Supersingular-isogeny Diffie-Hellman and efficient compression of public keys

Craig Costello[1] David Jao[2] Patrick Longa[1]
Michael Naehrig[1] Joost Renes[3] David Urbanik[2]

j.renes@cs.ru.nl

[1]Microsoft Research, Redmond, USA

[2]University of Waterloo, Ontario, Canada

[3]Radboud University, Nijmegen, The Netherlands

16 December 2016

Outline

- Introduction to SIDH
- Compression of public keys

Feel free to ask questions at any point!

http://eprint.iacr.org/2016/963.pdf

Supersingular-isogeny-based cryptography

- Proposed in [FJP14]
  - Identification
  - Key-exchange (SIDH)
  - Encryption
- Post-quantum secure
- Notably no signatures (yet)
- Recent proposal for public key compression [Aza+16]

The main idea

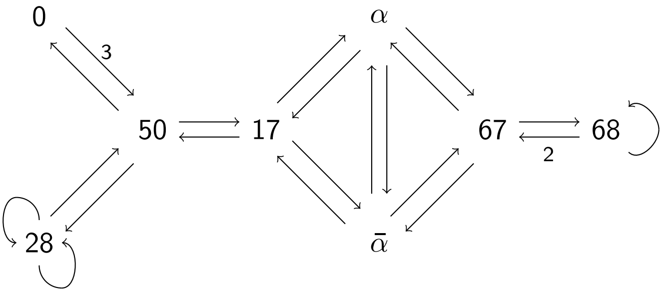
0 α

50 17 67 68

28 $\bar{\alpha}$

$X(\bar{\mathbb{F}}_{83}, 2)$ from [DG16]

The main idea



$X(\bar{\mathbb{F}}_{83}, 2)$ from [DG16]

The main idea

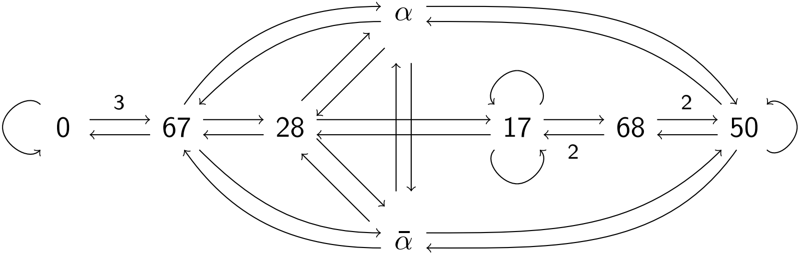
α

0 67 28 17 68 50

$\bar{\alpha}$

$X(\bar{\mathbb{F}}_{83}, 2)$ from [DG16]

The main idea



$X(\bar{\mathbb{F}}_{83}, 2)$ from [DG16]

High-level SIDH (1)

Public information:

- Starting node

High-level SIDH (1)

0 α

50 17 67 68

28 $\bar{\alpha}$

High-level SIDH (1)

0 α

50 17 67 68

28 $\bar{\alpha}$

High-level SIDH (1)

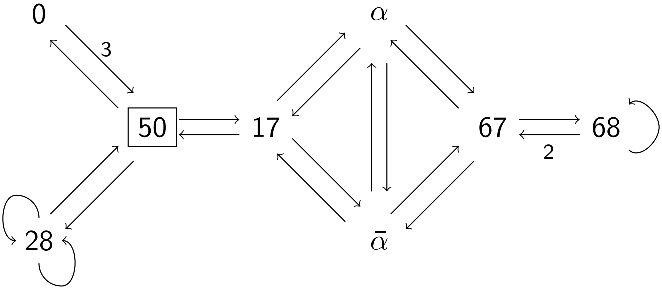
Public information:

- Starting node

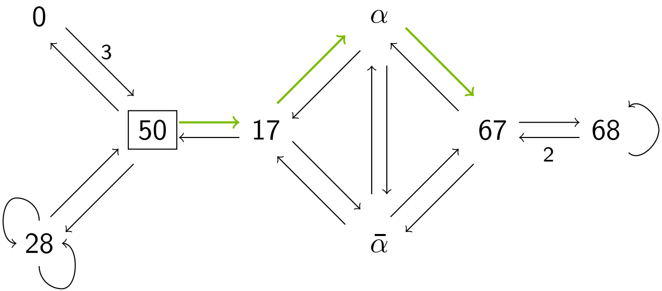
Key generation:

- A chooses secret walk in 2-graph, publishes final node

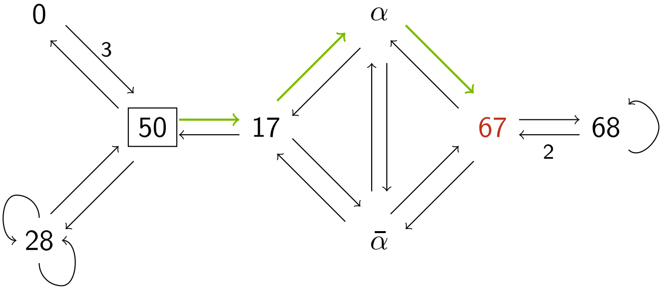
High-level SIDH (1)



High-level SIDH (1)



High-level SIDH (1)



High-level SIDH (1)

Public information:

- Starting node

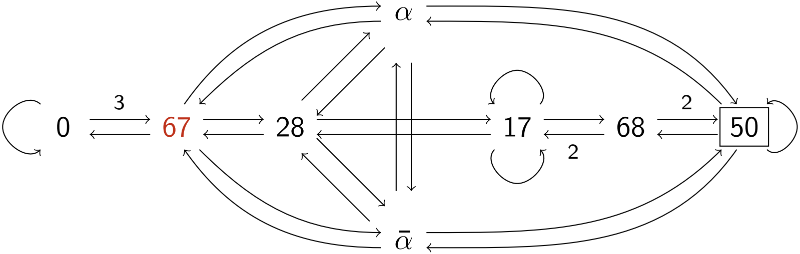
Key generation:

- A chooses secret walk in 2-graph, publishes final node

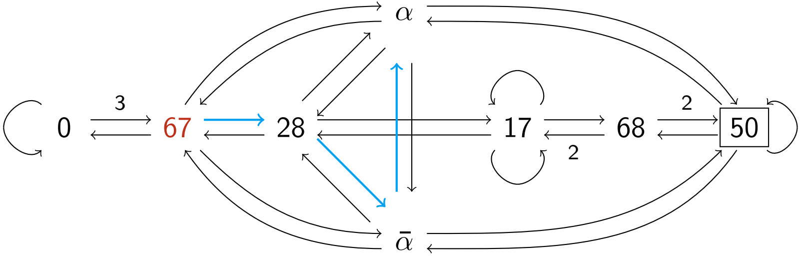
Key exchange:

- B starts a walk in the 3-graph starting at A's final node

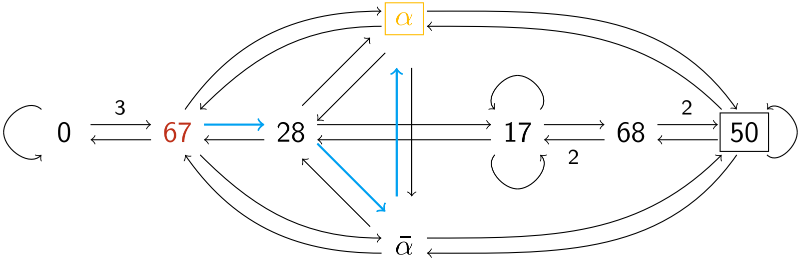
High-level SIDH (1)



High-level SIDH (1)



High-level SIDH (1)



High-level SIDH (2)

