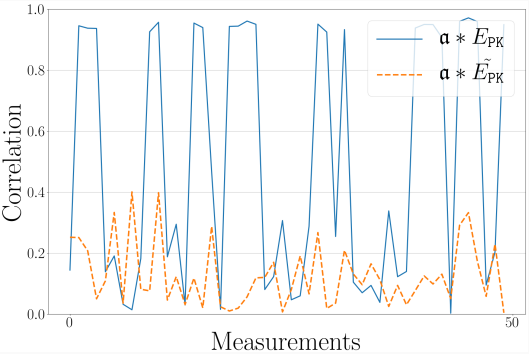
- We require ordered evaluations
  $\rightsquigarrow$ exact positions of computations involving A and C resp. $A + 2C$ and $4C$ are known!
- Simulation gets Hamming weights of all limbs and adds noise.
- Checks for strong correlation.
- average #measurements in SQALE-2048: 8,273
- average #measurements in CTIDH-511: 85,000

Simulation

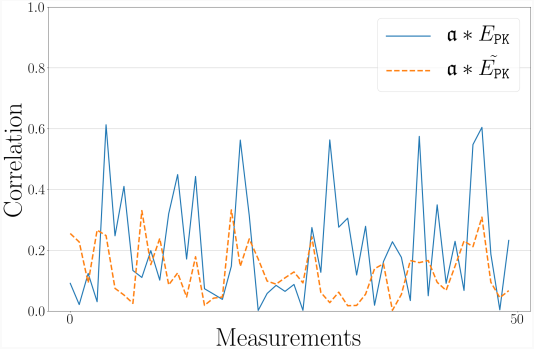
Simulation for different noise levels (signal-to-noise-ratio):



Simulation



(a) Correlation results without noise.



(b) Correlation results with SNR of 1.40.

Countermeasures

- Masking isogeny: Compute $\mathfrak{a} * E$ as $\mathfrak{z}^{-1} * (\mathfrak{a} * (\mathfrak{z} * E))$ with a masking isogeny $\mathfrak{z} * E$ of key space 2^k.
  $\rightsquigarrow$ increases required #samples by factor 2^k

[1] Thanks to the reviewers for this suggestion!

Countermeasures

- Masking isogeny: Compute $\mathfrak{a} * E$ as $\mathfrak{z}^{-1} * (\mathfrak{a} * (\mathfrak{z} * E))$ with a masking isogeny $\mathfrak{z} * E$ of key space 2^k.
  $\rightsquigarrow$ increases required #samples by factor 2^k
- Move to the surface:[1] Pick $p \equiv 7 \mod 8$ and work on the surface of the isogeny graph (see [Castryck-Decru-2020]).
  $\rightsquigarrow$ We are not aware of vulnerable curves in this setting.

[1] Thanks to the reviewers for this suggestion!

SIKE

- The attack applies to SIKE too: E_0 and E_6 are valid curves in SIKE

SIKE

- The attack applies to SIKE too: E_0 and E_6 are valid curves in SIKE
- Attack guesses secret key bits/trits and detects which guess leads to a path over E_0 or E_6.

SIKE

- The attack applies to SIKE too: E_0 and E_6 are valid curves in SIKE
- Attack guesses secret key bits/trits and detects which guess leads to a path over E_0 or E_6.
- Constructs public keys with the framework of [Adj–Chi-Domínguez–Mateu–Rodríguez-Henríquez-2022].

SIKE

- The attack applies to SIKE too: E_0 and E_6 are valid curves in SIKE
- Attack guesses secret key bits/trits and detects which guess leads to a path over E_0 or E_6.
- Constructs public keys with the framework of [Adj–Chi-Domínguez–Mateu–Rodríguez-Henríquez-2022].
- Required number of samples:

Scheme	SIKEp434	SIKEp503	SIKEp610	SIKEp751
Samples	228	265	320	398

Patient Zero & Patient Six: Zero-Value and Correlation Attacks on CSIDH and SIKE†

Thank you!

Paper: https://eprint.iacr.org/2022/904.pdf

Simulation: https://github.com/PaZeZeVaAt/simulation