A New Adaptive Attack on SIDH

Isogeny-Based Cryptography

Tako Boris Fouotsa, LASEC-EPFL

Join work with Christophe Petit, ULB & UoB

An Ordinary Day in Supersingularland, IBM Zürich, 7th July 2022

Introduction

- Isogeny-Based cryptography: very compact keys, ciphertexts and signatures*.
  But is a young field and schemes are relatively slow.
- Non generic cryptanalysis of SIDH:
  - GPST adaptive attack,
  - Petit's torsion point attacks on *imbalance variants* of SIDH.
- Torsion point attacks do not apply to SIDH parameters.

Our contribution:

- **A generalisation of the torsion point attacks**
- **A new adaptive attack on SIDH**

Introduction

- Isogeny-Based cryptography: very compact keys, ciphertexts and signatures*.
  But is a young field and schemes are relatively slow.

- Non generic cryptanalysis of SIDH:
  - GPST adaptive attack,
  - Petit's torsion point attacks on *imbalance variants* of SIDH.
- Torsion point attacks do not apply to SIDH parameters.

Our contribution:

- **A generalisation of the torsion point attacks**
- **A new adaptive attack on SIDH**

Introduction

- Isogeny-Based cryptography: very compact keys, ciphertexts and signatures*.
  But is a young field and schemes are relatively slow.
- Non generic cryptanalysis of SIDH:
  - GPST adaptive attack,
  - Petit's torsion point attacks on *imbalance variants* of SIDH.
- Torsion point attacks do not apply to SIDH parameters.

Our contribution:

- **A generalisation of the torsion point attacks**
- **A new adaptive attack on SIDH**

Outline

Elliptic curves and isogenies

SIDH: Supersingular Isogeny Diffie-Hellman

Torsion point attacks

Generalising the torsion point attacks

A new adaptive attack on SIDH

Summary

Elliptic curves and isogenies

Elliptic curves

- Smooth projective algebraic curve of genus 1. In large characteristic $p > 3$: $E : Y^2 = X^3 + aX + b$.
- Isomorphism classes: same j-invariant $j(E) = 1728\frac{4a^3}{4a^3+27b^2}$.
- E has an abelian group structure, and the n-torsion group for n $(p \nmid n)$
$$E[n] \simeq \mathbb{Z}/n\mathbb{Z} \oplus \mathbb{Z}/n\mathbb{Z}$$
- Over a finite field:
$$\begin{array}{ll} \mathrm{End}(E) \simeq \mathcal{O} \subset O_K, K = \mathbb{Q}\sqrt{-\Delta}) & \text{ordinary curve,} \\ \mathrm{End}(E) \simeq \mathcal{O}_{\max} \subset \mathcal{B}_{p,\infty} & \text{supersingular curve.} \end{array}$$

Isogenies

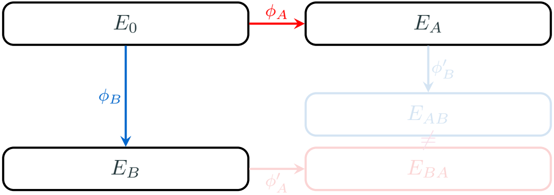
- Rational maps between elliptic curves that are group morphisms.
- They are given by Vélu formulas.
- Their degrees[1] are the size of their kernel.
- Efficiently computable when the degree is smooth, difficult to compute when the degree is not smooth.
- Pure isogeny problem: *given two isogenous elliptic curves E_1 and E_2, compute an isogeny $\phi : E_1 \to E_2$.*

[1] Separable isogenies.

SIDH: Supersingular Isogeny Diffie-Hellman

$$
\begin{tikzcd}
E_0 \arrow[r, "\phi_A"] \arrow[dd, "\phi_B"'] & E_A \arrow[d, "\phi'_B"] \\
& E_{AB} \arrow[d, phantom, "\neq"] \\
E_B \arrow[r, "\phi'_A"'] & E_{BA}
\end{tikzcd}
$$

How would you define ϕ'_A and ϕ'_B? Will the resulting diagram commute?

