# Efficient compression of SIDH public keys

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

[1]Microsoft Research, Redmond, USA

[2]University of Waterloo, Ontario, Canada

[3]Radboud University, Nijmegen, The Netherlands

1 May 2017

# Supersingular-isogeny Diffie-Hellman

- Post-quantum secure (ephemeral) key exchange [JF11]
- Based on hardness of finding large-degree isogenies
- Small keys ($\approx 564$ bytes public)
- Relatively slow compared to other PQ proposals
- Key compression ($\approx 385$ bytes), at very high cost [Aza+16]

# Supersingular-isogeny Diffie-Hellman

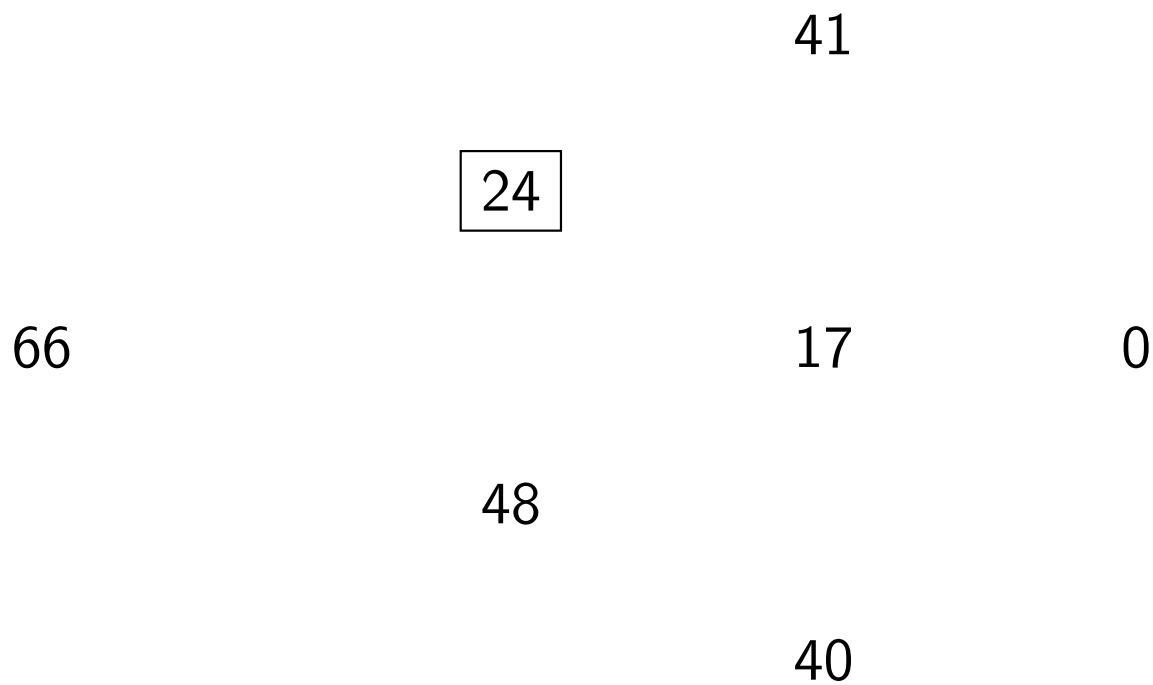
- Post-quantum secure (ephemeral) key exchange [JF11]
- Based on hardness of finding large-degree isogenies
- Small keys ($\approx$ 564 bytes public)
- Relatively slow compared to other PQ proposals
- Key compression ($\approx$ 385 bytes), at very high cost [Aza+16]

## This talk

- Key size reduced by 12.5% ($\approx$ 330 bytes)
- Compression up to 66$\times$ faster
- Decompression up to 15$\times$ faster

