Beyond SIDH: A survey on countermeasures and new constructions

École Polytechnique Fédérale de Lausanne, Switzerland

June 12th 2024

Outline

Overview – Public-key cryptography

- Public-key cryptography from a far:
  - Relies on **computationally hard problems** like integer factorization and discrete logarithm.
  - Basis for secure online payments and private messaging.
- Quantum threat and post-quantum cryptography:
  - Quantum computers solve those **computationally hard problems** in polynomial time

    $\Longrightarrow$ Post-quantum cryptography (to resist those attacks)

Overview – Isogeny-based cryptography

- Isogeny-based cryptography:
    - Uses maps between *supersingualr* elliptic curves
    - The *(pure) isogeny problem* is quantum-hard

- SIDH / SIKE:
    - 2011: Supersingular Isogeny Diffie-Hellman (SIDH)
    - 2016: Supersingular Isogeny Key Exchange (SIKE)
    - 2022:
        - May: SIKE advances to 4^{th} round of NIST's competition
        - August: SIDH is broken

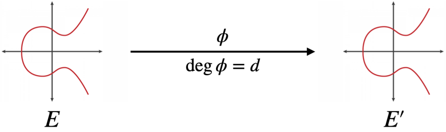
- After the attacks:
    - Countermeasures: M-SIDH, MD-SIDH
    - New constructions: FESTA, POKE

Part 1

Supersingular Isogeny Diffie-Hellman (SIDH)

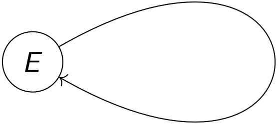
Isogenies between elliptic curves

- An isogeny $\phi : E \to E' := E/K$
  is a *surjective* ***homomorphism*** having *finite* ***kernel***
- The kernel K uniquely determines the isogeny
- $\#\ker\phi$ is the number of pre-images that each point has
- For this presentation (separable isogenies):
  $\deg\phi := \#\ker\phi$
- A d-isogeny is an isogeny whose degree is d



Isogenies – More facts

- An **endomorphism** $\phi : E \to E$ has degree > 1
  $(\mathrm{End}(E), +, \circ)$ is the endomorphism ring
  Example: $[m] : P \mapsto [m]P$ has degree m^2
- Torsion points: $E[m] := \{P \in E(\bar{k}) : [m]P = \mathcal{O}\}$



SIDH – The scheme

Public parameters: ℓ_A,ℓ_B, p primes. $E/\mathbb{F}_{p^2}$ supersingular. $\langle P_A, Q_A\rangle = E[\ell_A^a]$, $\langle P_B, Q_B\rangle = E[\ell_B^b]$. Typically $\ell_A = 2$ and $\ell_B = 3$ for efficiency reasons.
Alice: $A = [m_A]P_A + [n_A]Q_A$ with $m_A, n_A \in_\$ \mathbb{Z}/\ell_A^a\mathbb{Z}$.
Send to Bob: image of the torsion basis $\{\phi_A(P_B), \phi_A(Q_B)\}$ and $E_A = E/\langle A\rangle$.
Bob: $B = [m_B]P_B + [n_B]Q_B$ with $m_B, n_B \in_\$ \mathbb{Z}/\ell_B^b\mathbb{Z}$.
Send to Alice: image of the torsion basis $\{\phi_B(P_A), \phi_B(Q_A)\}$ and $E_B = E/\langle B\rangle$.

$\ker\phi_A = \langle A\rangle \subset E[\ell_A^a]$

$\ker\phi_B = \langle B\rangle \subset E[\ell_B^b]$

$$\begin{tikzcd}
E \arrow[r, "\phi_A"] \arrow[d, "\phi_B"'] & E_A \arrow[d, "\phi'_B"] \arrow[dl, dashed, shift left] \\
E_B \arrow[r, "\phi'_A"'] \arrow[ur, dashed, shift left] & E_{AB} = E_{BA}
\end{tikzcd}$$

$\ker\phi'_A = \langle\phi_B(A)\rangle$

$\ker\phi'_B = \langle\phi_A(B)\rangle$

Figure 1: Orange $\leftrightarrow$ Alice, blue $\leftrightarrow$ Bob. Alice and Bob ***close** the commutative diagram*. Taken from [De 17, Figure 16].