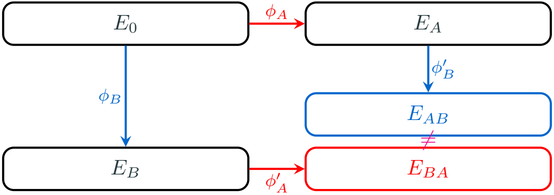


How would you define ϕ'_A and ϕ'_B? Will the resulting diagram commute?

SIDH

$$
\begin{tikzcd}
E_0 \arrow[r, "\phi_A"] \arrow[dd, "\phi_B"'] & E_A \arrow[d, "\phi'_B"] \\
 & E_{AB} \arrow[d, phantom, "\neq"] \\
E_B \arrow[r, "\phi'_A"'] & E_{BA}
\end{tikzcd}
$$

How would you define ϕ'_A and ϕ'_B? Will the resulting diagram commute?



How would you define ϕ'_A and ϕ'_B? Will the resulting diagram commute?

SIDH

$$p = N_A N_B - 1, \quad E_0[N_A] = \langle P_A, Q_A \rangle, \quad E_0[N_B] = \langle P_B, Q_B \rangle$$

$$\begin{tikzcd}
\boxed{E_0, P_A, Q_A, P_B, Q_B} \arrow[r, "\phi_A"] \arrow[d, "\phi_B"'] & \boxed{E_A, \phi_A(P_B), \phi_A(Q_B)} \arrow[d, "\phi'_B"] \\
\boxed{E_B, \phi_B(P_A), \phi_B(Q_A)} \arrow[r, "\phi'_A"'] & \boxed{E_{AB} = E_{BA}}
\end{tikzcd}$$

$\ker \phi_A = \langle P_A + [\alpha]Q_A \rangle, \quad \ker \phi_B = \langle P_B + [\beta]Q_B \rangle$
$\ker \phi'_A = \langle \phi_B(P_A) + [\alpha]\phi_B(Q_A) \rangle,$
$\ker \phi'_B = \langle \phi_A(P_B) + [\beta]\phi_A(Q_B) \rangle$

Validation method: $\quad e_{2^a}(\phi_B(P_A), \phi_B(Q_A)) = e_{2^a}(P_A, Q_A)^{3^b}.$

SSI-T Problem: Given E_0, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, compute ϕ_A.

SIDH

$$p = N_A N_B - 1, \quad E_0[N_A] = \langle P_A, Q_A \rangle, \quad E_0[N_B] = \langle P_B, Q_B \rangle$$

$$\begin{tikzcd}
\boxed{E_0, P_A, Q_A, P_B, Q_B} \arrow[r, "\phi_A"] \arrow[d, "\phi_B"'] & \boxed{E_A, \phi_A(P_B), \phi_A(Q_B)} \arrow[d, "\phi'_B"] \\
\boxed{E_B, \phi_B(P_A), \phi_B(Q_A)} \arrow[r, "\phi'_A"'] & \boxed{E_{AB} = E_{BA}}
\end{tikzcd}$$

$\ker \phi_A = \langle P_A + [\alpha]Q_A \rangle, \quad \ker \phi_B = \langle P_B + [\beta]Q_B \rangle$

$\ker \phi'_A = \langle \phi_B(P_A) + [\alpha]\phi_B(Q_A) \rangle,$

$\ker \phi'_B = \langle \phi_A(P_B) + [\beta]\phi_A(Q_B) \rangle$

Validation method: $\quad e_{2^a}(\phi_B(P_A), \phi_B(Q_A)) = e_{2^a}(P_A, Q_A)^{3^b}.$

SSI-T Problem: Given E_0, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, compute ϕ_A.

