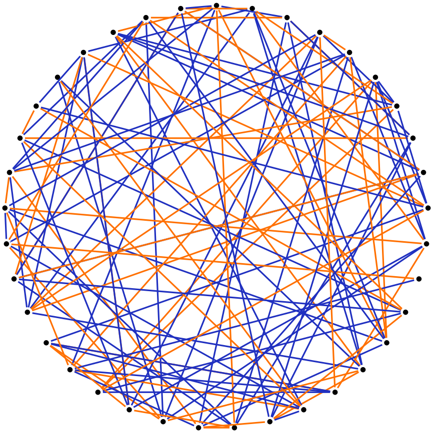
Isogeny-basd cryptography VI
SIDH

Tanja Lange
(with lots of slides by Lorenz Panny)

Eindhoven University of Technology

SAC – Post-quantum cryptography

SIDH – consider extension fields



The supersingular isogeny graph over $\mathbb{F}_{p^2}$ looks differently.

Isomorphism classes now taking isomorphisms over any extension field.
Each node is one j invariant, all classed are defined over $\mathbb{F}_{p^2}$.

SIDH: High-level view (2011 Jao–De Feo)

Problem: quadratic twists are identified, $\ell + 1$ isogenies of degree ℓ from any curve, no more sense of direction.

SIDH: High-level view (2011 Jao–De Feo)

Problem: quadratic twists are identified, $\ell + 1$ isogenies of degree ℓ from any curve, no more sense of direction.

$$
\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}
$$

SIDH: High-level view (2011 Jao–De Feo)

Problem: quadratic twists are identified, $\ell + 1$ isogenies of degree ℓ from any curve, no more sense of direction.

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.

SIDH: High-level view (2011 Jao–De Feo)

Problem: quadratic twists are identified, $\ell + 1$ isogenies of degree ℓ from any curve, no more sense of direction.

$$\begin{tikzcd} E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\ E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle \end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A \colon E \to E/A$; Bob computes $\varphi_B \colon E \to E/B$.
  (These isogenies correspond to walking on the isogeny graph.)

SIDH: High-level view (2011 Jao–De Feo)

Problem: quadratic twists are identified, $\ell + 1$ isogenies of degree ℓ from any curve, no more sense of direction.

$$\begin{tikzcd} E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\ E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle \end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A \colon E \to E/A$; Bob computes $\varphi_B \colon E \to E/B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the values E/A and E/B.

SIDH: High-level view (2011 Jao–De Feo)

Problem: quadratic twists are identified, $\ell+1$ isogenies of degree ℓ from any curve, no more sense of direction.

$$\begin{tikzcd}
E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\
E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle
\end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A\colon E \to E/A$; Bob computes $\varphi_B\colon E \to E/B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the values E/A and E/B.
- Alice <u>somehow</u> obtains $A' := \varphi_B(A)$. (Similar for Bob.)

SIDH: High-level view (2011 Jao–De Feo)

Problem: quadratic twists are identified, $\ell+1$ isogenies of degree ℓ from any curve, no more sense of direction.

$$\begin{tikzcd} E \arrow[r, "\varphi_A"] \arrow[d, "\varphi_B"'] & E/A \arrow[d, "\varphi_{B'}"] \\ E/B \arrow[r, "\varphi_{A'}"'] & E/\langle A, B\rangle \end{tikzcd}$$

- Alice & Bob pick secret subgroups A and B of E.
- Alice computes $\varphi_A\colon E \to E/A$; Bob computes $\varphi_B\colon E \to E/B$.
  (These isogenies correspond to walking on the isogeny graph.)
- Alice and Bob transmit the values E/A and E/B.
- Alice somehow obtains $A' := \varphi_B(A)$. (Similar for Bob.)
- They both compute the shared secret
  $$(E/B)/A' \cong E/\langle A, B\rangle \cong (E/A)/B'.$$
- Key is an isomorphism class; make this usable taking j-invariant.

SIDH's auxiliary points

Previous slide: "Alice somehow obtains $A' := \varphi_B(A)$."

Alice knows only A, Bob knows only φ_B.

SIDH's auxiliary points

Previous slide: "Alice <u>somehow</u> obtains $A' := \varphi_B(A)$."

Alice knows only A, Bob knows only φ_B.