SIDH – Hardness assumptions

- Many isogeny-based protocols $\to$ *(pure) isogeny problem*: Recover large-degree isogeny $\phi : E \to E'$ between two elliptic curves
- SIDH $\to$ *Supersingular Isogeny with Torsion (SSI-T) problem*: **Weaker** because parties publish the *image of a torsion basis* under the secret isogeny

Breaking SIDH in polynomial time

Historical stages of SIDH:

- 2011: SIDH is created
- 2014: SIDH is improved
- 2016: NIST post-quantum standardization competition
- 2016: SIKE is obtained from SIDH
- 2017: [Pet17] polynomial time with unbalanced parameters ($d_A \geq d_B^2$)
- 2022:
    - May: SIKE advances to 4^{th} round
    - August: SIDH is completely broken by the "SIDH attacks":
        - [CD23]: heuristic polynomial time, knowing $\mathrm{End}(E)$
        - [Mai+23]: provable polynomial time, knowing $\mathrm{End}(E)$
        - [Rob23]: provable polynomial time, **without** knowledge of $\mathrm{End}(E)$
    - September: SIKE is declared insecure
- 2023–today: Countermeasures and new constructions

Castryck, Decru – “An efficient key recovery attack on SIDH”

- Goal: recover Bob's 3^b-isogeny $\phi_B : E_0 \to E_B$
- How? Exploit the knowledge of two elements:
  - The *torsion point information* $\phi_B(P_A), \phi_B(Q_A)$ published by Bob
  - The *degree* $d_B = 3^b$ of the secret isogeny ϕ_B
- Idea: embed a *part of* ϕ_B into a ***higher-dimensional*** **isogeny**

Castryck, Decru – "An efficient key recovery attack on SIDH"

Strategy of the attack:

$$\phi_B : \underbrace{E_0 \xrightarrow{\phi_B} E_B}_{\deg \phi_B = 3^b}$$

Castryck, Decru – "An efficient key recovery attack on SIDH"

Strategy of the attack: **iterate** for a lot of steps!
1st step:

$$\phi_B : \underbrace{E_0 \xrightarrow{\kappa_1} E_1}_{\substack{\kappa_1 \,:\, E_0 \twoheadrightarrow E_1 \\ \deg \kappa_1 \,=\, 3^{\beta_1}}} \underbrace{\xrightarrow{\qquad \phi_1 \qquad} E_B}_{\substack{\phi_1 \,:\, E_1 \to E_B \\ \deg \phi_1 \,=\, 3^{b-\beta_1}}}$$

Does this isogeny *exist*?
Is E_1 on the path between E_0 and E_B?
Use ***Kani's criterion*** for this decision!

Castryck, Decru – "An efficient key recovery attack on SIDH"

Strategy of the attack: **iterate** for a lot of steps!
2^{nd} step: $\beta = \beta_2 - \beta_1$

$$\phi_1 : \underbrace{E_1 \xrightarrow{\kappa_2} E_2}_{\substack{\kappa_2 : E_1 \twoheadrightarrow E_2 \\ \deg \kappa_2 = 3^{\beta}}} \underbrace{\xrightarrow{\qquad \phi_2 \qquad} E_B}_{\substack{\phi_2 : E_2 \to E_B \\ \deg \phi_2 = 3^{b-\beta}}}$$

Does this isogeny *exist*?
Is E_2 on the path between E_1 and E_B?
Use ***Kani's criterion*** for this decision!