SIDH

$$p = N_A N_B - 1, \quad E_0[N_A] = \langle P_A, Q_A \rangle, \quad E_0[N_B] = \langle P_B, Q_B \rangle$$

$$\begin{tikzcd}
\boxed{E_0, P_A, Q_A, P_B, Q_B} \arrow[r, "\phi_A"] \arrow[d, "\phi_B"'] & \boxed{E_A, \phi_A(P_B), \phi_A(Q_B)} \arrow[d, "\phi'_B"] \\
\boxed{E_B, \phi_B(P_A), \phi_B(Q_A)} \arrow[r, "\phi'_A"'] & \boxed{E_{AB} = E_{BA}}
\end{tikzcd}$$

$\ker \phi_A = \langle P_A + [\alpha] Q_A \rangle, \quad \ker \phi_B = \langle P_B + [\beta] Q_B \rangle$

$\ker \phi'_A = \langle \phi_B(P_A) + [\alpha] \phi_B(Q_A) \rangle,$

$\ker \phi'_B = \langle \phi_A(P_B) + [\beta] \phi_A(Q_B) \rangle$

Validation method: $\quad e_{2^a}(\phi_B(P_A), \phi_B(Q_A)) = e_{2^a}(P_A, Q_A)^{3^b}.$

SSI-T Problem: Given E_0, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, compute ϕ_A.

SIDH

$$p = N_A N_B - 1, \quad E_0[N_A] = \langle P_A, Q_A \rangle, \quad E_0[N_B] = \langle P_B, Q_B \rangle$$

$$\begin{tikzcd}
E_0, P_A, Q_A, P_B, Q_B \arrow[r, "\phi_A"] \arrow[d, "\phi_B"'] & E_A, \phi_A(P_B), \phi_A(Q_B) \arrow[d, "\phi'_B"] \\
E_B, \phi_B(P_A), \phi_B(Q_A) \arrow[r, "\phi'_A"'] & E_{AB} = E_{BA}
\end{tikzcd}$$

$\ker \phi_A = \langle P_A + [\alpha]Q_A \rangle, \quad \ker \phi_B = \langle P_B + [\beta]Q_B \rangle$
$\ker \phi'_A = \langle \phi_B(P_A) + [\alpha]\phi_B(Q_A) \rangle,$
$\ker \phi'_B = \langle \phi_A(P_B) + [\beta]\phi_A(Q_B) \rangle$

Validation method: $\quad e_{2^a}(\phi_B(P_A), \phi_B(Q_A)) = e_{2^a}(P_A, Q_A)^{3^b}.$

SSI-T Problem: Given E_0, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, compute ϕ_A.

SIDH

$$p = N_A N_B - 1, \quad E_0[N_A] = \langle P_A, Q_A \rangle, \quad E_0[N_B] = \langle P_B, Q_B \rangle$$

$$\begin{tikzcd}
E_0, P_A, Q_A, P_B, Q_B \arrow[r, "\phi_A"] \arrow[d, "\phi_B"'] & E_A, \phi_A(P_B), \phi_A(Q_B) \arrow[d, "\phi'_B"] \\
E_B, \phi_B(P_A), \phi_B(Q_A) \arrow[r, "\phi'_A"'] & E_{AB} \; = \; E_{BA}
\end{tikzcd}$$

$\ker \phi_A = \langle P_A + [\alpha] Q_A \rangle, \quad \ker \phi_B = \langle P_B + [\beta] Q_B \rangle$
$\ker \phi'_A = \langle \phi_B(P_A) + [\alpha] \phi_B(Q_A) \rangle,$
$\ker \phi'_B = \langle \phi_A(P_B) + [\beta] \phi_A(Q_B) \rangle$

Validation method: $\quad e_{2^a}(\phi_B(P_A), \phi_B(Q_A)) = e_{2^a}(P_A, Q_A)^{3^b}.$

SSI-T Problem: Given E_0, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, compute ϕ_A.

Torsion point attacks

More facts about isogenies

- For any seperable d-isogeny $\varphi : E \to E'$, there exist a unique* d-isogeny $\hat{\varphi} : E' \to E$ called *the dual* of φ such that $\hat{\varphi} \circ \varphi = [d]_E$ and $\varphi \circ \hat{\varphi} = [d]_{E'}$.

$$\begin{tikzcd} E \arrow[r, shift left, "\varphi"] & E' \arrow[l, shift left, "\hat{\varphi}"] \end{tikzcd}$$

- We have

$$\ker \hat{\varphi} = \varphi(E[d]) \quad \text{and} \quad \ker \varphi = \hat{\varphi}(E'[d]).$$

Take away:

- The knowledge of φ is equivalent to the knowledge of $\hat{\varphi}$.
- You can recover the kernel of a d-isogeny φ by evaluating φ on the d-torsion group.