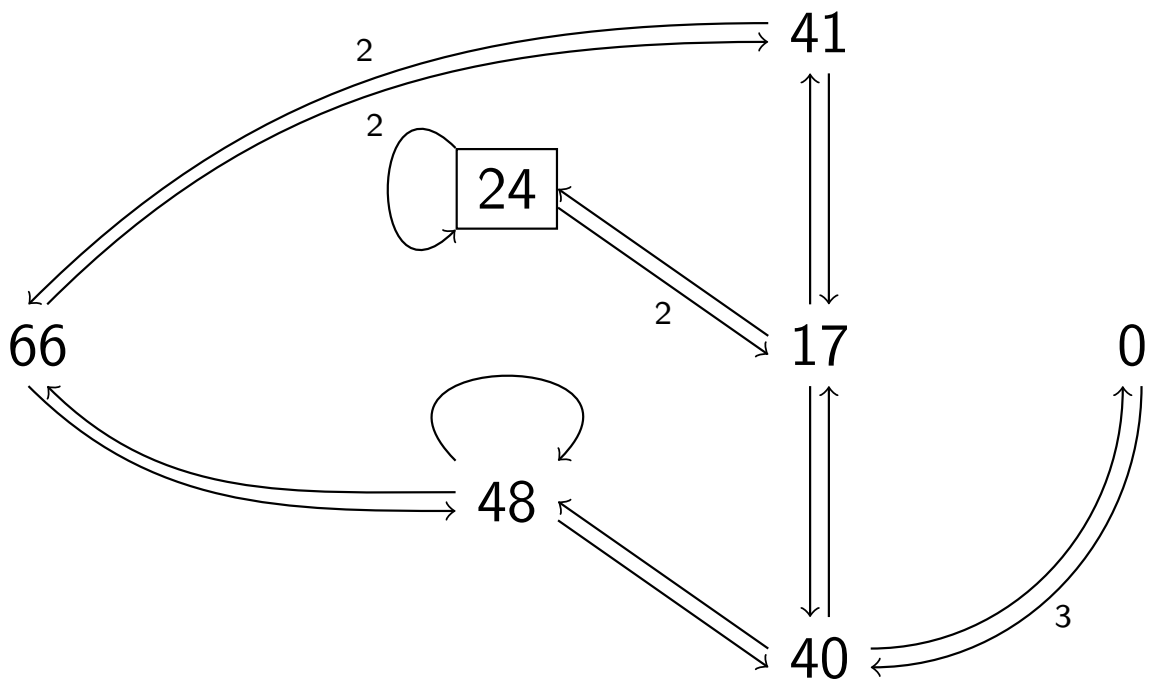
# Isogeny graphs

$$p = 2^3 \cdot 3^2 - 1, \quad E/\mathbb{F}_{p^2} : y^2 = x^3 + x, \quad j(E) = 24, \quad \ell = 2$$



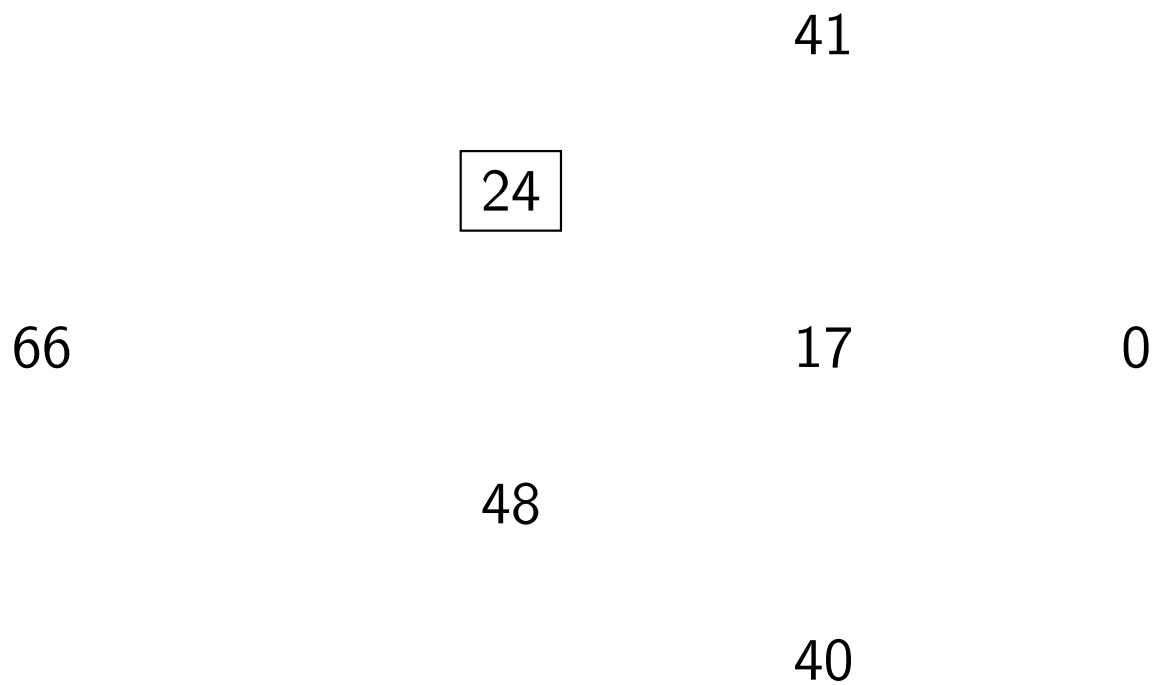
# Isogeny graphs

$$p = 2^3 \cdot 3^2 - 1, \quad E/\mathbb{F}_{p^2} : y^2 = x^3 + x, \quad j(E) = 24, \quad \ell = 2$$