A toolbox for breaking SIDH

Theorem (Petit's attack to unbalanced-SIDH, 2017)

Let an attacker know:

- d_A *and* d_B *sufficiently smooth coprime integers*
- $\{P_A, Q_A\}$ *the basis of* $E_0[d_A]$
- $\{\phi_B(P_A), \phi_B(Q_A)\}$ *a known basis of* $E_B[d_A]$ *(published by Bob)*

If $d_A \geq d_B^2$*, then the* d_B*-isogeny* $\phi_B : E_0 \to E_B$ *can be recovered efficiently.*

A toolbox for breaking SIDH

Theorem (SIDH attacks as a black-box, 2022)

Let an attacker know:

- d_A *and* d_B *sufficiently smooth coprime integers*
- $\{P_A, Q_A\}$ *the basis of* $E_0[d_A]$
- $\{\phi_B(P_A), \phi_B(Q_A)\}$ *a known basis of* $E_B[d_A]$ *(published by Bob)*

If $d_A \geq \sqrt{d_B}$, *then the* d_B*-isogeny* $\phi_B : E_0 \to E_B$ *can be recovered efficiently.*

Part 2

Countermeasures
&
New Constructions

M-SIDH & MD-SIDH – Overview

To make SIDH work, each party has to **reveal**:

- The **image of a torsion basis** under their secret isogeny
- The **degree** of their secret isogeny

But these cause SIDH to be **insecure**!

M-SIDH & MD-SIDH allow to **not throw away** the SIDH-framework, but at a hefty price:

- Much **larger** *public keys*, by a factor of at least 6.8 for same security
- Much **slower** *run-time*, by a factor of $O(\sqrt{\lambda}\log^{3/2}\lambda)$

M-SIDH – Masked torsion SIDH

- Goal: make the *image of the torsion basis* **not available** to the adversary, but still make the key exchange succeed.
- How? For each party:
  - **Scale** the *image of the torsion basis* by a random (secret) integer (delete it after usage)
    $\implies$ reveal less information
  - *Degree* of the isogeny is publicly known (as in SIDH), but it is different from an SIDH-degree ($d_A = \ell_A^a$ and $d_B = \ell_B^b$):
    $d_A = \prod_{i=1}^{t} \ell_i$ and $d_B = \prod_{i=1}^{t} q_i$ are coprime integers s.t. $d_A \approx d_B$

M-SIDH – Masked torsion SIDH

$d_A = \prod_{i=1}^{t} \ell_i$ and $d_B = \prod_{i=1}^{t} q_i$ are coprime integers s.t. $d_A \approx d_B$

Why degrees of this form?

Alice's *public key* is the tuple $\mathrm{pk}_A = (E_A, [\alpha]\phi_A(P_B), [\alpha]\phi_A(Q_B))$, with $\alpha \in \mu_2(d_B) := \{x \in \mathbb{Z}/d_B\mathbb{Z} \mid x^2 \equiv 1 \mod d_B\}$

$\implies$ there are an exponential number of square roots of 1 modulo d_B

$\implies$ the scalar cannot be recovered!

The rest of the protocol is analogous to SIDH

MD-SIDH – Masked Degree SIDH

- Goal: mask **both** the *degree* of the secret isogeny, and the *image of the torsion basis*.
- Idea:
  - make Alice use isogeny of degree d'_A s.t. $d'_A \mid d_A$
  - make Bob use isogeny of degree d'_B s.t. $d'_B \mid d_B$
- How? For each party:
  - **Scale** the *image of the torsion basis* by a random (secret) integer (delete it after usage)
    $\implies$ reveal less information
  - *Degree* of the isogeny is a random (secret) **divisor** (delete it after usage)
    $\implies$ reveal less information
    But d_A and d_B need to be different from the SIDH-degrees:
    $d_A = \prod_{i=1}^{t} \ell_i^{a_i}$ and $d_B = \prod_{i=1}^{t} q_i^{b_i}$ are coprime integers s.t. $d_A \approx d_B$

MD-SIDH – Masked Degree SIDH

$$d_A = \prod_{i=1}^{t} \ell_i^{a_i} \text{ and } d_B = \prod_{i=1}^{t} q_i^{b_i} \text{ are coprime integers s.t. } d_A \approx d_B$$

Why public parameters of this form?