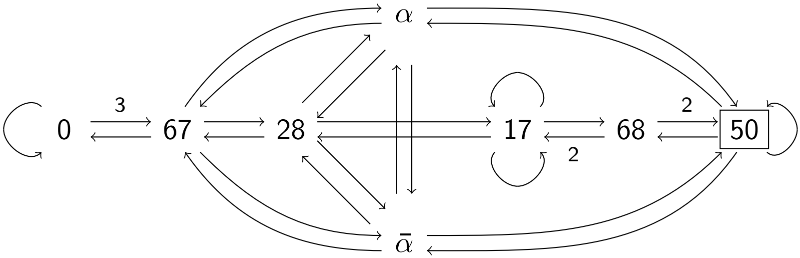
Public information:

- Starting node

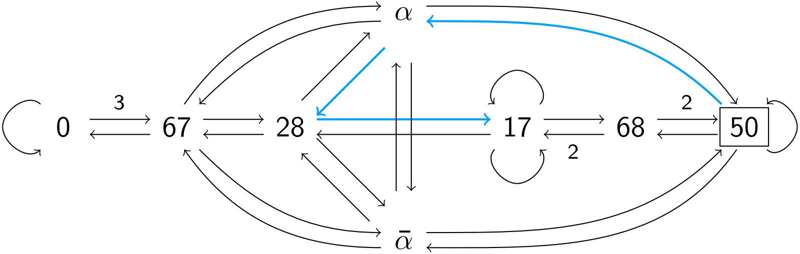
Key generation:

- B chooses secret walk in 3-graph, publishes final node

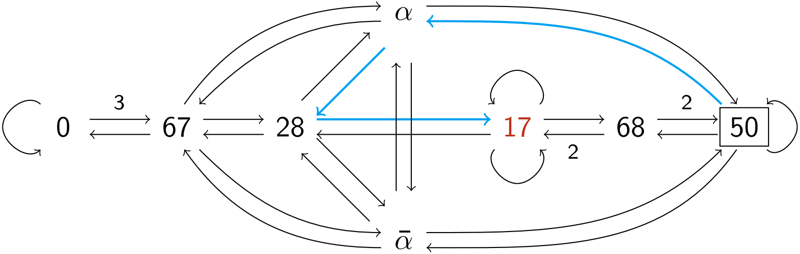
High-level SIDH (2)



High-level SIDH (2)



High-level SIDH (2)



High-level SIDH (2)

Public information:

- Starting node

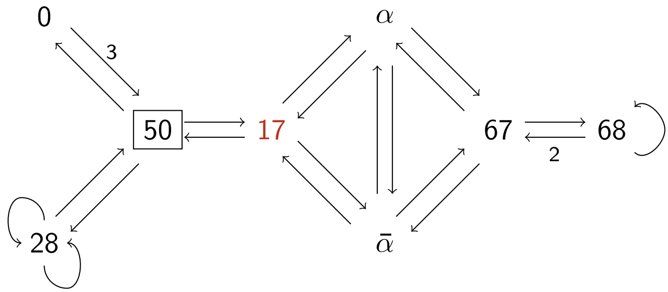
Key generation:

- B chooses secret walk in 3-graph, publishes final node

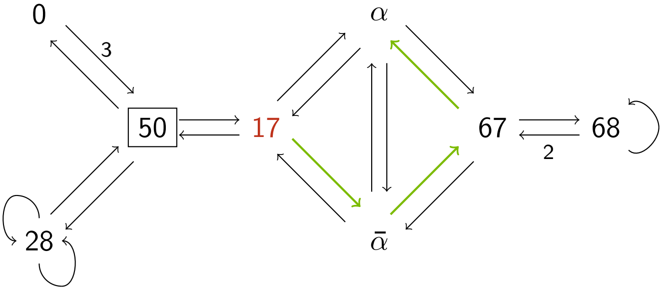
Key exchange:

- A starts a walk in the 2-graph starting at B's final node

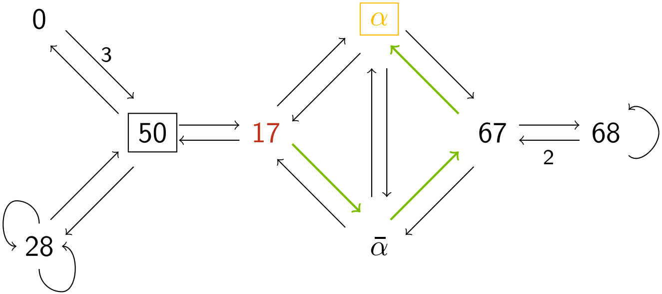
High-level SIDH (2)



High-level SIDH (2)



High-level SIDH (2)



Classic elliptic-curve notation & terminology

$E/\mathbb{F}_q$ = Elliptic curve defined over $\mathbb{F}_q$

$$E/\mathbb{F}_q : y^2 = x^3 + ax + b$$

Classic elliptic-curve notation & terminology

$$E/\mathbb{F}_q = \text{Elliptic curve defined over } \mathbb{F}_q$$
$$\mathcal{O} = \text{Point at infinity}$$

$$\mathcal{O} = (0 : 1 : 0)$$

Classic elliptic-curve notation & terminology

$$\begin{aligned} E/\mathbb{F}_q &= \text{Elliptic curve defined over } \mathbb{F}_q \\ \mathcal{O} &= \text{Point at infinity} \\ \langle P \rangle &= \text{Cyclic subgroup of } E \text{ generated by } P \end{aligned}$$

$$\langle P \rangle = \{[\alpha]\, P \mid \alpha \in \mathbb{Z}\} \subset E$$

Classic elliptic-curve notation & terminology

$$\begin{aligned} E/\mathbb{F}_q &= \text{Elliptic curve defined over } \mathbb{F}_q \\ \mathcal{O} &= \text{Point at infinity} \\ \langle P \rangle &= \text{Cyclic subgroup of } E \text{ generated by } P \\ j(E) &= j\text{-invariant of } E \end{aligned}$$

$$j(E) \in \mathbb{F}_q, \quad j(E) = j(E') \iff E \cong E'$$

More notation & terminology

ϕ = Isogeny $E_1 \to E_2$

A morphism $E_1 \to E_2$ such that

$$\phi(P+Q) = \phi(P) + \phi(Q)$$

for all $P, Q \in E_1$ (in particular $\phi(\mathcal{O}_{E_1}) = \mathcal{O}_{E_2}$)

More notation & terminology

$$\phi = \text{Isogeny } E_1 \to E_2$$
$$\ker \phi = \text{Kernel of } \phi$$

$$\ker \phi = \{P \in E_1 \mid \phi(P) = \mathcal{O}_{E_2}\} \subset E_1$$

More notation & terminology

$$\begin{aligned} \phi &= \text{Isogeny } E_1 \to E_2 \\ \ker\phi &= \text{Kernel of } \phi \\ \deg\phi &= \text{Degree of } \phi \end{aligned}$$

$$\deg\phi \approx \#\ker\phi$$

More notation & terminology

$$\begin{aligned} \phi &= \text{Isogeny } E_1 \to E_2 \\ \ker\phi &= \text{Kernel of } \phi \\ \deg\phi &= \text{Degree of } \phi \\ E/G &= \text{Curve defined by isogeny with kernel } G \end{aligned}$$

Given a subgroup $G \subset E$ there exist a unique isogeny

$$\phi : E \to E/G, \quad \ker\phi = G$$

More notation & terminology

$$
\begin{aligned}
\phi &= \text{Isogeny } E_1 \to E_2 \\
\ker \phi &= \text{Kernel of } \phi \\
\deg \phi &= \text{Degree of } \phi \\
E/G &= \text{Curve defined by isogeny with kernel } G \\
E/\langle P \rangle &= \text{Curve defined by isogeny with kernel } \langle P \rangle
\end{aligned}
$$