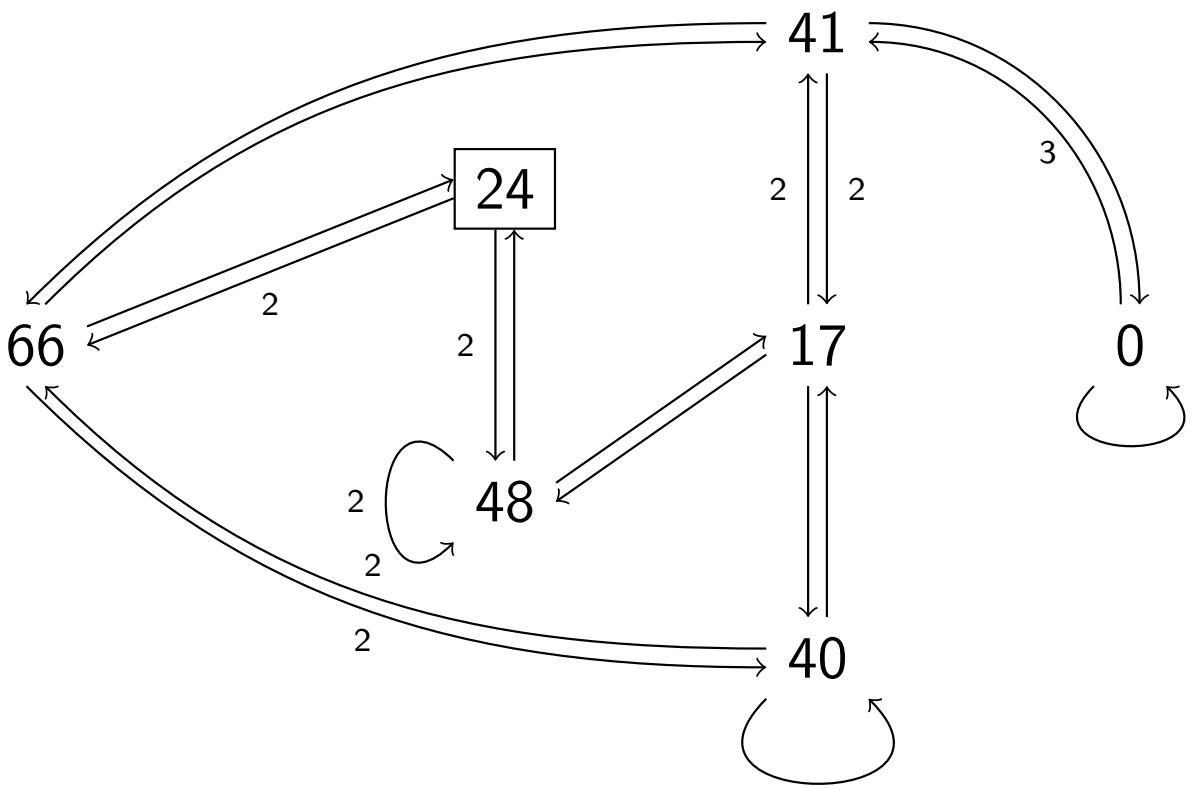
# Isogeny graphs

$$p = 2^3 \cdot 3^2 - 1, \quad E/\mathbb{F}_{p^2} : y^2 = x^3 + x, \quad j(E) = 24, \quad \ell = 2$$



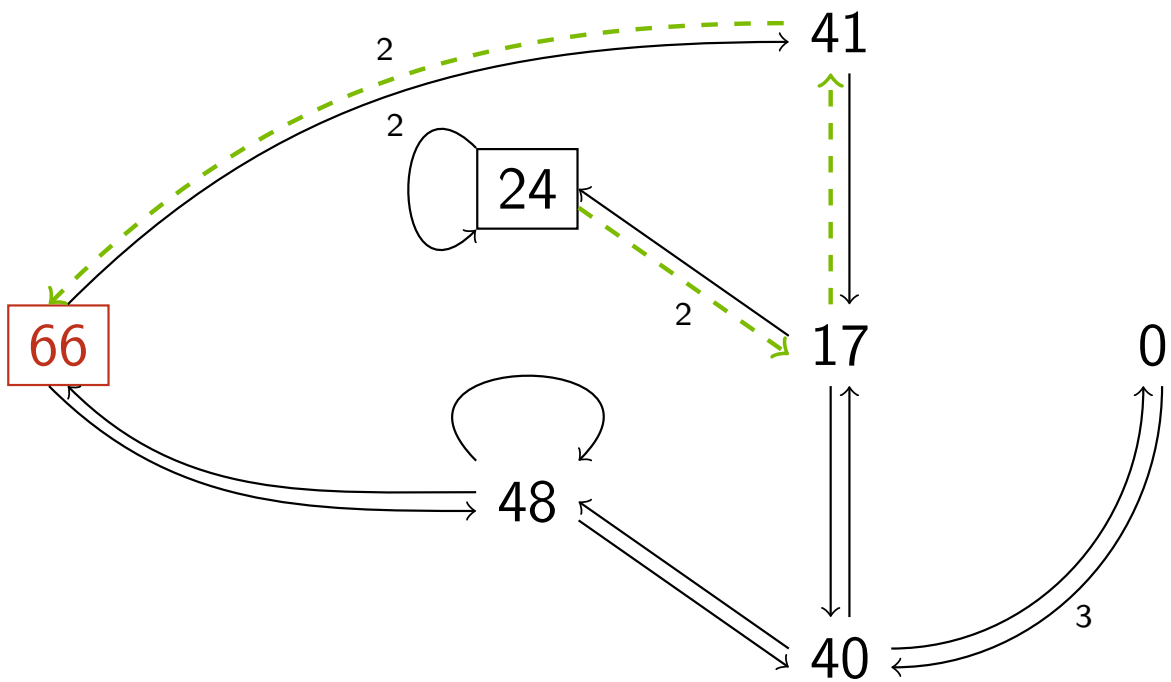
# Isogeny graphs

$$p = 2^3 \cdot 3^2 - 1, \quad E/\mathbb{F}_{p^2} : y^2 = x^3 + x, \quad j(E) = 24, \quad \ell = 3$$