The torsion point attacks

SSI-T Problem: Given E_0, P_B, Q_B, E_A, $\phi_A(P_B)$, $\phi_A(Q_B)$, compute ϕ_A.

Targets the **SSI-T** assuming that End(E_0) is known.

Is this a fair assumption?

- Case of SIDH, Yes, because $E_0 = E(1728)$ (or its neighbour) is a special curve: End(E_0) is known.
- General case, No. In fact, *computing the endomorphism ring of a random supersingular curve is a hard problem*, which is equivalent to the pure isogeny problem.
  But, we don't know how to generate supersingular curves with unknown endomorphism ring.

So it is definitely a fair assumption.

The torsion point attacks

Endormorphisms of E_0 are carried on to E_A through ϕ_A.

$\phi_A : E_0 \rightarrow E_A$ implies

$$\mathbb{Z} + \phi_A \circ \mathrm{End}(E_0) \circ \hat{\phi}_A \hookrightarrow \mathrm{End}(E_A)$$

$$[d] + \phi_A \circ \theta \circ \hat{\phi}_A = \tau$$

$$\begin{tikzcd} E_0 \arrow[loop left, "\theta"] \arrow[r, shift left, Rightarrow, "\phi_A"] & E_A \arrow[l, shift left, Rightarrow, "\hat{\phi}_A"] \arrow[loop right, "\tau"] \end{tikzcd}$$

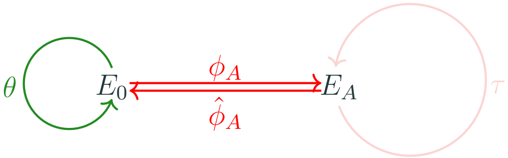
The torsion point attacks

Endormorphisms of E_0 are carried on to E_A through ϕ_A.

$\phi_A : E_0 \to E_A$ implies

$$\mathbb{Z} + \phi_A \circ \mathrm{End}(E_0) \circ \hat{\phi}_A \hookrightarrow \mathrm{End}(E_A)$$

$$[d] + \phi_A \circ \theta \circ \hat{\phi}_A = \tau$$



The torsion point attacks

Endormorphisms of E_0 are carried on to E_A through ϕ_A.

$\phi_A : E_0 \rightarrow E_A$ implies

$$\mathbb{Z} + \phi_A \circ \mathrm{End}(E_0) \circ \hat{\phi}_A \hookrightarrow \mathrm{End}(E_A)$$

$$[d] + \phi_A \circ \theta \circ \hat{\phi}_A = \tau$$

$$\begin{tikzcd}
E_0 \arrow[loop left, "\theta"] \arrow[r, shift left, Rightarrow, "\phi_A"] & E_A \arrow[l, shift left, Rightarrow, "\hat{\phi}_A"] \arrow[loop right, "\tau"]
\end{tikzcd}$$

The torsion point attacks

When $\tau = [d] + \phi_A \circ \theta \circ \hat{\phi}_A$ has degree $N_B^2 e$ where e is small, we can decompose τ as

$$\tau = \hat{\psi}_2 \circ \psi_e \circ \psi_1.$$

$$\begin{tikzcd}
 & & E_1 \arrow[dd, "\psi_e"] \\
E_0 \arrow[loop left, "\theta"] \arrow[r, shift left, "\phi_A"] & E_A \arrow[l, shift left, "\hat{\phi}_A"] \arrow[ur, "\psi_1"] \arrow[dr, "\psi_2"'] & \\
 & & E_2
\end{tikzcd}$$

- ψ_1 and ψ_2 can be computed from $\phi_A(P_B), \phi_A(Q_B)$.
- ψ_e is recovered by brute force.