Given a point $P \in E$ there exist a unique isogeny

$$\phi : E \to E/\langle P \rangle, \quad \ker \phi = \langle P \rangle$$

More notation & terminology

$$\begin{aligned}
\phi &= \text{Isogeny } E_1 \to E_2 \\
\ker\phi &= \text{Kernel of } \phi \\
\deg\phi &= \text{Degree of } \phi \\
E/G &= \text{Curve defined by isogeny with kernel } G \\
E/\langle P\rangle &= \text{Curve defined by isogeny with kernel } \langle P\rangle \\
E[m] &= m\text{-torsion subgroup of } E
\end{aligned}$$

$$E[m] = \{P \in E \mid [m]\,P = \mathcal{O}\} \cong \mathbb{Z}_m \times \mathbb{Z}_m$$

Even more notation & terminology

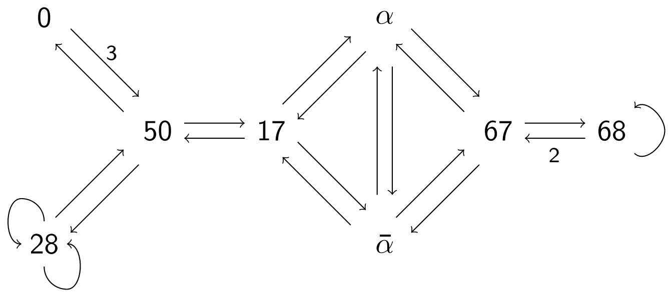
Consider isogeny graphs of supersingular elliptic curves

- The full graph is defined over $\mathbb{F}_{p^2}$
- There are about $p/12$ nodes
- The graph is connected
- In an ℓ-isogeny graph, every node has $\ell + 1$ outgoing edges

Decisional Supersingular Isogeny (DSSI) [FJP14]

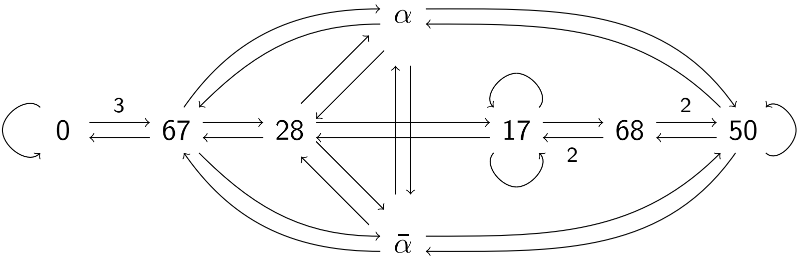
Let E_1 and E_2 be two supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. Let ℓ be a prime such that $\ell^e \mid \#E_1$ for some e. Determine whether E_1 is ℓ^e-isogenous to E_2.

A supersingular 2-isogeny graph



$X(\bar{\mathbb{F}}_{83}, 2)$ from [DG16]

A supersingular 3-isogeny graph



$X(\bar{\mathbb{F}}_{83}, 3)$ from [DG16]

Making things more concrete [CLN16]

- Fix $p = 2^{372}3^{239} - 1$
- Define $E/\mathbb{F}_{p^2} : y^2 = x^3 + x$ which has

$$\#E = (p+1)^2 = (2^{372} \cdot 3^{239})^2$$

- Large 2^{372}-torsion and 3^{239}-torsion subgroups

$$E[2^{372}] \cong \mathbb{Z}_{2^{372}} \times \mathbb{Z}_{2^{372}}, \quad E[3^{239}] \cong \mathbb{Z}_{3^{239}} \times \mathbb{Z}_{3^{239}}$$

Diffie-Hellman in a cyclic group

■ = private party A, ■ = private party B, ■ = public key

Group $G = \langle P \rangle$

P

Diffie-Hellman in a cyclic group

■ = private party A, ■ = private party B, ■ = public key

Group $G = \langle P \rangle$

P

$S = \alpha P$

Diffie-Hellman in a cyclic group

■ = private party A, ■ = private party B, ■ = public key

Group $G = \langle P \rangle$

$$P \xmapsto{\alpha : G \to G} S$$

Diffie-Hellman in a cyclic group

■ = private party A, ■ = private party B, ■ = public key

Group $G = \langle P \rangle$

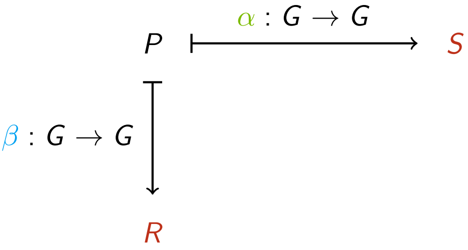
$$P \xmapsto{\alpha : G \to G} S$$

$$R = \beta P$$

Diffie-Hellman in a cyclic group

■ = private party A, ■ = private party B, ■ = public key

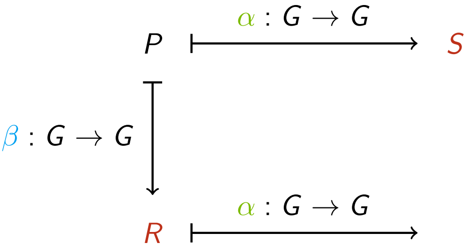
Group $G = \langle P \rangle$



Diffie-Hellman in a cyclic group

= private party A, = private party B, = public key

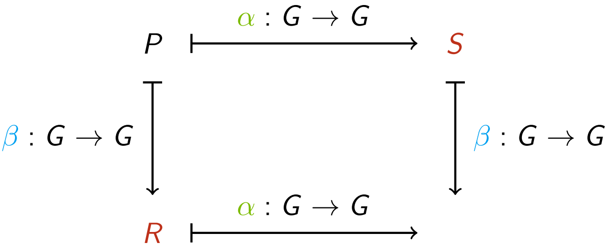
Group $G = \langle P \rangle$



Diffie-Hellman in a cyclic group

■ = private party A, ■ = private party B, ■ = public key

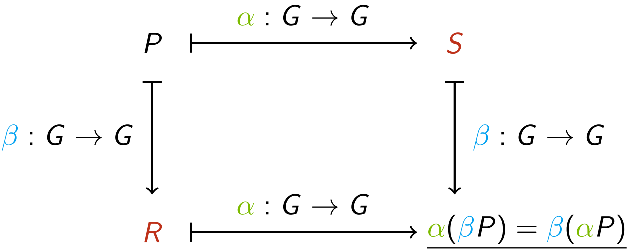
Group $G = \langle P \rangle$



Diffie-Hellman in a cyclic group

■ = private party A, ■ = private party B, ■ = public key

Group $G = \langle P \rangle$



Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

■ = private party A, ■ = private party B, ■ = public key

E

Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

■ = private party A, ■ = private party B, ■ = public key

E $S \in E[2^{372}]$

Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

■ = private party A, ■ = private party B, ■ = public key

$$E \xmapsto[S]{\phi_A} E/\langle S\rangle$$

Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

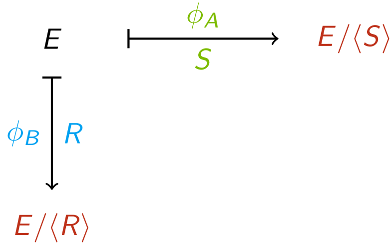
■ = private party A, ■ = private party B, ■ = public key

$$E \xmapsto[S]{\phi_A} E/\langle S \rangle$$

$R \in E[3^{239}]$

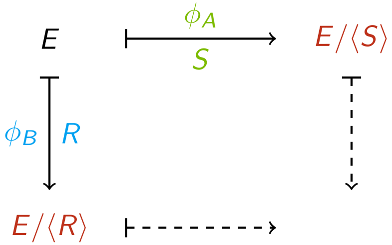
Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