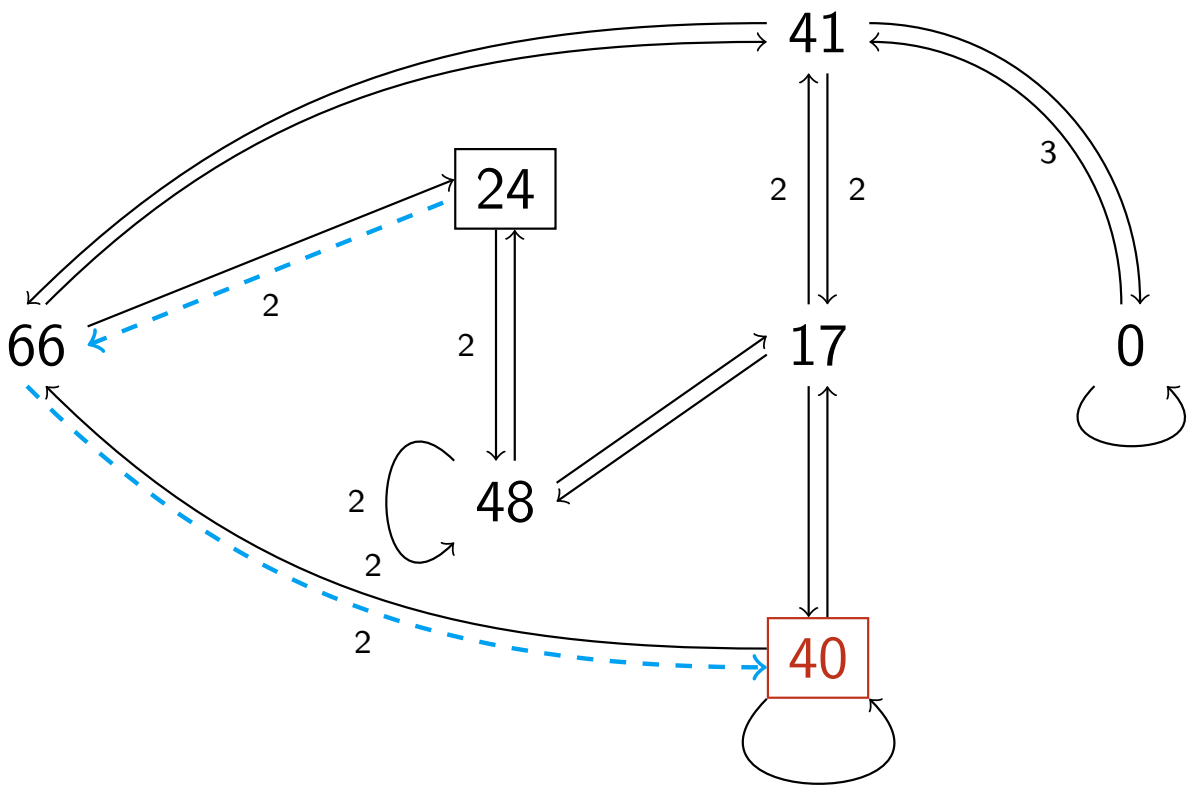
# Key generation

= private party $A$, = private party $B$, = public keys



# Key generation

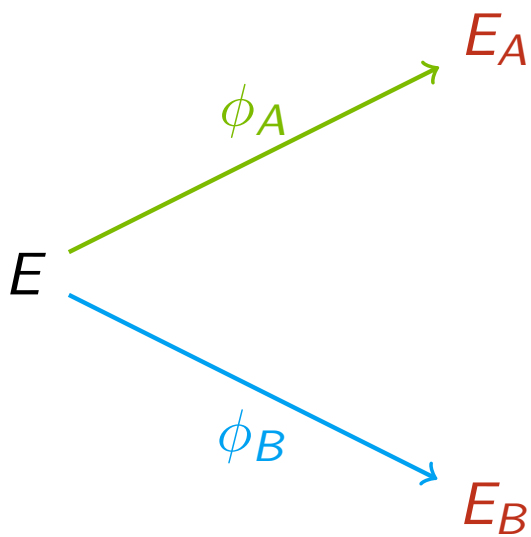
= private party $A$, = private party $B$, = public keys