SIDH has $d_A = \ell_A^a$ and $d_B = \ell_B^b$,
but this way we only have $a+1$ and $b+1$ possible divisors...

We want **more divisors** for more security!
Note: t depends on λ

The rest of the protocol is analogous to SIDH

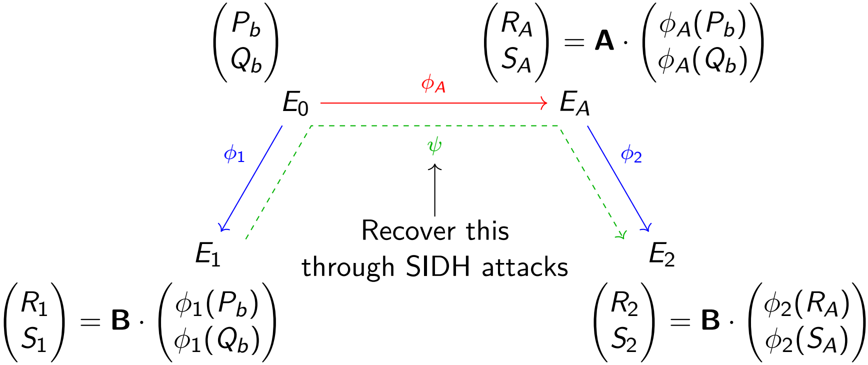
FESTA – Fast Encryption from Supersingular Torsion Attack

- Isogeny-based Public-key **Encryption** scheme
- How?
  Use SIDH attacks in a *constructive* way to create a trapdoor function f_{pk}
- Run-times:
  - KeyGen in 4.47 seconds
  - Enc in 3.09 seconds
  - Dec in 9.14 seconds

FESTA – The trapdoor function

Overview – 4 algorithms:

- $(E_0, P_b, Q_b) \leftarrow \mathsf{SetUp}(\lambda)$
- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\lambda)$ s.t. $\mathsf{sk} = (\mathbf{A}, \phi_A)$, $\mathsf{pk} = (E_A, R_A, S_A)$
- $(E_1, R_1, S_1, E_2, R_2, S_2) \leftarrow f_{\mathsf{pk}}(\phi_1, \phi_2, \mathbf{B})$
- $(\phi_1, \phi_2, \mathbf{B}) \leftarrow f_{\mathsf{pk}}^{-1}(E_1, R_1, S_1, E_2, R_2, S_2)$



FESTA – The trapdoor function

Overview – 4 algorithms:

- $(E_0, P_b, Q_b) \leftarrow \mathsf{SetUp}(\lambda)$
- $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(\lambda)$ s.t. $\mathsf{sk} = (\mathbf{A}, \phi_A)$, $\mathsf{pk} = (E_A, R_A, S_A)$
- $(E_1, R_1, S_1, E_2, R_2, S_2) \leftarrow f_{\mathsf{pk}}(\phi_1, \phi_2, \mathbf{B})$
- $(\phi_1, \phi_2, \mathbf{B}) \leftarrow f_{\mathsf{pk}}^{-1}(E_1, R_1, S_1, E_2, R_2, S_2)$

$$\begin{pmatrix} P_b \\ Q_b \end{pmatrix} \qquad \begin{pmatrix} R_A \\ S_A \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} \phi_A(P_b) \\ \phi_A(Q_b) \end{pmatrix}$$

$$\begin{tikzcd}
 & E_0 \arrow[r, "\phi_A"] \arrow[dl, "\phi_1"'] & E_A \arrow[dr, "\phi_2"] & \\
E_1 \arrow[rrr, dashed, "\psi"] & & & E_2
\end{tikzcd}$$