The torsion point attacks

Once $\tau = [d] + \phi_A \circ \theta \circ \hat{\phi}_A$ is known:

$$\ker \hat{\phi}_A =^{2} \ker(\tau - [d]) \cap E_2[N_A]$$

Break SSI-T $\Rightarrow$ find d, θ such that

$$\deg([d] + \phi_A \circ \theta \circ \hat{\phi}_A) = N_B^2 e.$$

$j(E_0) = 1728 \Rightarrow$ **norm eq.** $: d^2 + N_A^2(c^2 + p(b^2 + a^2)) = N_B^2 e.$

Easy to find solutions when $N_B > pN_A$.

SIDH : $N_A \approx N_B \approx \sqrt{p}$. Still Secure !

[2] under a small condition on θ

Generalising the torsion point attacks

SSI-TG Problem: Given E_0, $G_1, G_2, G_3 \subset E_0[N_B]$ pairwise disjoint cyclic groups of order N_B, E_A, $\phi_A(G_1)$, $\phi_A(G_2)$, $\phi_A(G_3)$, compute ϕ_A.

Lemma: $E_0[N_B] = \langle P_B, Q_B \rangle$. Given $\phi_A(G_1)$, $\phi_A(G_2)$, $\phi_A(G_3)$, there exists an integer λ coprime to N_B such that one can evaluate $\phi_\lambda = [\lambda] \circ \phi_A$ on $E_0[N_B]$.
Moreover, λ^2 can be recovered through a DL comp.:

$$e_{N_B}(\phi_\lambda(P_B), \phi_\lambda(Q_B)) = e_{N_B}(P_B, Q_B)^{\lambda^2 N_A}.$$

N_B not a prime power $\Rightarrow$ λ^2 may have multiple square roots.

Generalising torsion point attacks

SSI-TG Problem: Given E_0, $G_1, G_2, G_3 \subset E_0[N_B]$ pairwise disjoint cyclic groups of order N_B, E_A, $\phi_A(G_1)$, $\phi_A(G_2)$, $\phi_A(G_3)$, compute ϕ_A.

Lemma: $E_0[N_B] = \langle P_B, Q_B \rangle$. Given $\phi_A(G_1)$, $\phi_A(G_2)$, $\phi_A(G_3)$, there exists an integer λ coprime to N_B such that one can evaluate $\phi_\lambda = [\lambda] \circ \phi_A$ on $E_0[N_B]$.
Moreover, λ^2 can be recovered through a DL comp.:

$$e_{N_B}(\phi_\lambda(P_B), \phi_\lambda(Q_B)) = e_{N_B}(P_B, Q_B)^{\lambda^2 N_A}.$$

N_B not a prime power $\Rightarrow$ λ^2 may have multiple square roots.

Generalising torsion point attacks

SSI-TG Problem: Given E_0, $G_1, G_2, G_3 \subset E_0[N_B]$ pairwise disjoint cyclic groups of order N_B, E_A, $\phi_A(G_1)$, $\phi_A(G_2)$, $\phi_A(G_3)$, compute ϕ_A.

Lemma: $E_0[N_B] = \langle P_B, Q_B \rangle$. Given $\phi_A(G_1)$, $\phi_A(G_2)$, $\phi_A(G_3)$, there exists an integer λ coprime to N_B such that one can evaluate $\phi_\lambda = [\lambda] \circ \phi_A$ on $E_0[N_B]$.
Moreover, λ^2 can be recovered through a DL comp.:

$$e_{N_B}(\phi_\lambda(P_B), \phi_\lambda(Q_B)) = e_{N_B}(P_B, Q_B)^{\lambda^2 N_A}.$$

N_B not a prime power $\Rightarrow$ λ^2 may have multiple square roots.

Generalising torsion point attacks

Remark: we don't need to know λ in order to evaluate $\tau = [d] + \phi_A \circ \theta \circ \hat{\phi}_A$ on $E_0[N_B]$, λ^2 suffices.