- Alice picks A as $\langle P + [a]Q \rangle$ for fixed public $P, Q \in E$.
- Bob includes $\varphi_B(P)$ and $\varphi_B(Q)$ in his public key.

SIDH's auxiliary points

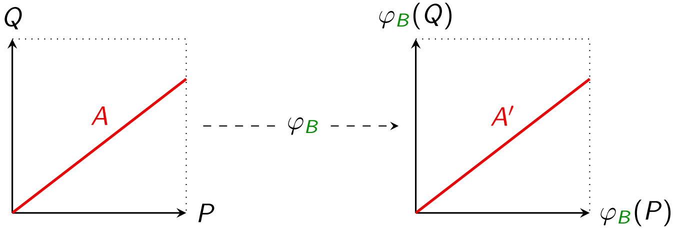
Previous slide: "Alice <u>somehow</u> obtains $A' := \varphi_B(A)$."

Alice knows only A, Bob knows only φ_B.

<u>Solution</u>: φ_B is a group homomorphism!

- Alice picks A as $\langle P + [a]Q \rangle$ for fixed public $P, Q \in E$.
- Bob includes $\varphi_B(P)$ and $\varphi_B(Q)$ in his public key.

$\Longrightarrow$ Now Alice can compute A' as $\langle \varphi_B(P) + [a]\varphi_B(Q) \rangle$!



Using images of P and Q also lets Alice keep direction in iterative computation of φ_A.

SIDH in one slide

Public parameters:

- large prime $p = 2^n 3^m - 1$, supersingular $E/\mathbb{F}_{p^2}$ with $(p+1)^2$ points.
- bases (P, Q) and (R, S) of $E[2^n]$ and $E[3^m]$.
  Want these points defined over $\mathbb{F}_{p^2}$ for efficiency.
  Parameter choice ensures this. Recall $E[\ell] \cong \mathbb{Z}/\ell \times \mathbb{Z}/\ell$.

<u>Alice</u>	<u>public</u>	<u>Bob</u>
$a \xleftarrow{\text{random}} \{0, \ldots, 2^n-1\}$		$b \xleftarrow{\text{random}} \{0, \ldots, 3^m-1\}$
$A = \langle P + [a]Q \rangle$		$B = \langle R + [b]S \rangle$
compute $\varphi_A \colon E \to E/A$		compute $\varphi_B \colon E \to E/B$
$E/A,\ \varphi_A(R),\ \varphi_A(S)$	$\longrightarrow$ / $\longleftarrow$	$E/B,\ \varphi_B(P),\ \varphi_B(Q)$
$A' = \langle \varphi_B(P) + [a]\varphi_B(Q) \rangle$		$B' = \langle \varphi_A(R) + [b]\varphi_A(S) \rangle$
$s = j\big((E/B)/A'\big)$		$s = j\big((E/A)/B'\big)$

Decomposing smooth isogenies

- In SIDH, $\#A = 2^n$ and $\#B = 3^m$ are "crypto-sized"
  Vélu's formulas take $\Theta(\#G)$ to compute $\varphi_G \colon E \to E/G$.

Decomposing smooth isogenies

- In SIDH, $\#A = 2^n$ and $\#B = 3^m$ are "crypto-sized"
  Vélu's formulas take $\Theta(\#G)$ to compute $\varphi_G \colon E \to E/G$.

!! Evaluate φ_G as a chain of small-degree isogenies:
For $G \cong \mathbb{Z}/\ell^k$, set $\ker \psi_i = [\ell^{k-i}](\psi_{i-1} \circ \cdots \circ \psi_1)(G)$.

$$\begin{tikzcd} E \arrow[r, "\psi_1"] \arrow[rrrr, bend right, "\varphi_G"'] & E_1 \arrow[r, "\psi_2"] & \ldots \arrow[r, "\psi_{k-1}"] & E_{k-1} \arrow[r, "\psi_k"] & E/G \end{tikzcd}$$

Decomposing smooth isogenies

- In SIDH, $\#A = 2^n$ and $\#B = 3^m$ are "crypto-sized"
  Vélu's formulas take $\Theta(\#G)$ to compute $\varphi_G \colon E \to E/G$.