# Supersingular-isogeny Diffie-Hellman [JF11]

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

$\nearrow$ = 2-graph walk, $\searrow$ = 3-graph walk,

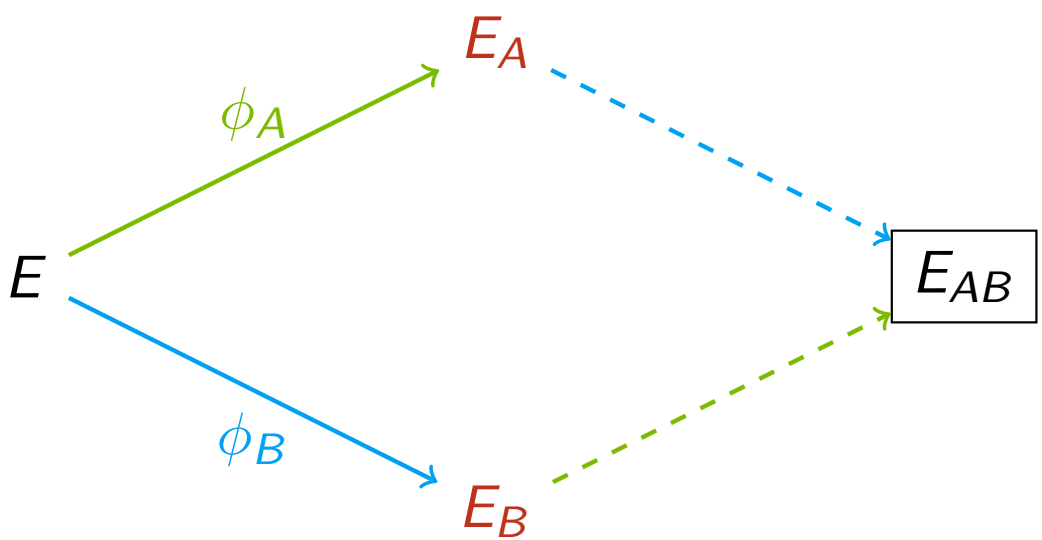


# Supersingular-isogeny Diffie-Hellman [JF11]

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

↗ = 2-graph walk, ↘ = 3-graph walk,

$$E_A[\ell^e] = \langle P, Q \rangle$$



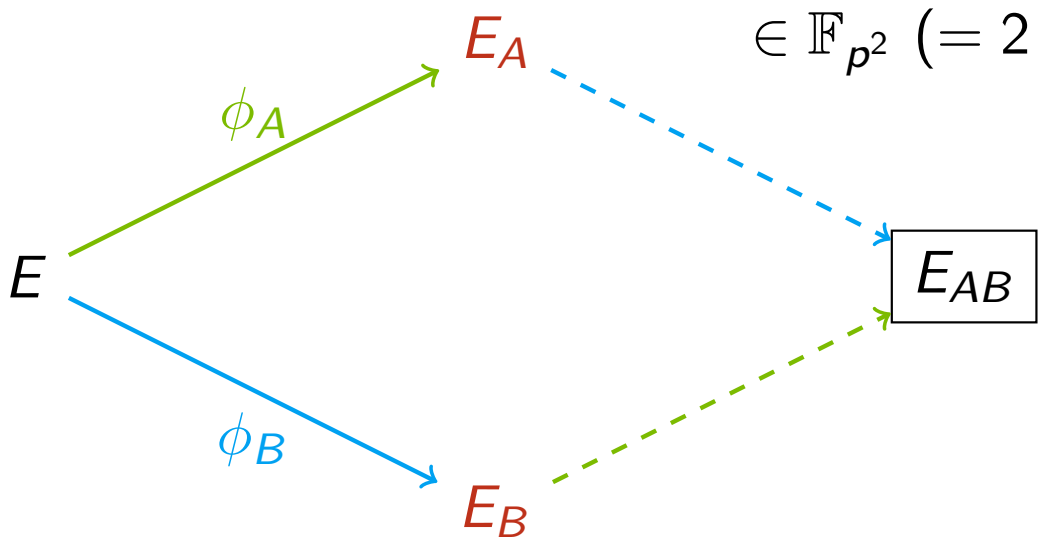
# Supersingular-isogeny Diffie-Hellman [JF11]

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

↗ = 2-graph walk, ↘ = 3-graph walk,

$E_A[\ell^e] = \langle P, Q \rangle \quad \in \mathbb{F}_{p^2}^2 \ (= 4 \log p \text{ bits})$

$\in \mathbb{F}_{p^2} \ (= 2 \log p \text{ bits})$



# Supersingular-isogeny Diffie-Hellman [JF11]

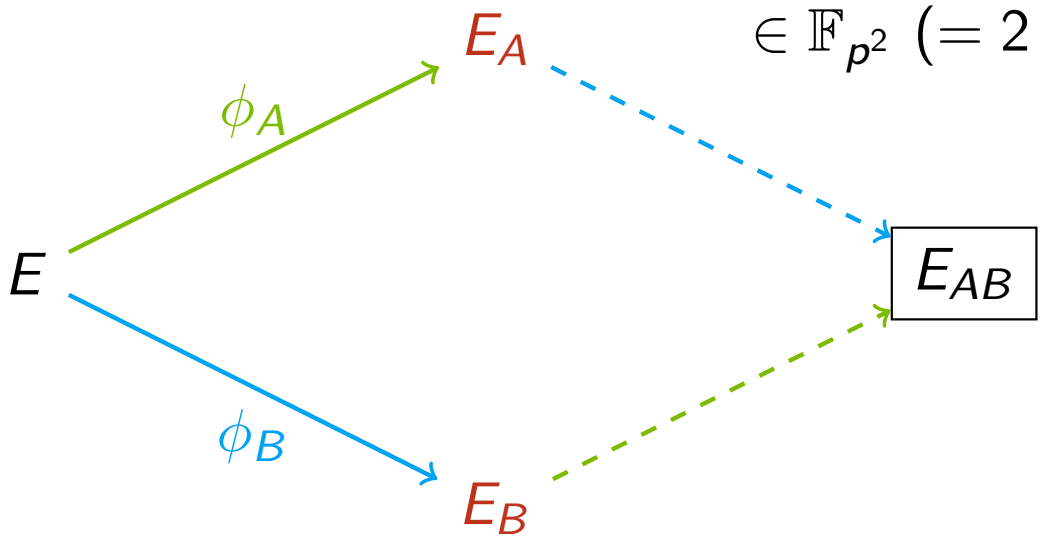
■ = private party $A$, ■ = private party $B$, ■ = public key

