CSIDH:
An Efficient Post-Quantum Commutative Group Action

Wouter Castryck\textsuperscript{1}  Tanja Lange\textsuperscript{2}  Chloe Martindale\textsuperscript{2}  Lorenz Panny\textsuperscript{2}  Joost Renes\textsuperscript{3}

\textsuperscript{1}KU Leuven  \textsuperscript{2}TU Eindhoven  \textsuperscript{3}Radboud Universiteit

Crypto Working Group, Utrecht, 14 September 2018
six, said
Why CSIDH?

- Drop-in post-quantum replacement for (EC)DH
Why CSIDH?

- Drop-in post-quantum replacement for (EC)DH
- Non-interactive key exchange (full public-key validation); previously an open problem post-quantumly (w/ reasonable speed)
Why CSIDH?

- Drop-in post-quantum replacement for (EC)DH
- Non-interactive key exchange (full public-key validation); previously an open problem post-quantumly (w/ reasonable speed)
- Small keys: 64 bytes at conjectured AES-128 security level
Why CSIDH?

- Drop-in post-quantum replacement for (EC)DH
- Non-interactive key exchange (full public-key validation); previously an open problem post-quantumly (w/ reasonable speed)
- Small keys: 64 bytes at conjectured AES-128 security level
- Competitive speed: ~85 ms for a full key exchange
Why CSIDH?

- **Drop-in** post-quantum replacement for (EC)DH
- **Non-interactive key exchange** (full public-key validation); previously an open problem post-quantumly (w/ reasonable speed)
- **Small** keys: 64 bytes at conjectured AES-128 security level
- **Competitive** speed: $\sim 85$ ms for a full key exchange
- **Flexible**: compatible with 0-RTT protocols such as QUIC; recent preprint uses CSIDH for ‘SeaSign’ signatures
Post-quantum Diffie-Hellman?

Traditionally, Diffie-Hellman works in a group $G$ via the map

$$\mathbb{Z} \times G \rightarrow G$$

$$(x, g) \mapsto g^x.$$
Post-quantum Diffie-Hellman?

Traditionally, Diffie-Hellman works in a group $G$ via the map

$$\mathbb{Z} \times G \rightarrow G$$

$$(x, g) \mapsto g^x.$$

Shor’s algorithm quantumly computes $x$ from $g^x$ in any group in polynomial time.
Post-quantum Diffie-Hellman!

Traditionally, Diffie-Hellman works in a group $G$ via the map

$$\mathbb{Z} \times G \rightarrow G \quad (x, g) \mapsto g^x.$$

Shor’s algorithm quantumly computes $x$ from $g^x$ in any group in polynomial time.

$\Rightarrow$ Idea:

Replace exponentiation on the group $G$ by a group action of a group $H$ on a set $S$:

$$H \times S \rightarrow S.$$
Square-and-multiply

Suppose $G \cong \mathbb{Z}/23$ and that Alice computes $g^{13}$. 
Square-and-multiply

Suppose $G \cong \mathbb{Z}/23$ and that Alice computes $g^{13}$. 
Square-and-multiply

Suppose \( G \cong \mathbb{Z}/23 \) and that Alice computes \( g^{13} \).
Square-and-multiply

Suppose $G \cong \mathbb{Z}/23$ and that Alice computes $g^{13}$. 
Square-and-multiply

Suppose $G \cong \mathbb{Z}/23$ and that Alice computes $g^{13}$. 
Cycles are compatible: [right, then left] = [left, then right], etc.
Cycles are compatible: [right, then left] = [left, then right], etc.
Cycles are compatible: [right, then left] = [left, then right], etc.

\[
\begin{array}{c}
\text{Cycles!} \\
\end{array}
\]
Cycles are compatible: [right, then left] = [left, then right], etc.
Union of cycles: rapid mixing

CSIDH: Nodes are now elliptic curves and edges are isogenies.
Union of cycles: rapid mixing