■ = private party A, ■ = private party B, ■ = public key



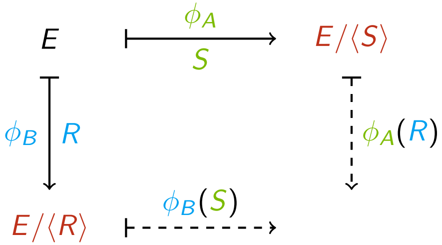
Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

■ = private party A, ■ = private party B, ■ = public key



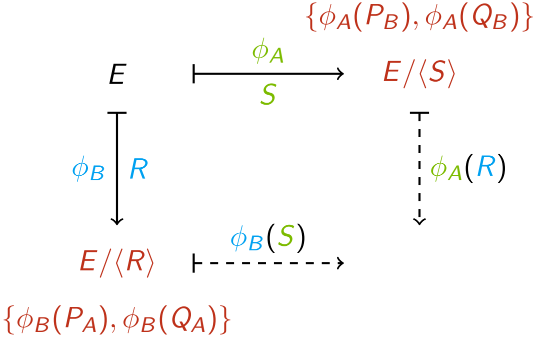
Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

■ = private party A, ■ = private party B, ■ = public key



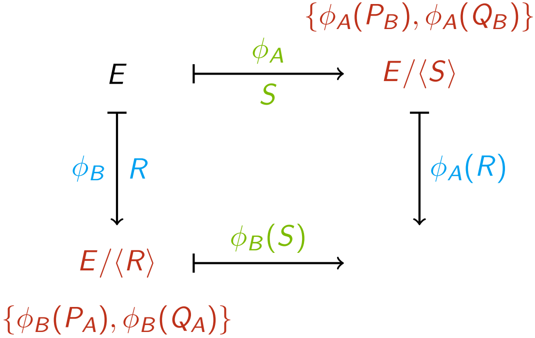
Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

■ = private party A, ■ = private party B, ■ = public key



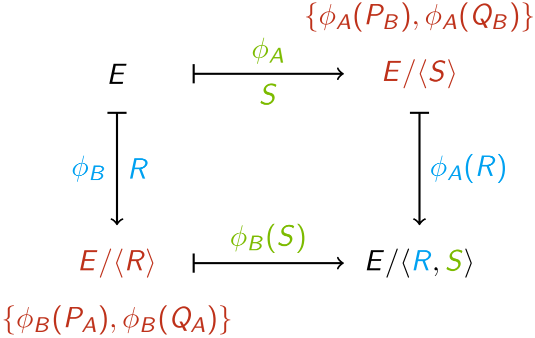
Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

■ = private party A, ■ = private party B, ■ = public key



Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

■ = private party A, ■ = private party B, ■ = public key



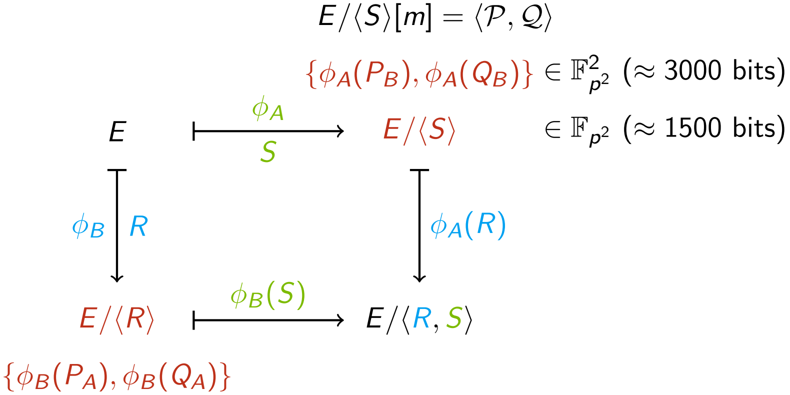
Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

= private party A, = private party B, = public key

$$
\begin{array}{ccccc}
 & & \{\phi_A(P_B), \phi_A(Q_B)\} & \in \mathbb{F}_{p^2}^2 & (\approx 3000 \text{ bits}) \\
E & \xmapsto[S]{\phi_A} & E/\langle S \rangle & \in \mathbb{F}_{p^2} & (\approx 1500 \text{ bits}) \\
\phi_B \Big\downarrow R & & \Big\downarrow \phi_A(R) & & \\
E/\langle R \rangle & \xmapsto{\phi_B(S)} & E/\langle R, S \rangle & & \\
\{\phi_B(P_A), \phi_B(Q_A)\} & & & &
\end{array}
$$

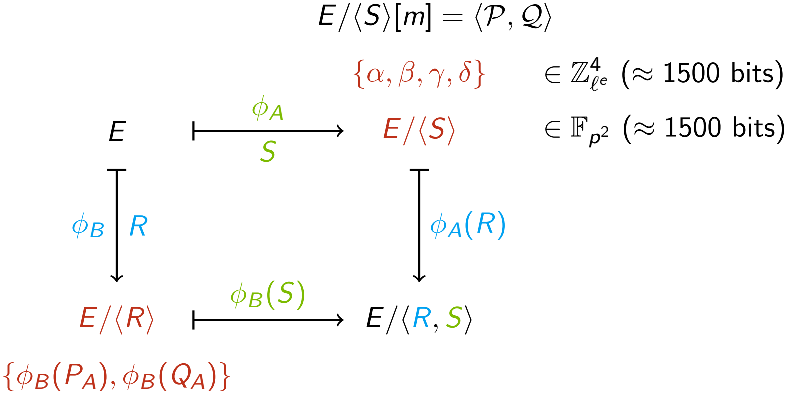
Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

(green) = private party A, (blue) = private party B, (red) = public key



Supersingular-isogeny Diffie-Hellman [FJP14; Aza+16]

■ = private party A, ■ = private party B, ■ = public key



Contributions

1. Further compress from $\mathbb{F}_{p^2} \times \mathbb{Z}_{\ell^e}^4$ to $\mathbb{F}_{p^2} \times \mathbb{Z}_{\ell^e}^3 \times \mathbb{Z}_2$
2. Speed up generation of ℓ^e-torsion basis
3. Speed up discrete logarithm computation
   - Use efficient parallel Tate pairing computation

$$E(\mathbb{F}_{p^2})[\ell^e] \times E(\mathbb{F}_{p^2})/\ell^e E(\mathbb{F}_{p^2}) \to \mu_{\ell^e}$$

   - Compute fast discrete logarithms in μ_{ℓ^e}
4. Speed up decompression

Benchmark: Key size reduced by 12.5%. Compression up to 57 times faster, decompression up to 17 times faster.

Improved compression

Given public key $(j(E), \alpha, \beta, \gamma, \delta)$ the shared secret is

$$\mathtt{K} = E/\langle \alpha R + \beta S + \lambda (\gamma R + \delta S) \rangle$$

Either α or β is invertible (wlog assume α), and thus we compute

$$\mathtt{K} = E/\langle R + \alpha^{-1}\beta S + \lambda \left(\alpha^{-1}\gamma R + \alpha^{-1}\delta S\right) \rangle$$

and set the public key to

$$\begin{cases} \left(j(E), \alpha^{-1}\beta, \alpha^{-1}\gamma, \alpha^{-1}\delta, 0\right) & \text{if } \alpha \in \mathbb{Z}_{\ell^e}^* \\ \left(j(E), \beta^{-1}\alpha, \beta^{-1}\gamma, \beta^{-1}\delta, 1\right) & \text{if } \beta \in \mathbb{Z}_{\ell^e}^* \end{cases}$$

Improved ℓ^e-torsion basis computation

Given an elliptic curve E such that $\#E = 2^{372} \cdot 3^{239}$, find

1. $P_A, Q_A \in E$ such that $E[2^{372}] = \langle P_A, Q_A \rangle$
2. $P_B, Q_B \in E$ such that $E[3^{239}] = \langle P_B, Q_B \rangle$