↗ = 2-graph walk, ↘ = 3-graph walk,

$$E_A[\ell^e] = \langle R, S \rangle$$

$$E_A[\ell^e] = \langle P, Q \rangle \quad \in \mathbb{F}_{p^2}^2 \ (= 4 \log p \text{ bits})$$

$$\in \mathbb{F}_{p^2} \ (= 2 \log p \text{ bits})$$



# Supersingular-isogeny Diffie-Hellman [JF11]

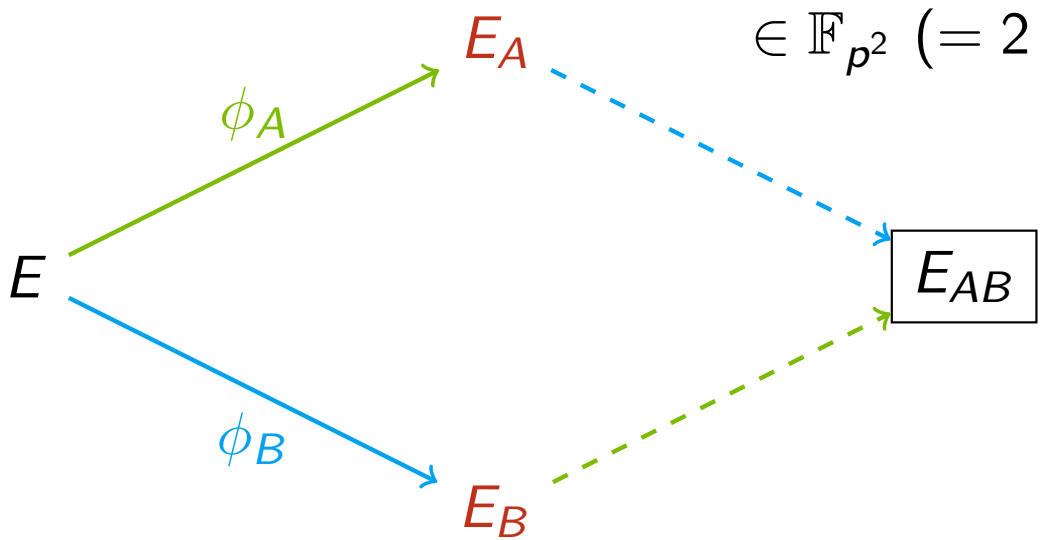
■ = private party $A$, ■ = private party $B$, ■ = public key

↗ = 2-graph walk, ↘ = 3-graph walk,

$$E_A[\ell^e] = \langle R, S \rangle$$

$(\alpha, \beta, \gamma, \delta) \in \mathbb{Z}_{\ell^e}^4$ ($\approx 2 \log p$ bits)

$\in \mathbb{F}_{p^2}$ ($= 2 \log p$ bits)



# Public-key compression [Aza+16]

**Compression**

$$\langle P, Q\rangle \longrightarrow \begin{array}{c}\langle R, S\rangle \\ \langle \alpha R + \beta S, \gamma R + \delta S\rangle\end{array} \longrightarrow (\alpha, \beta, \gamma, \delta)$$

**Decompression**

$$(\alpha, \beta, \gamma, \delta) \longrightarrow \begin{array}{c}\langle R, S\rangle \\ (\alpha, \beta, \gamma, \delta)\end{array} \longrightarrow \langle P, Q\rangle$$

# Public-key compression [Aza+16]

**Compression**

$$\langle P, Q\rangle \longrightarrow \begin{array}{c}\langle R, S\rangle \\ \langle \alpha R + \beta S, \gamma R + \delta S\rangle\end{array} \xrightarrow{\text{Expensive}} (\alpha, \beta, \gamma, \delta)$$

**Decompression**

$$(\alpha, \beta, \gamma, \delta) \longrightarrow \begin{array}{c}\langle R, S\rangle \\ (\alpha, \beta, \gamma, \delta)\end{array} \longrightarrow \langle P, Q\rangle$$

# Public-key compression [Aza+16]

Significantly improve efficiency (up to 66×)

**Compression**

$$\langle P, Q\rangle \longrightarrow \begin{matrix}\langle R, S\rangle \\ \langle \alpha R + \beta S, \gamma R + \delta S\rangle\end{matrix} \longrightarrow (\alpha, \beta, \gamma, \delta)$$

**Decompression**

$$(\alpha, \beta, \gamma, \delta) \longrightarrow \begin{matrix}\langle R, S\rangle \\ (\alpha, \beta, \gamma, \delta)\end{matrix} \longrightarrow \langle P, Q\rangle$$

Significantly improve efficiency (up to 15×)