CSIDH: Nodes are now \textit{elliptic curves} and edges are \textit{isogenies}.
Graphs of elliptic curves

Nodes: Supersingular curves

**$E_A$:** $y^2 = x^3 + Ax^2 + x$ over $F_{419}$.

Edges: 3-, 5-, and 7-isogenies (certain kinds of maps).
Graphs of elliptic curves

Nodes: Supersingular curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
Graphs of elliptic curves

Nodes: Supersingular curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_{419}$.
Edges: 3-, 5-, and 7-isogenies (certain kinds of maps).
Graphs of elliptic curves

A 3-isogeny

$E_{51}: y^2 = x^3 + 51x^2 + x$  $\longrightarrow$  $E_9: y^2 = x^3 + 9x^2 + x$

$(x, y) \longmapsto \left( \frac{97x^3 - 183x^2 + x}{x^2 - 183x + 97}, \frac{133x^3 + 154x^2 - 5x + 97}{-x^3 + 65x^2 + 128x - 133} \right)
Diffie-Hellman on ‘nice’ graphs

Alice
[+, −, +, −]

Bob
[+, +, −, +]
Diffie-Hellman on ‘nice’ graphs

Alice
\[ [+,-,+,-] \]

Bob
\[ [+,-,-,+] \]
Diffie-Hellman on ‘nice’ graphs

Alice

$[+, -, +, -]$

Bob

$[+, -, +, +]$
Diffie-Hellman on ‘nice’ graphs

Alice

[+, −, +, −]

Bob

[+, +, −, +]
Diffie-Hellman on ‘nice’ graphs

Alice

[+, −, +, −]

Bob

[+, +, −, +]
Diffie-Hellman on ‘nice’ graphs

Alice: [+ , −, +, −]

Bob: [+ , +, −, +]
Diffie-Hellman on ‘nice’ graphs

Alice

Bob

\[
\begin{bmatrix}
+, -, +, - \\
\uparrow
\end{bmatrix}
\]

\[
\begin{bmatrix}
+, +, -, + \\
\uparrow
\end{bmatrix}
\]
Diffie-Hellman on ‘nice’ graphs

Alice

\[ [+, -, +, -] \]

Bob

\[ [+, +, -, +] \]
Diffie-Hellman on ‘nice’ graphs

Alice
\[ [+,-,+,-] \]

Bob
\[ [+,-,+] \]
Diffie-Hellman on ‘nice’ graphs

Alice

[+, −, +, −]

Bob

[+, +, −, +]
Diffie-Hellman on ‘nice’ graphs

Alice  
[+, −, +, −]

Bob  
[+, +, −, +]
A walkable graph

Important properties for such a walk:

IP1  ➤ The graph is a composition of compatible cycles.
IP2  ➤ We can compute neighbours in given directions.
The graph used in CSIDH is constructed as a composition of graphs $G_\ell$ of $\ell$-isogenies.
IP1: A composition of cycles

- The graph used in CSIDH is constructed as a composition of graphs $G_\ell$ of ‘\(\ell\)-isogenies’.

- In our example, these are

```
G_3:
```

![Graph diagram with labeled points and connections]
The graph used in CSIDH is constructed as a composition of graphs $G_\ell$ of ‘$\ell$-isogenies’.

In our example, these are

$G_5$: 

![Diagram showing the graph $G_5$ with labeled points $E_0, E_{158}, E_{410}, \ldots$]
IP1: A composition of cycles

- The graph used in CSIDH is constructed as a composition of graphs $G_\ell$ of $\ell$-isogenies.

- In our example, these are $E_{0}E_{158}E_{410}E_{368}E_{404}E_{75}E_{144}E_{191}E_{174}E_{413}E_{379}E_{124}E_{199}E_{390}E_{29}E_{220}E_{295}E_{40}E_{6}E_{245}E_{228}E_{275}E_{344}E_{15}E_{51}E_{9}E_{261}$. 