In fact we have:

$$\phi_\lambda \circ \theta \circ \hat{\phi}_\lambda = ([\lambda] \circ \phi_A) \circ \theta \circ (\widehat{[\lambda] \circ \phi_\lambda}) = [\lambda^2] \circ \phi_A \circ \theta \circ \hat{\phi}_A.$$

Hence

$$\tau = [d] + [\lambda^{-2}] \circ \phi_\lambda \circ \theta \circ \hat{\phi}_\lambda.$$

The rest of the attack is unchanged.

Remark: we don't need to know λ in order to evaluate $\tau = [d] + \phi_A \circ \theta \circ \hat{\phi}_A$ on $E_0[N_B]$, λ^2 suffices.

In fact we have:

$$\phi_\lambda \circ \theta \circ \hat{\phi}_\lambda = ([\lambda] \circ \phi_A) \circ \theta \circ (\widehat{[\lambda] \circ \phi_\lambda}) = [\lambda^2] \circ \phi_A \circ \theta \circ \hat{\phi}_A.$$

Hence

$$\tau = [d] + [\lambda^{-2}] \circ \phi_\lambda \circ \theta \circ \hat{\phi}_\lambda.$$

The rest of the attack is unchanged.

A new adaptive attack on SIDH

key exchange oracle:

$$O(E,R,S,E') = \begin{cases} 1 & \text{if } E/\langle R+[\alpha]S\rangle = E' \\ 0 & \text{if } E/\langle R+[\alpha]S\rangle \neq E' \end{cases}$$

Idea of the attack

1. Actively (using the key exchange oracle) recover the action of ϕ_A on large pairwise disjoint cyclic groups $G_1, G_2, G_3 \subset E_0[NN_B]$ of order NN_B where $p < N$.
2. Use the generalised torsion point attacks to recover ϕ_A.

An overview

key exchange oracle:

$$O(E, R, S, E') = \begin{cases} 1 & \text{if } E/\langle R + [\alpha]S \rangle = E' \\ 0 & \text{if } E/\langle R + [\alpha]S \rangle \neq E' \end{cases}$$

Idea of the attack

1 Actively (using the key exchange oracle) recover the action of ϕ_A on large pairwise disjoint cyclic groups $G_1, G_2, G_3 \subset E_0[NN_B]$ of order NN_B where $p < N$.

2 Use the generalised torsion point attacks to recover ϕ_A.

An overview

key exchange oracle:

$$O(E, R, S, E') = \begin{cases} 1 & \text{if } E/\langle R + [\alpha]S \rangle = E' \\ 0 & \text{if } E/\langle R + [\alpha]S \rangle \neq E' \end{cases}$$

Idea of the attack

1. Actively (using the key exchange oracle) recover the action of ϕ_A on large pairwise disjoint cyclic groups $G_1, G_2, G_3 \subset E_0[NN_B]$ of order NN_B where $p < N$.
2. Use the generalised torsion point attacks to recover ϕ_A. ✓

Step 1: Recovering the action of ϕ_A on a larger group

Set $N = \prod_{i=1}^{e} \ell_i^2$, $\quad \ell_i$ coprime to $N_A N_B$.

Let $\ell \mid N$ and let G be a cyclic group of order $\ell^2 N_B$.

$$
\begin{tikzcd}
E_0 \arrow[r, "\phi_B"] \arrow[d, "\phi_A"'] \arrow[rr, bend left, "\phi_G"] & E_B \arrow[r] \arrow[d, dotted] & E_G \arrow[d, "\phi'_A"] \\
E_A \arrow[r, "\phi'_B"'] \arrow[rr, bend right, dotted, "\phi_H"'] & E_{AB} \arrow[r, dotted] & E_H = E_{\phi_A(G)}
\end{tikzcd}
$$

Query: $O(E_G, R, S, E_H)$, $R = [\ell^{-1}]\phi_G(P_A)$, $S = [\ell^{-1}]\phi_G(Q_A)$

Step 1: Recovering the action of ϕ_A on a larger group

Set $N = \prod_{i=1}^{e} \ell_i^2$, $\quad \ell_i$ coprime to $N_A N_B$.