Improved ℓ^e-torsion basis computation

Given an elliptic curve E such that $\#E = 2^{372} \cdot 3^{239}$, find

1. $P_A, Q_A \in E$ such that $E[2^{372}] = \langle P_A, Q_A \rangle$
2. $P_B, Q_B \in E$ such that $E[3^{239}] = \langle P_B, Q_B \rangle$

Naïve approach:

1. Choose $P \in E$ until $[3^{239}]P \in E[2^{372}] \setminus E[2^{371}]$
2. Choose $Q \in E$ until $[3^{239}]Q \in E[2^{372}] \setminus E[2^{371}]$
3. If $E[2^{372}] \neq \langle [3^{239}]P, [3^{239}]Q \rangle$, go back to step 2
4. Choose $P \in E$ until $[2^{372}]P \in E[3^{239}] \setminus E[3^{238}]$
5. Choose $Q \in E$ until $[2^{372}]Q \in E[3^{239}] \setminus E[3^{238}]$
6. If $E[3^{239}] \neq \langle [2^{372}]P, [2^{372}]Q \rangle$, go back to step 5

Improved ℓ^e-torsion basis computation

Given an elliptic curve E such that $\#E = 2^{372} \cdot 3^{239}$, find

1. $P_A, Q_A \in E$ such that $E[2^{372}] = \langle P_A, Q_A \rangle$
2. $P_B, Q_B \in E$ such that $E[3^{239}] = \langle P_B, Q_B \rangle$

Improvements:

1. Choose $P \in E$ until $[3^{239}]P \in E[2^{372}] \setminus E[2^{371}]$
2. Choose $Q \in E$ until $[3^{239}]Q \in E[2^{372}] \setminus E[2^{371}]$

Improved ℓ^e-torsion basis computation

Given an elliptic curve E such that $\#E = 2^{372} \cdot 3^{239}$, find

1. $P_A, Q_A \in E$ such that $E[2^{372}] = \langle P_A, Q_A \rangle$
2. $P_B, Q_B \in E$ such that $E[3^{239}] = \langle P_B, Q_B \rangle$

Improvements:

1. Choose $P \in E$ until $[2^{371}]\left([3^{239}]P\right) \neq \mathcal{O}$
2. Choose $Q \in E$ until $[3^{239}]Q \in E[2^{372}] \setminus E[2^{371}]$

Improved ℓ^e-torsion basis computation

Given an elliptic curve E such that $\#E = 2^{372} \cdot 3^{239}$, find

1. $P_A, Q_A \in E$ such that $E[2^{372}] = \langle P_A, Q_A \rangle$
2. $P_B, Q_B \in E$ such that $E[3^{239}] = \langle P_B, Q_B \rangle$

Improvements:

1. Choose $P \in E$ until $[2^{371}]\left([3^{239}]P\right) \neq \mathcal{O}$
2. Choose $Q \in E$ until $[3^{239}]Q \in E[2^{372}] \setminus E[2^{371}]$

$$\forall R \in E, x(R) \text{ is not a square} \implies [2^{371}]R \neq \mathcal{O}$$

Improved ℓ^e-torsion basis computation

Given an elliptic curve E such that $\#E = 2^{372} \cdot 3^{239}$, find

1. $P_A, Q_A \in E$ such that $E[2^{372}] = \langle P_A, Q_A \rangle$
2. $P_B, Q_B \in E$ such that $E[3^{239}] = \langle P_B, Q_B \rangle$

Improvements:

1. Choose $P \in E$ until $x([3^{239}]P)$ is not a square
2. Choose $Q \in E$ until $[3^{239}]Q \in E[2^{372}] \setminus E[2^{371}]$

$$\forall R \in E, x(R) \text{ is not a square} \implies [2^{371}]R \neq \mathcal{O}$$

Improved ℓ^e-torsion basis computation

Given an elliptic curve E such that $\#E = 2^{372} \cdot 3^{239}$, find

1. $P_A, Q_A \in E$ such that $E[2^{372}] = \langle P_A, Q_A \rangle$
2. $P_B, Q_B \in E$ such that $E[3^{239}] = \langle P_B, Q_B \rangle$

Improvements:

1. Choose non-squares $x \in \mathbb{F}_{p^2}$ until $x^3 + Ax^2 + x$ is a square
2. Set $P = [3^{239}](x, \sqrt{x^3 + Ax^2 + x})$
3. Choose $Q \in E$ until $[3^{239}]Q \in E[2^{372}] \setminus E[2^{371}]$

Improved ℓ^e-torsion basis computation

Given an elliptic curve E such that $\#E = 2^{372} \cdot 3^{239}$, find

1. $P_A, Q_A \in E$ such that $E[2^{372}] = \langle P_A, Q_A \rangle$
2. $P_B, Q_B \in E$ such that $E[3^{239}] = \langle P_B, Q_B \rangle$

Improvements:

1. Choose non-squares $x \in \mathbb{F}_{p^2}$ until $x^3 + Ax^2 + x$ is a square
2. Set $P = [3^{239}](x, \sqrt{x^3 + Ax^2 + x})$
3. Choose non-squares $z \in \mathbb{F}_{p^2}$ until $z^3 + Az^2 + z$ is a square
4. Set $Q = [3^{239}](z, \sqrt{z^3 + Az^2 + z})$

Improved ℓ^e-torsion basis computation

Given an elliptic curve E such that $\#E = 2^{372} \cdot 3^{239}$, find

1. $P_A, Q_A \in E$ such that $E[2^{372}] = \langle P_A, Q_A \rangle$
2. $P_B, Q_B \in E$ such that $E[3^{239}] = \langle P_B, Q_B \rangle$

Improvements:

1. Choose non-squares $x \in \mathbb{F}_{p^2}$ until $x^3 + Ax^2 + x$ is a square
2. Set $P = [3^{239}](x, \sqrt{x^3 + Ax^2 + x})$
3. Choose non-squares $z \in \mathbb{F}_{p^2}$ until $z^3 + Az^2 + z$ is a square
4. Set $Q = [3^{239}](z, \sqrt{z^3 + Az^2 + z})$
5. If $E[2^{372}] \neq \langle P, Q \rangle$, go back to step 3

Improved ℓ^e-torsion basis computation

Given an elliptic curve E such that $\#E = 2^{372} \cdot 3^{239}$, find

1. $P_A, Q_A \in E$ such that $E[2^{372}] = \langle P_A, Q_A \rangle$
2. $P_B, Q_B \in E$ such that $E[3^{239}] = \langle P_B, Q_B \rangle$

Improvements:

1. Choose non-squares $x \in \mathbb{F}_{p^2}$ until $x^3 + Ax^2 + x$ is a square
2. Set $P = [3^{239}](x, \sqrt{x^3 + Ax^2 + x})$
3. Choose non-squares $z \in \mathbb{F}_{p^2}$ until $z^3 + Az^2 + z$ is a square
4. Set $Q = [3^{239}](z, \sqrt{z^3 + Az^2 + z})$
5. If $E[2^{372}] \neq \langle P, Q \rangle$, go back to step 3
6. For 3^{239}-torsion basis do similar (bit more involved) tricks

Efficient computation of Tate pairings

$$e_0 = e\left(R_1, R_2\right) = f_{n,R_1}(R_2)^{(p^2-1)/n}$$
$$e_1 = e\left(R_1, P\right) = f_{n,R_1}(P)^{(p^2-1)/n}$$
$$e_2 = e\left(R_1, Q\right) = f_{n,R_1}(Q)^{(p^2-1)/n}$$
$$e_3 = e\left(R_2, P\right) = f_{n,R_2}(P)^{(p^2-1)/n}$$
$$e_4 = e\left(R_2, Q\right) = f_{n,R_2}(Q)^{(p^2-1)/n}$$