$G_7$: 

![Graph diagram](image_url)
IP1: A composition of cycles

- The graph used in CSIDH is constructed as a composition of graphs $G_\ell$ of $\ell$-isogenies.

- In our example, these are

$$G_3 \cup G_5 \cup G_7:$$
The graph used in CSIDH is constructed as a composition of graphs $G_\ell$ of $\ell$-isogenies.

Generally, the $G_\ell$ look something like $G_3$ and $G_5$: 

- $G_3$: 
- $G_5$:
IP1: A composition of cycles

- The graph used in CSIDH is constructed as a composition of graphs $G_\ell$ of $\ell$-isogenies.

- Generally, the $G_\ell$ look something like

- $G_3$:

- $G_5$:

- We want to make sure $G_\ell$ is just a cycle.
**IP2: Compute neighbours in given directions**

The edges of $G_\ell$ are $\ell$-isogenies.

$$E_{51}: y^2 = x^3 + 51x^2 + x \quad \longrightarrow \quad E_9: y^2 = x^3 + 9x^2 + x$$

$$(x, y) \quad \longrightarrow \quad \left( \frac{97x^3 - 183x^2 + x}{x^2 - 183x + 97}, y \cdot \frac{133x^3 + 154x^2 - 5x + 97}{-x^3 + 65x^2 + 128x - 133} \right)$$
IP2: Compute neighbours in given directions

The edges of $G_\ell$ are $\ell$-isogenies.

\[ E_{51} : y^2 = x^3 + 51x^2 + x \quad \longrightarrow \quad E_9 : y^2 = x^3 + 9x^2 + x \]

\[ (x, y) \quad \longrightarrow \quad \left( \frac{97x^3 - 183x^2 + x}{x^2 - 183x + 97} , y \cdot \frac{133x^3 + 154x^2 - 5x + 97}{-x^3 + 65x^2 + 128x - 133} \right) \]

- The orientation of $G_\ell$ is mathematically well-defined (canonical way to compute the ‘left’ or ‘right’ isogeny).
The edges of $G_\ell$ are $\ell$-isogenies.

$E_{51}: y^2 = x^3 + 51x^2 + x \quad \longrightarrow \quad E_9: y^2 = x^3 + 9x^2 + x$

$(x, y) \quad \longmapsto \quad \left( \frac{97x^3 - 183x^2 + x}{x^2 - 183x + 97}, y \cdot \frac{133x^3 + 154x^2 - 5x + 97}{-x^3 + 65x^2 + 128x - 133} \right)$

- **The orientation** of $G_\ell$ is mathematically well-defined (canonical way to compute the ‘left’ or ‘right’ isogeny).
- **The cost grows** with $\ell \rightsquigarrow$ want small $\ell$. 
IP2: Compute neighbours in given directions

The edges of $G_\ell$ are $\ell$-isogenies.

$$E_{51} : y^2 = x^3 + 51x^2 + x \quad \longrightarrow \quad E_9 : y^2 = x^3 + 9x^2 + x$$

$$(x, y) \longmapsto \left( \frac{97x^3 - 183x^2 + x}{x^2 - 183x + 97}, y \cdot \frac{133x^3 + 154x^2 - 5x + 97}{-x^3 + 65x^2 + 128x - 133} \right)$$

- The orientation of $G_\ell$ is mathematically well-defined (canonical way to compute the ‘left’ or ‘right’ isogeny).
- The cost grows with $\ell \sim \text{want small } \ell$.
- Generally needs big extension fields...
Point counting

Both ‘IP’s are connected to the number of points on the curves.
Point counting

Both ‘IP’s are connected to the number of points on the curves.

It seems difficult to find a curve with a given number of points (and such that the graph is big). [De Feo–Kieffer–Smith]
Salt water (CSIDH, get it?) is a solution (do bad chemistry jokes belong in crypto talks?)