Let $\ell \mid N$ and let G be a cyclic group of order $\ell^2 N_B$.

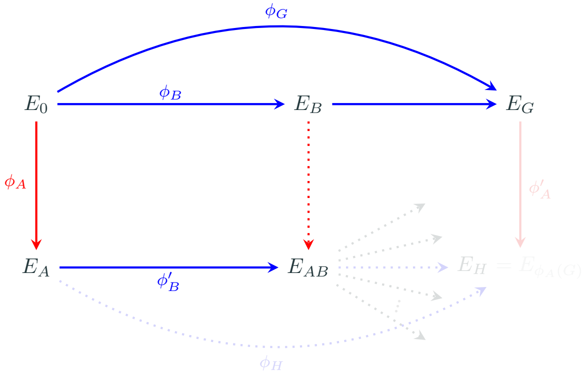
$$
\begin{tikzcd}
E_0 \arrow[r, "\phi_B"] \arrow[d, "\phi_A"'] \arrow[rr, bend left, "\phi_G"] & E_B \arrow[r] \arrow[d, dotted] & E_G \arrow[d, "\phi_A'"] \\
E_A \arrow[r, "\phi_B'"'] \arrow[rr, bend right, dotted, "\phi_H"'] & E_{AB} \arrow[r, dotted] & E_H = E_{\phi_A(G)}
\end{tikzcd}
$$

Query: $O(E_G, R, S, E_H)$, $R = [\ell^{-1}]\phi_G(P_A)$, $S = [\ell^{-1}]\phi_G(Q_A)$

Step 1: Recovering the action of ϕ_A on a larger group

Set $N = \prod_{i=1}^{e} \ell_i^2$, $\quad \ell_i$ coprime to $N_A N_B$.

Let $\ell \mid N$ and let G be a cyclic group of order $\ell^2 N_B$.

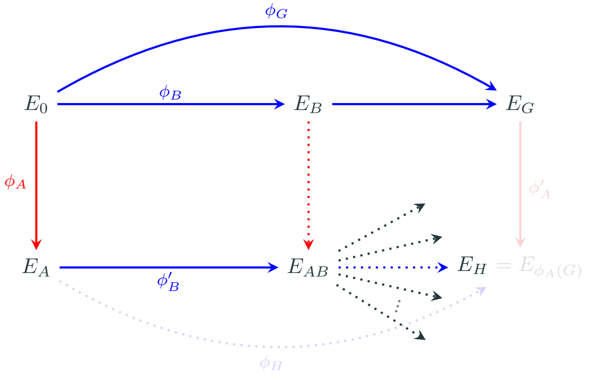


Query: $O(E_G, R, S, E_H)$, $R = [\ell^{-1}]\phi_G(P_A)$, $S = [\ell^{-1}]\phi_G(Q_A)$

Step 1: Recovering the action of ϕ_A on a larger group

Set $N = \prod_{i=1}^{e} \ell_i^2$, $\quad \ell_i$ coprime to $N_A N_B$.

Let $\ell \mid N$ and let G be a cyclic group of order $\ell^2 N_B$.

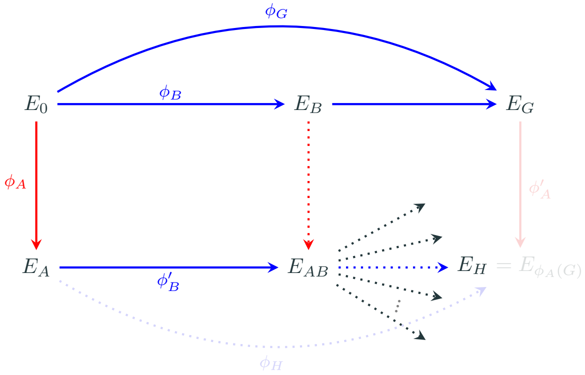


Query: $O(E_G, R, S, E_H)$, $R = [\ell^{-1}]\phi_G(P_A)$, $S = [\ell^{-1}]\phi_G(Q_A)$

Step 1: Recovering the action of ϕ_A on a larger group

Set $N = \prod_{i=1}^{e} \ell_i^2$, $\quad \ell_i$ coprime to $N_A N_B$.