Efficient computation of Tate pairings

$$
\begin{aligned}
e_0 &= e(R_1, R_2) = f_{n,R_1}(R_2)^{(p^2-1)/n} \\
e_1 &= e(R_1, P) = f_{n,R_1}(P)^{(p^2-1)/n} \\
e_2 &= e(R_1, Q) = f_{n,R_1}(Q)^{(p^2-1)/n} \\
e_3 &= e(R_2, P) = f_{n,R_2}(P)^{(p^2-1)/n} \\
e_4 &= e(R_2, Q) = f_{n,R_2}(Q)^{(p^2-1)/n}
\end{aligned}
$$

Miller's loop [Are+09]

$S_1 \leftarrow R_1,\ S_2 \leftarrow R_1,\ S_3 \leftarrow R_1,\ S_4 \leftarrow R_2,\ S_5 \leftarrow R_2,\ f_i \leftarrow 1$

for $i = n - 1$ to 0 **do**

  Compute $g_{S_1,S_1}, g_{S_2,S_2}, g_{S_3,S_3}, g_{S_4,S_4}, g_{S_5,S_5}$

  $f_1 \leftarrow f_1^2 \cdot g_{S_1,S_1}(R_2),\ S_1 \leftarrow [2]S_1$

  $f_2 \leftarrow f_2^2 \cdot g_{S_2,S_2}(P),\ S_2 \leftarrow [2]S_2$

  $f_3 \leftarrow f_3^2 \cdot g_{S_3,S_3}(Q),\ S_3 \leftarrow [2]S_3$

  $f_4 \leftarrow f_4^2 \cdot g_{S_4,S_4}(P),\ S_4 \leftarrow [2]S_4$

  $f_5 \leftarrow f_5^2 \cdot g_{S_5,S_5}(Q),\ S_5 \leftarrow [2]S_5$

  if $n_i = 1$ **then**

    Compute $g_{S_1,R_1}, g_{S_2,R_1}, g_{S_3,R_1}, g_{S_4,R_2}, g_{S_5,R_2}$

    $f_1 \leftarrow f_1 \cdot g_{S_1,R_1}(R_2),\ S_1 \leftarrow S_1 + R_1$

    $f_2 \leftarrow f_2 \cdot g_{S_2,R_1}(P),\ S_2 \leftarrow S_2 + R_1$

    $f_3 \leftarrow f_3 \cdot g_{S_3,R_1}(Q),\ S_3 \leftarrow S_3 + R_1$

    $f_4 \leftarrow f_4 \cdot g_{S_4,R_2}(P),\ S_4 \leftarrow S_4 + R_2$

    $f_5 \leftarrow f_5 \cdot g_{S_5,R_2}(Q),\ S_5 \leftarrow S_5 + R_2$

  end if

end for

Miller's loop [Are+09]

$S_1 \leftarrow R_1,\ S_2 \leftarrow R_1,\ S_3 \leftarrow R_1,\ S_4 \leftarrow R_2,\ S_5 \leftarrow R_2,\ f_i \leftarrow 1$

for $i = n-1$ to 0 **do**

  Compute $g_{S_1,S_1}, g_{S_2,S_2}, g_{S_3,S_3}, g_{S_4,S_4}, g_{S_5,S_5}$

  $f_1 \leftarrow f_1^2 \cdot g_{S_1,S_1}(R_2),\ S_1 \leftarrow [2]S_1$

  $f_2 \leftarrow f_2^2 \cdot g_{S_2,S_2}(P),\ S_2 \leftarrow [2]S_2$

  $f_3 \leftarrow f_3^2 \cdot g_{S_3,S_3}(Q),\ S_3 \leftarrow [2]S_3$

  $f_4 \leftarrow f_4^2 \cdot g_{S_4,S_4}(P),\ S_4 \leftarrow [2]S_4$

  $f_5 \leftarrow f_5^2 \cdot g_{S_5,S_5}(Q),\ S_5 \leftarrow [2]S_5$

  if $n_i = 1$ **then**

    Compute $g_{S_1,R_1}, g_{S_2,R_1}, g_{S_3,R_1}, g_{S_4,R_2}, g_{S_5,R_2}$

    $f_1 \leftarrow f_1 \cdot g_{S_1,R_1}(R_2),\ S_1 \leftarrow S_1 + R_1$

    $f_2 \leftarrow f_2 \cdot g_{S_2,R_1}(P),\ S_2 \leftarrow S_2 + R_1$

    $f_3 \leftarrow f_3 \cdot g_{S_3,R_1}(Q),\ S_3 \leftarrow S_3 + R_1$

    $f_4 \leftarrow f_4 \cdot g_{S_4,R_2}(P),\ S_4 \leftarrow S_4 + R_2$

    $f_5 \leftarrow f_5 \cdot g_{S_5,R_2}(Q),\ S_5 \leftarrow S_5 + R_2$

  end if

end for

Miller's loop [Are+09]

```
S ← R_1, S_4 ← R_2, S_5 ← R_2, f_i ← 1
for i = n − 1 to 0 do
    Compute g_{S,S}, g_{S,S}, g_{S,S}, g_{S_4,S_4}, g_{S_5,S_5}
    f_1 ← f_1^2 · g_{S,S}(R_2), S_1 ← [2]S_1
    f_2 ← f_2^2 · g_{S,S}(P), S_2 ← [2]S_2
    f_3 ← f_3^2 · g_{S,S}(Q), S_3 ← [2]S_3
    f_4 ← f_4^2 · g_{S_4,S_4}(P), S_4 ← [2]S_4
    f_5 ← f_5^2 · g_{S_5,S_5}(Q), S_5 ← [2]S_5
    if n_i = 1 then
        Compute g_{S,R_1}, g_{S,R_1}, g_{S,R_1}, g_{S_4,R_2}, g_{S_5,R_2}
        f_1 ← f_1 · g_{S,R_1}(R_2), S_1 ← S_1 + R_1
        f_2 ← f_2 · g_{S,R_1}(P), S_2 ← S_2 + R_1
        f_3 ← f_3 · g_{S,R_1}(Q), S_3 ← S_3 + R_1
        f_4 ← f_4 · g_{S_4,R_2}(P), S_4 ← S_4 + R_2
        f_5 ← f_5 · g_{S_5,R_2}(Q), S_5 ← S_5 + R_2
    end if
end for
```

Miller's loop [Are+09]

- $S \leftarrow R_1,\ S_4 \leftarrow R_2,\ S_5 \leftarrow R_2,\ f_i \leftarrow 1$
- **for** $i = n-1$ to 0 **do**
  - Compute $g_{S,S},\ g_{S_4,S_4},\ g_{S_5,S_5}$
  - $f_1 \leftarrow f_1^2 \cdot g_{S,S}(R_2),\ S_1 \leftarrow [2]S_1$
  - $f_2 \leftarrow f_2^2 \cdot g_{S,S}(P),\ S_2 \leftarrow [2]S_2$
  - $f_3 \leftarrow f_3^2 \cdot g_{S,S}(Q),\ S_3 \leftarrow [2]S_3$
  - $f_4 \leftarrow f_4^2 \cdot g_{S_4,S_4}(P),\ S_4 \leftarrow [2]S_4$
  - $f_5 \leftarrow f_5^2 \cdot g_{S_5,S_5}(Q),\ S_5 \leftarrow [2]S_5$
  - **if** $n_i = 1$ **then**
    - Compute $g_{S,R_1},\ g_{S_4,R_2},\ g_{S_5,R_2}$
    - $f_1 \leftarrow f_1 \cdot g_{S,R_1}(R_2),\ S_1 \leftarrow S_1 + R_1$
    - $f_2 \leftarrow f_2 \cdot g_{S,R_1}(P),\ S_2 \leftarrow S_2 + R_1$
    - $f_3 \leftarrow f_3 \cdot g_{S,R_1}(Q),\ S_3 \leftarrow S_3 + R_1$
    - $f_4 \leftarrow f_4 \cdot g_{S_4,R_2}(P),\ S_4 \leftarrow S_4 + R_2$
    - $f_5 \leftarrow f_5 \cdot g_{S_5,R_2}(Q),\ S_5 \leftarrow S_5 + R_2$
  - **end if**
- **end for**