1. ▶ Choose some small odd primes $\ell_1, \ldots, \ell_n$. 

magic math happens!

3. ▶ $E_0$ has $p + 1$ points.
▶ Let the nodes of $G_\ell$ be those $E_A$ with $p + 1$ points.
▶ Then every $G_\ell$ is a disjoint union of cycles.
▶ All $G_\ell$ are compatible.
▶ Computations need only $F_p$-arithmetic.
Salt water (CSIDH, get it?) is a solution (do bad chemistry jokes belong in crypto talks?)

1. Choose some small odd primes \( \ell_1, \ldots, \ell_n \).
   Make sure \( p = 4 \cdot \ell_1 \cdots \ell_n - 1 \) is prime.
1. ▶ Choose some small odd primes $\ell_1, \ldots, \ell_n$.
   ▶ Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
   ▶ Fix the curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$. 

Salt water (CSIDH, get it?) is a solution (do bad chemistry jokes belong in crypto talks?)
Salt water (CSIDH, get it?) is a solution (do bad chemistry jokes belong in crypto talks?)

1. ▶ Choose some small odd primes $\ell_1, \ldots, \ell_n$.
   ▶ Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
   ▶ Fix the curve $E_0: y^2 = x^3 + x$ over $\mathbb{F}_p$.

2. ✨ magie math happens!
Salt water (CSIDH, get it?) is a solution (do bad chemistry jokes belong in crypto talks?)

1. ▶ Choose some small odd primes $\ell_1, \ldots, \ell_n$.
   ▶ Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
   ▶ Fix the curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$.

2. ✨ magic math happens!

3. ▶ $E_0$ has $p + 1$ points.
Salt water (CSIDH, get it?) is a solution (do bad chemistry jokes belong in crypto talks?)

1. ▶ Choose some small odd primes $\ell_1, \ldots, \ell_n$.
   ▶ Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
   ▶ Fix the curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$.

2. ✩ magic math happens!

3. ▶ $E_0$ has $p + 1$ points.
   ▶ Let the nodes of $G_{\ell_i}$ be those $E_A$ with $p + 1$ points.
Salt water (CSIDH, get it?) is a solution (do bad chemistry jokes belong in crypto talks?)

1. ▶ Choose some small odd primes \( \ell_1, \ldots, \ell_n \).
   ▶ Make sure \( p = 4 \cdot \ell_1 \cdots \ell_n - 1 \) is prime.
   ▶ Fix the curve \( E_0 : y^2 = x^3 + x \) over \( \mathbb{F}_p \).

2. ⚡️ magic math happens!

3. ▶ \( E_0 \) has \( p + 1 \) points.
   ▶ Let the nodes of \( G_{\ell_i} \) be those \( E_A \) with \( p + 1 \) points.
   ▶ Then every \( G_{\ell_i} \) is a disjoint union of cycles.
Salt water (CSIDH, get it?) is a solution (do bad chemistry jokes belong in crypto talks?)

1. ▶ Choose some small odd primes $\ell_1, \ldots, \ell_n$.
   ▶ Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
   ▶ Fix the curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$.

2. ✶ magic math happens!

3. ▶ $E_0$ has $p + 1$ points.
   ▶ Let the nodes of $G_{\ell_i}$ be those $E_A$ with $p + 1$ points.
   ▶ Then every $G_{\ell_i}$ is a disjoint union of cycles.
   ▶ All $G_{\ell_i}$ are compatible.
Salt water (CSIDH, get it?) is a solution (do bad chemistry jokes belong in crypto talks?)

1. ▶ Choose some small odd primes $\ell_1, \ldots, \ell_n$.
   ▶ Make sure $p = 4 \cdot \ell_1 \cdots \ell_n - 1$ is prime.
   ▶ Fix the curve $E_0: y^2 = x^3 + x$ over $\mathbb{F}_p$.

2. ➡️ magic math happens!