Let $\ell \mid N$ and let G be a cyclic group of order $\ell^2 N_B$.



Query: $O(E_G, R, S, E_H)$, $R = [\ell^{-1}]\phi_G(P_A)$, $S = [\ell^{-1}]\phi_G(Q_A)$

Step 1: Recovering the action of ϕ_A on a larger group

Set $N = \prod_{i=1}^{e} \ell_i^2, \quad \ell_i$ coprime to $N_A N_B$.

Let $\ell \mid N$ and let G be a cyclic group of order $\ell^2 N_B$.

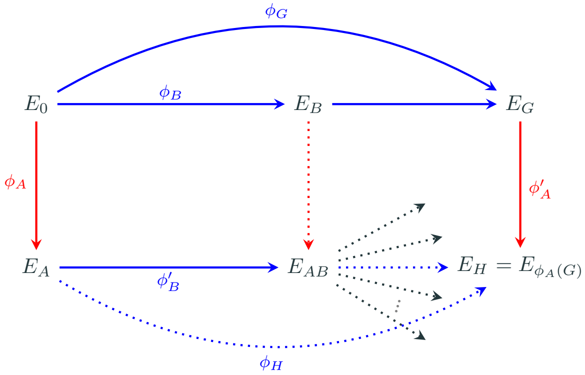
$$\begin{tikzcd}
E_0 \arrow[r, "\phi_B"] \arrow[rr, bend left, "\phi_G"] \arrow[d, "\phi_A"'] & E_B \arrow[r] \arrow[d, dotted] & E_G \arrow[d, "\phi'_A"] \\
E_A \arrow[r, "\phi'_B"'] \arrow[rr, bend right, dotted, "\phi_H"'] & E_{AB} \arrow[r, dotted] & E_H = E_{\phi_A(G)}
\end{tikzcd}$$

Query: $O(E_G, R, S, E_H)$, $R = [\ell^{-1}]\phi_G(P_A)$, $S = [\ell^{-1}]\phi_G(Q_A)$

Step 1: Recovering the action of ϕ_A on a larger group

Set $N = \prod_{i=1}^{e} \ell_i^2$, $\quad \ell_i$ coprime to $N_A N_B$.

Let $\ell \mid N$ and let G be a cyclic group of order $\ell^2 N_B$.



Query: $O(E_G, R, S, E_H)$, $R = [\ell^{-1}]\phi_G(P_A)$, $S = [\ell^{-1}]\phi_G(Q_A)$

Countermeasures

- Start from a supersingular curve E_0 with unknown endomorphism ring, this would counter the torsion point attacks that are used as building block in the attack.
- Use FO-transform as in SIKE: when running the re-encryption step in the FO, Alice will notice that the public key used was malicious.

Summary

Summary

We have presented:

- A generalisation of the torsion point attacks
- A new adaptive attack on SIDH
- Some countermeasures

Take away:

- Torsion point attacks become relevant to SIDH parameters in an adaptive setting!
- New cryptanalytic tool !

Golden open questions:

- How far can we push torsion point attacks?
- And CSIDH? Any hope for an adaptive attack?

Summary

We have presented:

- A generalisation of the torsion point attacks
- A new adaptive attack on SIDH
- Some countermeasures

Take away:

- Torsion point attacks become relevant to SIDH parameters in an adaptive setting!
- New cryptanalytic tool !

Golden open questions:

- How far can we push torsion point attacks?
- And CSIDH? Any hope for an adaptive attack?

Summary

We have presented:

- A generalisation of the torsion point attacks
- A new adaptive attack on SIDH
- Some countermeasures

Take away:

- Torsion point attacks become relevant to SIDH parameters in an adaptive setting!
- New cryptanalytic tool !

Golden open questions:

- How far can we push torsion point attacks?
- And CSIDH? Any hope for an adaptive attack?

Happy to discuss your comments and questions !!!

Full paper available at:

https://eprint.iacr.org/2021/1322