!! Evaluate φ_G as a chain of small-degree isogenies:
For $G \cong \mathbb{Z}/\ell^k$, set $\ker \psi_i = [\ell^{k-i}](\psi_{i-1} \circ \cdots \circ \psi_1)(G)$.

$$\begin{tikzcd} E \arrow[r, "\psi_1"] \arrow[rrrr, bend right, "\varphi_G"'] & E_1 \arrow[r, "\psi_2"] & \ldots \arrow[r, "\psi_{k-1}"] & E_{k-1} \arrow[r, "\psi_k"] & E/G \end{tikzcd}$$

$\rightsquigarrow$ Complexity: $O(k^2 \cdot \ell)$. Exponentially smaller than ℓ^k!
"Optimal strategy" improves this to $O(k \log k \cdot \ell)$.

Decomposing smooth isogenies

- In SIDH, $\#A = 2^n$ and $\#B = 3^m$ are "crypto-sized"
  Vélu's formulas take $\Theta(\#G)$ to compute $\varphi_G \colon E \to E/G$.

!! Evaluate φ_G as a chain of small-degree isogenies:
For $G \cong \mathbb{Z}/\ell^k$, set $\ker \psi_i = [\ell^{k-i}](\psi_{i-1} \circ \cdots \circ \psi_1)(G)$.

$$\begin{tikzcd} E \arrow[r, "\psi_1"] \arrow[rrrr, bend right, "\varphi_G"'] & E_1 \arrow[r, "\psi_2"] & \ldots \arrow[r, "\psi_{k-1}"] & E_{k-1} \arrow[r, "\psi_k"] & E/G \end{tikzcd}$$

$\rightsquigarrow$ Complexity: $O(k^2 \cdot \ell)$. Exponentially smaller than ℓ^k!
"Optimal strategy" improves this to $O(k \log k \cdot \ell)$.

- BTW: The choice of p makes sure everything stays over $\mathbb{F}_{p^2}$.

Security of SIDH

The SIDH graph has size $\lfloor p/12 \rfloor + \varepsilon$.
Each secret isogeny φ_A, φ_B is a walk of about $\log p/2$ steps.
Alice & Bob can choose from about $\sqrt{p}$ secret keys each,
so their keys are in small corners of the key space.

Security of SIDH

The SIDH graph has size $\lfloor p/12 \rfloor + \varepsilon$.
Each secret isogeny φ_A, φ_B is a walk of about $\log p/2$ steps.
Alice & Bob can choose from about $\sqrt{p}$ secret keys each,
so their keys are in small corners of the key space.

Classical attacks:

- Cannot reuse keys without extra caution. (next slide)
- Meet-in-the-middle: $\tilde{\mathcal{O}}(p^{1/4})$ time & space.
- Collision finding: $\tilde{\mathcal{O}}(p^{3/8}/\sqrt{memory}/cores)$.

Security of SIDH

The SIDH graph has size $\lfloor p/12 \rfloor + \varepsilon$.
Each secret isogeny φ_A, φ_B is a walk of about $\log p/2$ steps.
Alice & Bob can choose from about $\sqrt{p}$ secret keys each,
so their keys are in small corners of the key space.

Classical attacks:

- Cannot reuse keys without extra caution. (next slide)
- Meet-in-the-middle: $\tilde{\mathcal{O}}(p^{1/4})$ time & space.
- Collision finding: $\tilde{\mathcal{O}}(p^{3/8}/\sqrt{memory}/cores)$.

Quantum attacks:

- Claw finding: claimed $\tilde{\mathcal{O}}(p^{1/6})$. 2019 Jaques–Schank: $\tilde{\mathcal{O}}(p^{1/4})$:
  "An adversary with enough quantum memory to run Tani's algorithm with the query-optimal parameters could break SIKE faster by using the classical control hardware to run van Oorschot–Wiener."

Thou shalt not reuse SIDH keys

- Recall: Bob sends $P' = \varphi_B(P)$ and $Q' = \varphi_B(Q)$ to Alice.
  She computes $A' = \langle P' + [a]Q' \rangle$ and, from that, obtains s.

Thou shalt not reuse SIDH keys

- Recall: Bob sends $P' = \varphi_B(P)$ and $Q' = \varphi_B(Q)$ to Alice. She computes $A' = \langle P' + [a]Q' \rangle$ and, from that, obtains s.
- Bob cheats and sends $Q'' = Q' + [2^{n-1}]P'$ instead of Q'. Alice computes $A'' = \langle P' + [a]Q'' \rangle$.