3. ▶ $E_0$ has $p + 1$ points.
   ▶ Let the nodes of $G_{\ell_i}$ be those $E_A$ with $p + 1$ points.
   ▶ Then every $G_{\ell_i}$ is a disjoint union of cycles.
   ▶ All $G_{\ell_i}$ are compatible.
   ▶ Computations need only $\mathbb{F}_p$-arithmetic.
Representing nodes of the graph

Side effect of magic:

- Every node of $G_{\ell_i}$ can be written as

$$E_A : y^2 = x^3 + Ax^2 + x.$$
Representing nodes of the graph

Side effect of magic:

- Every node of $G_{\ell_i}$ can be written as

$$E_A : y^2 = x^3 + Ax^2 + x.$$  

$\Rightarrow$ Can compress every node to a single value $A \in \mathbb{F}_p$. 
Representing nodes of the graph

Side effect of magic:

- Every node of $G_{\ell_i}$ can be written as

  $$E_A : y^2 = x^3 + Ax^2 + x.$$ 

$\Rightarrow$ Can compress every node to a single value $A \in \mathbb{F}_p$.  
$\Rightarrow$ Tiny keys!
Does any $A$ work?

---

1. This algorithm has a small chance of false positives, but we actually use a variant that proves that $E_A$ has $p + 1$ points.
Does any $A$ work?

No.

---

\[^1\text{This algorithm has a small chance of false positives, but we actually use a variant that } \text{proves} \text{ that } E_A \text{ has } p + 1 \text{ points.}\]
Does any $A$ work?

No.

- About $\sqrt{p}$ of all $A \in \mathbb{F}_p$ are valid keys.

---

$^1$This algorithm has a small chance of false positives, but we actually use a variant that proves that $E_A$ has $p + 1$ points.
Does any $A$ work?

No.

- About $\sqrt{p}$ of all $A \in \mathbb{F}_p$ are valid keys.
- **Public-key validation**: Check that $E_A$ has $p + 1$ points.
  
  Easy Monte-Carlo algorithm: Pick random $P$ on $E_A$ and check $[p + 1]P = \infty$.\(^1\)

\(^1\)This algorithm has a small chance of false positives, but we actually use a variant that *proves* that $E_A$ has $p + 1$ points.
Security

Classical:
- Meet-in-the-middle variants: Time $O\left(\sqrt[4]{p}\right)$. 

Quantum:
- Hidden-shift algorithms: Subexponential complexity.
- Literature contains mostly asymptotics.
- Time-space trade-off: Fastest variants need huge memory.
- Concrete estimates are to be done.
- (Recent preprint [BS] ignores much of the cost!)
Security

Classical:

- **Meet-in-the-middle** variants: Time $O(\sqrt[p]{p})$.

Quantum:

- **Hidden-shift** algorithms: *Subexponential* complexity.
Security

Classical:

Quantum:
- Hidden-shift algorithms: Subexponential complexity.
  - Literature contains mostly asymptotics.
  - Time-space trade-off: Fastest variants need huge memory.
  - Concrete estimates are to be done.
Security

Classical:

Quantum:
- Hidden-shift algorithms: Subexponential complexity.
  - Literature contains mostly asymptotics.
  - Time-space trade-off: Fastest variants need huge memory.
  - Concrete estimates are to be done.
  - (Recent preprint [BS] ignores much of the cost!)
## Parameters