# Finding a canonical basis

Find $R, S$ such that $E[2^{372}] = \langle R, S \rangle$, where

$$\#E(\mathbb{F}_{p^2}) = \left(2^{372}3^{239}\right)^2.$$

# Finding a canonical basis

Find $R, S$ such that $E[2^{372}] = \langle R, S \rangle$, where

$$\#E(\mathbb{F}_{p^2}) = \left(2^{372}3^{239}\right)^2.$$

## Finding an element of order $2^{372}$

1. Deterministically pick $R \in E(\mathbb{F}_{p^2}) \setminus 2E(\mathbb{F}_{p^2})$

# Finding a canonical basis

Find $R, S$ such that $E[2^{372}] = \langle R, S \rangle$, where

$$\#E(\mathbb{F}_{p^2}) = \left(2^{372}3^{239}\right)^2.$$

## Finding an element of order $2^{372}$

1. Deterministically pick $R \in E(\mathbb{F}_{p^2}) \setminus 2E(\mathbb{F}_{p^2})$

For $E : y^2 = x(x-\gamma)(x-\delta)$,

$$R \in 2E(\mathbb{F}_{p^2}) \iff x_R, x_R - \delta, x_R - \gamma \text{ are squares}$$

# Finding a canonical basis

Find $R, S$ such that $E[2^{372}] = \langle R, S \rangle$, where

$$\#E(\mathbb{F}_{p^2}) = \left(2^{372}3^{239}\right)^2.$$

## Finding an element of order $2^{372}$

1. Deterministically pick a non-square $x_R \in \mathbb{F}_{p^2}$

For $E : y^2 = x(x-\gamma)(x-\delta)$,

$$R \in 2E(\mathbb{F}_{p^2}) \iff x_R, x_R - \delta, x_R - \gamma \text{ are squares}$$

# Finding a canonical basis

Find $R, S$ such that $E[2^{372}] = \langle R, S \rangle$, where

$$\#E(\mathbb{F}_{p^2}) = \left(2^{372}3^{239}\right)^2.$$

## Finding an element of order $2^{372}$

1. Deterministically pick a non-square $x_R \in \mathbb{F}_{p^2}$
2. If $x_R^3 + Ax_R^2 + x_R$ is not a square, goto 1

# Finding a canonical basis

Find $R, S$ such that $E[2^{372}] = \langle R, S \rangle$, where

$$\#E(\mathbb{F}_{p^2}) = \left(2^{372}3^{239}\right)^2.$$

## Finding an element of order $2^{372}$

1. Deterministically pick a non-square $x_R \in \mathbb{F}_{p^2}$
2. If $x_R^3 + Ax_R^2 + x_R$ is not a square, goto 1
3. Set $R \leftarrow (x_R, \sqrt{x_R^3 + Ax_R^2 + x_R})$

# Finding a canonical basis

Find $R, S$ such that $E[2^{372}] = \langle R, S \rangle$, where

$$\#E(\mathbb{F}_{p^2}) = \left(2^{372}3^{239}\right)^2.$$

## Finding an element of order $2^{372}$

1. Deterministically pick a non-square $x_R \in \mathbb{F}_{p^2}$
2. If $x_R^3 + Ax_R^2 + x_R$ is not a square, goto 1
3. Set $R \leftarrow (x_R, \sqrt{x_R^3 + Ax_R^2 + x_R})$
4. Set $R \leftarrow [3^{239}]R$

# Finding a canonical basis

Find $R, S$ such that $E[2^{372}] = \langle R, S \rangle$, where

$$\#E(\mathbb{F}_{p^2}) = \left(2^{372}3^{239}\right)^2.$$

## Finding an element of order $2^{372}$

1. Deterministically pick a non-square $x_R \in \mathbb{F}_{p^2}$
2. If $x_R^3 + Ax_R^2 + x_R$ is not a square, goto 1
3. Set $R \leftarrow (x_R, \sqrt{x_R^3 + Ax_R^2 + x_R})$
4. Set $R \leftarrow [3^{239}]R$

## Finding a canonical basis of $E[2^{372}]$

1. Pick $R \in E(\mathbb{F}_{p^2})$ of order $2^{372}$
2. Pick $S \in E(\mathbb{F}_{p^2})$ of order $2^{372}$
3. If $E[2^{372}] \neq \langle R, S \rangle$, goto 2.

# Transferring to $\mu_n$ via reduced Tate pairing

Transfer the discrete logs to $\mu_n$

$$e = e(R, S) \qquad e^{\beta} = e(R, P) \qquad e^{\delta} = e(R, Q)$$
$$e^{-\alpha} = e(S, P) \qquad e^{-\gamma} = e(S, Q)$$

such that $P = \alpha R + \beta S$ and $Q = \gamma R + \delta S$

# Transferring to $\mu_n$ via reduced Tate pairing

Transfer the discrete logs to $\mu_n$

$$e = e(R,S) \qquad e^{\beta} = e(R,P) \qquad e^{\delta} = e(R,Q)$$
$$e^{-\alpha} = e(S,P) \qquad e^{-\gamma} = e(S,Q)$$

such that $P = \alpha R + \beta S$ and $Q = \gamma R + \delta S$

$e(R,S)$ $e(R,P)$ $e(R,Q)$ $e(S,P)$ $e(S,Q)$

---

$f_0 \leftarrow f_{n,R}$

$f_0 \leftarrow f_0(S)$

$\vdots$

# Transferring to $\mu_n$ via reduced Tate pairing

Transfer the discrete logs to $\mu_n$

$$e = e(R,S) \qquad e^{\beta} = e(R,P) \qquad e^{\delta} = e(R,Q)$$
$$e^{-\alpha} = e(S,P) \qquad e^{-\gamma} = e(S,Q)$$