Miller's loop [Are+09]

$S \leftarrow R_1$, $S_4 \leftarrow R_2$, $S_5 \leftarrow R_2$, $f_i \leftarrow 1$
for $i = n - 1$ to 0 **do**
  Compute $g_{S,S}$, g_{S_4,S_4}, g_{S_5,S_5}
  $f_1 \leftarrow f_1^2 \cdot g_{S,S}(R_2)$, $S_1 \leftarrow [2]S_1$
  $f_2 \leftarrow f_2^2 \cdot g_{S,S}(P\ )$, $S_2 \leftarrow [2]S_2$
  $f_3 \leftarrow f_3^2 \cdot g_{S,S}(Q\ )$, $S_3 \leftarrow [2]S_3$
  $f_4 \leftarrow f_4^2 \cdot g_{S_4,S_4}(P\ )$, $S_4 \leftarrow [2]S_4$
  $f_5 \leftarrow f_5^2 \cdot g_{S_5,S_5}(Q\ )$, $S_5 \leftarrow [2]S_5$
  if $n_i = 1$ **then**
    Compute g_{S,R_1}, g_{S_4,R_2}, g_{S_5,R_2}
    $f_1 \leftarrow f_1 \cdot g_{S,R_1}(R_2)$, $S_1 \leftarrow S_1 + R_1$
    $f_2 \leftarrow f_2 \cdot g_{S,R_1}(P\ )$, $S_2 \leftarrow S_2 + R_1$
    $f_3 \leftarrow f_3 \cdot g_{S,R_1}(Q\ )$, $S_3 \leftarrow S_3 + R_1$
    $f_4 \leftarrow f_4 \cdot g_{S_4,R_2}(P\ )$, $S_4 \leftarrow S_4 + R_2$
    $f_5 \leftarrow f_5 \cdot g_{S_5,R_2}(Q\ )$, $S_5 \leftarrow S_5 + R_2$
  end if
end for

Miller's loop [Are+09]

$S \leftarrow R_1$, $S_4 \leftarrow R_2$, $S_5 \leftarrow R_2$, $f_i \leftarrow 1$
for $i = n - 1$ to 0 **do**
  Compute $g_{S,S}$, g_{S_4,S_4}, g_{S_5,S_5}
  $f_1 \leftarrow f_1^2 \cdot g_{S,S}(R_2)$,
  $f_2 \leftarrow f_2^2 \cdot g_{S,S}(P\ )$,
  $f_3 \leftarrow f_3^2 \cdot g_{S,S}(Q\ )$, $S \leftarrow [2]S$
  $f_4 \leftarrow f_4^2 \cdot g_{S_4,S_4}(P\ )$, $S_4 \leftarrow [2]S_4$
  $f_5 \leftarrow f_5^2 \cdot g_{S_5,S_5}(Q\ )$, $S_5 \leftarrow [2]S_5$
  if $n_i = 1$ **then**
    Compute g_{S,R_1}, g_{S_4,R_2}, g_{S_5,R_2}
    $f_1 \leftarrow f_1 \cdot g_{S,R_1}(R_2)$,
    $f_2 \leftarrow f_2 \cdot g_{S,R_1}(P\ )$,
    $f_3 \leftarrow f_3 \cdot g_{S,R_1}(Q\ )$, $S \leftarrow S + R_1$
    $f_4 \leftarrow f_4 \cdot g_{S_4,R_2}(P\ )$, $S_4 \leftarrow S_4 + R_2$
    $f_5 \leftarrow f_5 \cdot g_{S_5,R_2}(Q\ )$, $S_5 \leftarrow S_5 + R_2$
  end if
end for

Miller's loop [Are+09]

$S \leftarrow R_1$, $S_4 \leftarrow R_2$, $S_5 \leftarrow R_2$, $f_i \leftarrow 1$
for $i = n - 1$ to 0 **do**
    Compute $g_{S,S}$, g_{S_4,S_4}, g_{S_5,S_5}
    $f_1 \leftarrow f_1^2 \cdot g_{S,S}(R_2)$,
    $f_2 \leftarrow f_2^2 \cdot g_{S,S}(P)$,
    $f_3 \leftarrow f_3^2 \cdot g_{S,S}(Q)$, $S \leftarrow [2]S$
    $f_4 \leftarrow f_4^2 \cdot g_{S_4,S_4}(P)$, $S_4 \leftarrow [2]S_4$
    $f_5 \leftarrow f_5^2 \cdot g_{S_5,S_5}(Q)$, $S_5 \leftarrow [2]S_5$
    if $n_i = 1$ **then**
        Compute g_{S,R_1}, g_{S_4,R_2}, g_{S_5,R_2}
        $f_1 \leftarrow f_1 \cdot g_{S,R_1}(R_2)$,
        $f_2 \leftarrow f_2 \cdot g_{S,R_1}(P)$,
        $f_3 \leftarrow f_3 \cdot g_{S,R_1}(Q)$, $S \leftarrow S + R_1$
        $f_4 \leftarrow f_4 \cdot g_{S_4,R_2}(P)$, $S_4 \leftarrow S_4 + R_2$
        $f_5 \leftarrow f_5 \cdot g_{S_5,R_2}(Q)$, $S_5 \leftarrow S_5 + R_2$
    end if
end for

Miller's loop [Are+09]

$S \leftarrow R_1,\ T \leftarrow R_2,\ f_i \leftarrow 1$

for $i = n-1$ to 0 **do**

  Compute $g_{S,S},\ g_{T,T}$,

  $f_1 \leftarrow f_1^2 \cdot g_{S,S}(R_2)$,

  $f_2 \leftarrow f_2^2 \cdot g_{S,S}(P\ )$,

  $f_3 \leftarrow f_3^2 \cdot g_{S,S}(Q\ ),\ S \leftarrow [2]S$

  $f_4 \leftarrow f_4^2 \cdot g_{T,T}(P\ )$,

  $f_5 \leftarrow f_5^2 \cdot g_{T,T}(Q\ ),\ T \leftarrow [2]T$

  if $n_i = 1$ **then**

    Compute $g_{S,R_1},\ g_{T,R_2}$,

    $f_1 \leftarrow f_1 \cdot g_{S,R_1}(R_2)$,

    $f_2 \leftarrow f_2 \cdot g_{S,R_1}(P\ )$,

    $f_3 \leftarrow f_3 \cdot g_{S,R_1}(Q\ ),\ S \leftarrow S + R_1$

    $f_4 \leftarrow f_4 \cdot g_{T,R_2}(P\ )$,

    $f_5 \leftarrow f_5 \cdot g_{T,R_2}(Q\ ),\ T \leftarrow T + R_2$

  end if

end for

Easy and hard exponentiation

$$f_i \leftarrow f_i^{(p^2-1)/n}$$

Easy and hard exponentiation

$$f_i \leftarrow f_i^{p-1} = \frac{f_i^p}{f_i} \text{ (easy)}$$

$$f_i \leftarrow f_i^{(p+1)/n} \text{ (hard)}$$

Easy and hard exponentiation

$$f_i \leftarrow f_i^{p-1} = \frac{f_i^p}{f_i} \text{ (easy)}$$

$$f_i \leftarrow f_i^{(p+1)/n} \text{ (hard)}$$

Use optimized arithmetic in cyclotomic subgroup:

$$f_i \in G_{p+1} \subset \mathbb{F}_{p^2}$$

$$\mathbf{I} \approx \mathbf{M}, \quad \mathbf{S} \approx 2\mathbf{s}, \quad \mathbf{C} \approx 2\mathbf{m} + 1\mathbf{s}$$