<table>
<thead>
<tr>
<th>CSIDH-(\log p)</th>
<th>target NIST level</th>
<th>public key size</th>
<th>private key size</th>
<th>time (full exchange)</th>
<th>cycles (full exchange)</th>
<th>stack memory</th>
<th>classical security</th>
<th>quantum security claimed by [BS] (take cum grano salis)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSIDH-512</td>
<td>1</td>
<td>64 b</td>
<td>32 b</td>
<td>85 ms</td>
<td>212e6</td>
<td>4368 b</td>
<td>128</td>
<td>71</td>
</tr>
<tr>
<td>CSIDH-1024</td>
<td>3</td>
<td>128 b</td>
<td>64 b</td>
<td></td>
<td></td>
<td></td>
<td>256</td>
<td>88</td>
</tr>
<tr>
<td>CSIDH-1792</td>
<td>5</td>
<td>224 b</td>
<td>112 b</td>
<td></td>
<td></td>
<td></td>
<td>448</td>
<td>104</td>
</tr>
</tbody>
</table>
Work in progress & future work

- Fast and constant-time implementation
Work in progress & future work

- Fast and constant-time implementation
- Reliable security analysis
Work in progress & future work

- Fast and constant-time implementation
- Reliable security analysis
- More applications
Work in progress & future work

- Fast and constant-time implementation
- Reliable security analysis
- More applications

- [Your paper here!]
Thank you!
References

Mentioned in this talk:

- Castryck, Lange, Martindale, Panny, Renes:  
  *CSIDH: An Efficient Post-Quantum Commutative Group Action*  

- De Feo, Kieffer, Smith:  
  *Towards practical key exchange from ordinary isogeny graphs*  

- De Feo, Galbraith:  
  *SeaSign: Compact isogeny signatures from class group actions*  
  https://ia.cr/2018/824

- [BS] Bonnetain, Schrottenloher:  
  *Quantum Security Analysis of CSIDH and Ordinary Isogeny-based Schemes*  
  https://ia.cr/2018/537

Other related work:

- Delfs, Galbraith:  
  *Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$*  

- Childs, Jao, Soukharev:  
  *Constructing elliptic curve isogenies in quantum subexponential time*  
  https://arxiv.org/abs/1012.4019

- Meyer, Reith:  
  *A faster way to the CSIDH*  

- Jao, LeGrow, Leonardi, Ruiz-Lopez:  
  *A polynomial quantum space attack on CRS and CSIDH*  
  (MathCrypt 2018)

- Biasse, Iezzi, Jacobson:  
  *A note on the security of CSIDH*  
Where’s the group?

\[ E_{51} : y^2 = x^3 + 51x^2 + x \quad \longrightarrow \quad E_9 : y^2 = x^3 + 9x^2 + x \]

\[ (x, y) \quad \mapsto \quad \left( \frac{97x^3 - 183x^2 + x}{x^2 - 183x + 97}, y \cdot \frac{133x^3 + 154x^2 - 5x + 97}{-x^3 + 65x^2 + 128x - 133} \right) \]

\[ E_9 = E_{51}/\alpha \text{ where } \alpha \text{ is the ideal } (3, \pi - 1) \text{ of } \text{End}_{\mathbb{F}_p}(E_{51}). \]
Where’s the group?

$E_{51}: y^2 = x^3 + 51x^2 + x \quad \rightarrow \quad E_9: y^2 = x^3 + 9x^2 + x$

$(x, y) \quad \mapsto \quad \left(\frac{97x^3-183x^2+x}{x^2-183x+97}, y \cdot \frac{133x^3+154x^2-5x+97}{-x^3+65x^2+128x-133}\right)$

$\blacktriangleright \quad E_9 = E_{51}/\mathfrak{a}$ where $\mathfrak{a}$ is the ideal $(3, \pi - 1)$ of $\text{End}_{\mathbb{F}_p}(E_{51})$.

$\blacktriangleright \quad$ For our choices of $A$, $\text{End}_{\mathbb{F}_p}(E_A) \cong \mathbb{Z}[\sqrt{-p}]$.

$\blacktriangleright \quad$ The group action is

$$\text{cl}(\mathbb{Z}[\sqrt{-p}]) \times \{E_A\} \quad \rightarrow \quad \{E_A\}$$

$$([\mathfrak{a}], E) \quad \mapsto \quad E/\mathfrak{a}$$

(modulo details).