$$\begin{pmatrix} \psi(R_1) \\ \psi(S_1) \end{pmatrix} = [\deg \phi_1] \cdot \mathbf{A}^{-1} \cdot \begin{pmatrix} R_2 \\ S_2 \end{pmatrix}$$

$$\deg \psi = d_1 \cdot d_A \cdot d_2$$

$$\begin{pmatrix} R_1 \\ S_1 \end{pmatrix} = \mathbf{B} \cdot \begin{pmatrix} \phi_1(P_b) \\ \phi_1(Q_b) \end{pmatrix} \qquad \begin{pmatrix} R_2 \\ S_2 \end{pmatrix} = \mathbf{B} \cdot \begin{pmatrix} \phi_2(R_A) \\ \phi_2(S_A) \end{pmatrix}$$

POKE – Point-based Key Exchange

- POKE at the moment is:
  - The **most compact** post-quantum PKE ($p \approx 2^{3\lambda}$)
  - The **most efficient** isogeny-based PKE (runtime $\approx$ 0.3 seconds)
- How? Alice and Bob use different ***types*** of isogenies:

Name	**Description**	**Alice**	**Bob**
SIDH isogeny	Rational 1-dimensional	✓	✓
FESTA isogeny	Irrational 2-dimensional	✓	✗

- Different types of isogenies have different ***representations***

POKE – Isogeny representations

- Different types of isogenies have different ***representations***:
  1. SIDH isogenies:
     - Representation: 1 curve (domain), 1 kernel
     - To close the commutative diagram, must reveal the **randomly scaled** image of the torsion basis under the secret isogeny (M-SIDH)
  2. FESTA isogenies:
     - Representation: 2 curves (domain, codomain), degree, image of the 2^a-torsion basis (*constructive* application of SIDH attacks)
     - To close the commutative diagram, ***POKE construction***

Name	**Description**	**Alice**	**Bob**
SIDH isogeny	Rational 1-dimensional	✓	✓
FESTA isogeny	Irrational 2-dimensional	✓	✗

POKE construction

$$
\begin{tikzcd}
E_0,\ X_0 \in E_0[x] \arrow[r, "\phi_A"] \arrow[d, "\phi_B"'] & E_A,\ X_A = [\alpha]\phi_A(X_0) \arrow[d, "\phi'_B"] \arrow[dl, dashed, shift left] \\
E_B,\ X_B = [\beta]\phi_B(X_0) \arrow[r, "\phi'_A"'] \arrow[ur, dashed, shift left] & E_{AB},\ X_{AB} = [\alpha]\phi'_A(X_B) = [\beta]\phi'_B(X_A)
\end{tikzcd}
$$

Part 3

Conclusions

Conclusions

Key insights:

- SIDH attacks were a temporary setback, but led to new developments
- Optimism that robust quantum-resistant solutions can be achieved

Thank you for your attention!

Questions?

References I

[De 17] Luca De Feo. "Mathematics of isogeny based cryptography". In: *arXiv preprint arXiv: 1711.04062* (2017). DOI: 10.48550/arXiv.1711.04062.

[Pet17] Christophe Petit. "Faster algorithms for isogeny problems using torsion point images". In: *Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II 23*. Springer. 2017, pp. 330–353. DOI: 10.1007/978-3-319-70697-9_12.

References II

[CD23] Wouter Castryck and Thomas Decru. "An efficient key recovery attack on SIDH". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 423–447. DOI: 10.1007/978-3-031-30589-4_15.

[Mai+23] Luciano Maino et al. "A Direct Key Recovery Attack on SIDH". In: *Advances in Cryptology – EUROCRYPT 2023*. Ed. by Carmit Hazay and Martijn Stam. Cham: Springer Nature Switzerland, 2023, pp. 448–471. ISBN: 978-3-031-30589-4. DOI: 10.1007/978-3-031-30589-4_16.

[Rob23] Damien Robert. "Breaking SIDH in polynomial time". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2023, pp. 472–503. DOI: 10.1007/978-3-031-30589-4_17.