Efficient Pohlig-Hellman in μ_{ℓ^e}

<u>The problem</u>:

- Given a group $\langle g \rangle \cong \mu_{\ell^e}$
- Given $r \in \langle g \rangle$
- Compute α such that $r = g^\alpha$

Efficient Pohlig-Hellman in μ_{ℓ^e}

The problem:

- Given a group $\langle g \rangle \cong \mu_{\ell^e}$
- Given $r \in \langle g \rangle$
- Compute α such that $r = g^\alpha$

Note that

$$\langle g \rangle \cong \mu_{\ell^e} \subset G_{p+1} \subset \mathbb{F}_{p^2},$$

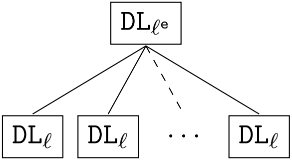
so again

$$\mathbf{I} \approx \mathbf{M}, \quad \mathbf{S} \approx 2\mathbf{s}, \quad \mathbf{C} \approx 2\mathbf{m} + 1\mathbf{s}$$

Pohlig-Hellman

$\# G_1 = \ell^e$

$\# G_2 = \ell$



Nested Pohlig-Hellman

$\#G_1 = \ell^{e_1}$

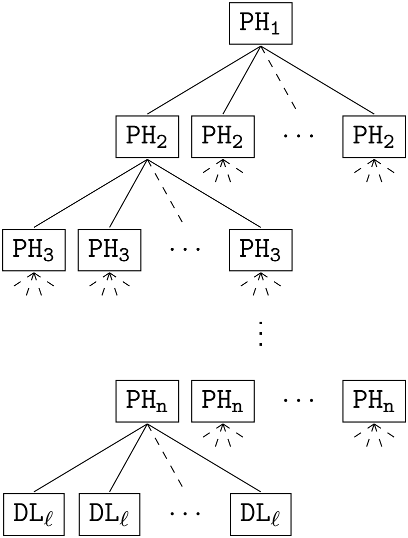
$\#G_2 = \ell^{e_2}$

$\#G_3 = \ell^{e_3}$

$\vdots$

$\#G_n = \ell^{e_n}$

$\#G_{n+1} = \ell$



Nested Pohlig-Hellman

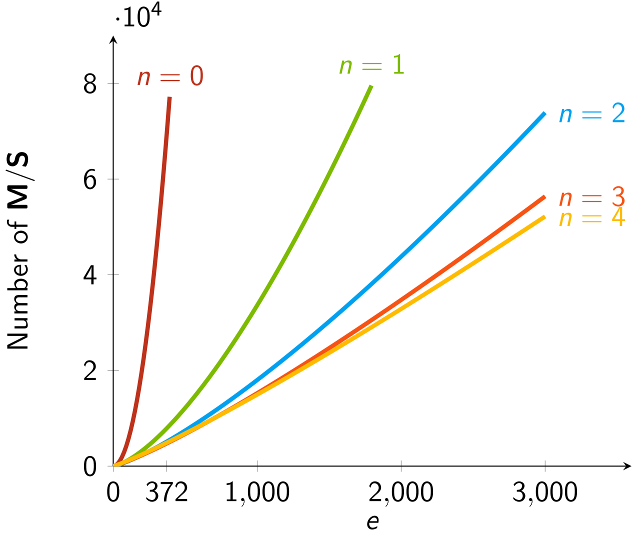
- Turns out the optimal choices for e_i are

$$(e_1, \ldots, e_{n+1}) = \left(e, e^{\frac{n}{n+1}}, e^{\frac{n-1}{n+1}} \ldots, 1\right)$$

- Assuming $\log \ell \approx 1$ we have complexity

$$f_n(e) \approx \frac{1}{2}(n+1)e \cdot e^{\frac{1}{n+1}} + (n+1)e$$

Nested Pohlig-Hellman



Comparison in Magma implementation ($\ell = 2$)

#	windows				$\mathbb{F}_{p^2}$		table size
n	w_1	w_2	w_3	w_4	**M**	**S**	$\mathbb{F}_{p^2}$
0	–	–	–	–	372	69 378	375
1	19	–	–	–	375	7 445	43
2	51	7	–	–	643	4 437	25
3	84	21	5	–	716	3 826	25
4	114	35	11	3	1 065	3 917	27

Mixing key exchange and decompression

Key exchange w.r.t. public key $(j(E), \beta, \gamma, \delta, 0)$:

1. Compute the basis $\{R, S\}$
2. Decompress $P = R + \beta S$ and $Q = \gamma R + \delta S$
3. Compute $P + \lambda Q$
4. Compute $E/\langle P + \lambda Q \rangle$

Mixing key exchange and decompression

Key exchange w.r.t. public key $(j(E), \beta, \gamma, \delta, 0)$:

1. Compute the basis $\{R, S\}$
2. Decompress $P = R + \beta S$ and $Q = \gamma R + \delta S$
3. Compute $P + \lambda Q$
4. Compute $E/\langle P + \lambda Q \rangle$

Instead, do all scalar multiplications at once:

1. Compute the basis $\{R, S\}$
2. Compute

$$\langle P + \lambda Q \rangle = \langle R + (1 + \lambda\gamma)^{-1} (\beta + \lambda\delta) S \rangle$$

3. Compute $E/\langle P + \lambda Q \rangle$

Benchmarks

Implementation		This work	Prior work ([Aza+16])
PK (bytes)	uncompressed	564	768
	compressed	330	385
cc $\times 10^6$	A SIDH	90	–
	A compression	115	6,081
	A decompression	32	539
	B SIDH	102	–
	B compression	135	7,747
	B decompression	36	493
	Total	192	535
	Total (compression)	510	15,395

Thanks

Questions?

References I

[Are+09] Christophe Arene, Tanja Lange, Michael Naehrig and Christophe Ritzenthaler. "Faster Computation of the Tate Pairing". In: *IACR Cryptology ePrint Archive* 2009 (2009), p. 155. URL: http://eprint.iacr.org/2009/155.

[Aza+16] Reza Azarderakhsh, David Jao, Kassem Kalach, Brian Koziel and Christopher Leonardi. "Key Compression for Isogeny-Based Cryptosystems". In: *Proceedings of the 3rd ACM International Workshop on ASIA Public-Key Cryptography, AsiaPKC@AsiaCCS, Xi'an, China, May 30 - June 03, 2016*. Ed. by Keita Emura, Goichiro Hanaoka and Rui Zhang. ACM, 2016, pp. 1–10. DOI: 10.1145/2898420.2898421. URL: http://doi.acm.org/10.1145/2898420.2898421.

References II

[CLN16] Craig Costello, Patrick Longa and Michael Naehrig. "Efficient Algorithms for Supersingular Isogeny Diffie-Hellman". In: *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9814. Lecture Notes in Computer Science. Springer, 2016, pp. 572–601. DOI: 10.1007/978-3-662-53018-4_21. URL: http://dx.doi.org/10.1007/978-3-662-53018-4_21.

[DG16] Christina Delfs and Steven D. Galbraith. "Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$". In: *Des. Codes Cryptography* 78.2 (2016), pp. 425–440. DOI: 10.1007/s10623-014-0010-1. URL: http://dx.doi.org/10.1007/s10623-014-0010-1.

[FJP14] Luca De Feo, David Jao and Jérôme Plût. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *J. Mathematical Cryptology* 8.3 (2014), pp. 209–247. DOI: 10.1515/jmc-2012-0015. URL: http://dx.doi.org/10.1515/jmc-2012-0015.