Thou shalt not reuse SIDH keys

- Recall: Bob sends $P' = \varphi_B(P)$ and $Q' = \varphi_B(Q)$ to Alice. She computes $A' = \langle P' + [a]Q' \rangle$ and, from that, obtains s.
- Bob cheats and sends $Q'' = Q' + [2^{n-1}]P'$ instead of Q'. Alice computes $A'' = \langle P' + [a]Q'' \rangle$.

  If $a = 2u$: $[a]Q'' = [a]Q' + [u][2^n]P' = [a]Q'$.
  If $a = 2u+1$:
  $[a]Q'' = [a]Q' + [u][2^n]P' + [2^{n-1}]P' = [a]Q' + [2^{n-1}]P'$.

Thou shalt not reuse SIDH keys

- Recall: Bob sends $P' = \varphi_B(P)$ and $Q' = \varphi_B(Q)$ to Alice. She computes $A' = \langle P' + [a]Q' \rangle$ and, from that, obtains s.
- Bob cheats and sends $Q'' = Q' + [2^{n-1}]P'$ instead of Q'. Alice computes $A'' = \langle P' + [a]Q'' \rangle$.

  If $a = 2u$: $[a]Q'' = [a]Q' + [u][2^n]P' = [a]Q'$.
  If $a = 2u+1$:
  $[a]Q'' = [a]Q' + [u][2^n]P' + [2^{n-1}]P' = [a]Q' + [2^{n-1}]P'$.

  $\Longrightarrow$ Bob learns the parity of a.

Thou shalt not reuse SIDH keys

- Recall: Bob sends $P' = \varphi_B(P)$ and $Q' = \varphi_B(Q)$ to Alice.
  She computes $A' = \langle P' + [a]Q' \rangle$ and, from that, obtains s.

- Bob cheats and sends $Q'' = Q' + [2^{n-1}]P'$ instead of Q'.
  Alice computes $A'' = \langle P' + [a]Q'' \rangle$.

  If $a = 2u$: $[a]Q'' = [a]Q' + [u][2^n]P' = [a]Q'$.
  If $a = 2u+1$:
  $[a]Q'' = [a]Q' + [u][2^n]P' + [2^{n-1}]P' = [a]Q' + [2^{n-1}]P'$.

  $\Longrightarrow$ Bob learns the parity of a.

  Similarly, he can completely recover a in $O(n)$ queries.

Thou shalt not reuse SIDH keys

- Recall: Bob sends $P' = \varphi_B(P)$ and $Q' = \varphi_B(Q)$ to Alice.
  She computes $A' = \langle P' + [a]Q' \rangle$ and, from that, obtains s.
- Bob cheats and sends $Q'' = Q' + [2^{n-1}]P'$ instead of Q'.
  Alice computes $A'' = \langle P' + [a]Q'' \rangle$.

  If $a = 2u$: $[a]Q'' = [a]Q' + [u][2^n]P' \quad = [a]Q'$.
  If $a = 2u+1$:
  $[a]Q'' = [a]Q' + [u][2^n]P' + [2^{n-1}]P' = [a]Q' + [2^{n-1}]P'$.

  $\Longrightarrow$ Bob learns the parity of a.

  Similarly, he can completely recover a in $O(n)$ queries.

Validating that Bob is honest is $\approx$ as hard as breaking SIDH.

$\Longrightarrow$ only usable with ephemeral keys or as a KEM *"SIKE."*.

Comparison

Key bits where all known attacks take 2^λ operations (naive serial attack metric, ignoring memory cost):

	pre-quantum	post-quantum
SIDH, SIKE	$(24 + o(1))\lambda$	$(36 + o(1))\lambda$
compressed	$(14 + o(1))\lambda$	$(21 + o(1))\lambda$
CSIDH	$(4 + o(1))\lambda$	superlinear
ECDH	$(2 + o(1))\lambda$	exponential

Find more attacks on SIDH.

See "How to not break SIDH" `https://eprint.iacr.org/2019/558` by Chloe Martindale and Lorenz Panny for some failed attempts.