such that $P = \alpha R + \beta S$ and $Q = \gamma R + \delta S$

$e(R,S)$ $e(R,P)$ $e(R,Q)$ $e(S,P)$ $e(S,Q)$

---

$f_0 \leftarrow f_{n,R}$ $f_1 \leftarrow f_{n,R}$

$f_0 \leftarrow f_0(S)$ $f_1 \leftarrow f_1(P)$

$\vdots$ $\vdots$

# Transferring to $\mu_n$ via reduced Tate pairing

Transfer the discrete logs to $\mu_n$

$$e = e(R,S) \qquad e^{\beta} = e(R,P) \qquad e^{\delta} = e(R,Q)$$
$$e^{-\alpha} = e(S,P) \qquad e^{-\gamma} = e(S,Q)$$

such that $P = \alpha R + \beta S$ and $Q = \gamma R + \delta S$

$e(R,S)$ $e(R,P)$ $e(R,Q)$ $e(S,P)$ $e(S,Q)$

---

$f_0 \leftarrow f_{n,R}$

$f_0 \leftarrow f_0(S)$ $f_1 \leftarrow f_0(P)$

$\vdots$ $\vdots$

# Transferring to $\mu_n$ via reduced Tate pairing

Transfer the discrete logs to $\mu_n$

$$e = e(R,S) \qquad e^{\beta} = e(R,P) \qquad e^{\delta} = e(R,Q)$$
$$e^{-\alpha} = e(S,P) \qquad e^{-\gamma} = e(S,Q)$$

such that $P = \alpha R + \beta S$ and $Q = \gamma R + \delta S$

$e(R,S)$ $e(R,P)$ $e(R,Q)$ $e(S,P)$ $e(S,Q)$

---

$f_0 \leftarrow f_{n,R}$ $f_2 \leftarrow f_{n,R}$

$f_0 \leftarrow f_0(S)$ $f_1 \leftarrow f_0(P)$ $f_2 \leftarrow f_2(Q)$

$\vdots$ $\vdots$ $\vdots$

# Transferring to $\mu_n$ via reduced Tate pairing

Transfer the discrete logs to $\mu_n$

$$e = e(R,S) \qquad e^{\beta} = e(R,P) \qquad e^{\delta} = e(R,Q)$$
$$e^{-\alpha} = e(S,P) \qquad e^{-\gamma} = e(S,Q)$$

such that $P = \alpha R + \beta S$ and $Q = \gamma R + \delta S$

$e(R,S)$ $e(R,P)$ $e(R,Q)$ $e(S,P)$ $e(S,Q)$

---

$f_0 \leftarrow f_{n,R}$

$f_0 \leftarrow f_0(S)$ $f_1 \leftarrow f_0(P)$ $f_2 \leftarrow f_0(Q)$

$\vdots$ $\vdots$ $\vdots$

# Transferring to $\mu_n$ via reduced Tate pairing

Transfer the discrete logs to $\mu_n$

$$e = e(R,S) \qquad e^{\beta} = e(R,P) \qquad e^{\delta} = e(R,Q)$$
$$e^{-\alpha} = e(S,P) \qquad e^{-\gamma} = e(S,Q)$$

such that $P = \alpha R + \beta S$ and $Q = \gamma R + \delta S$

$e(R,S)$ $e(R,P)$ $e(R,Q)$ $e(S,P)$ $e(S,Q)$

---

$f_0 \leftarrow f_{n,R}$ $f_3 \leftarrow f_{n,S}$

$f_0 \leftarrow f_0(S)$ $f_1 \leftarrow f_0(P)$ $f_2 \leftarrow f_0(Q)$ $f_3 \leftarrow f_3(P)$

$\vdots$ $\vdots$ $\vdots$ $\vdots$

# Transferring to $\mu_n$ via reduced Tate pairing

Transfer the discrete logs to $\mu_n$

$$e = e(R,S) \qquad e^{\beta} = e(R,P) \qquad e^{\delta} = e(R,Q)$$
$$e^{-\alpha} = e(S,P) \qquad e^{-\gamma} = e(S,Q)$$

such that $P = \alpha R + \beta S$ and $Q = \gamma R + \delta S$

$e(R,S)$ $e(R,P)$ $e(R,Q)$ $e(S,P)$ $e(S,Q)$

---

$f_0 \leftarrow f_{n,R}$ $f_3 \leftarrow f_{n,S}$ $f_4 \leftarrow f_{n,S}$

$f_0 \leftarrow f_0(S)$ $f_1 \leftarrow f_0(P)$ $f_2 \leftarrow f_0(Q)$ $f_3 \leftarrow f_3(P)$ $f_4 \leftarrow f_4(Q)$

$\vdots$ $\vdots$ $\vdots$ $\vdots$ $\vdots$

# Transferring to $\mu_n$ via reduced Tate pairing

Transfer the discrete logs to $\mu_n$

$$e = e(R,S) \qquad e^{\beta} = e(R,P) \qquad e^{\delta} = e(R,Q)$$
$$e^{-\alpha} = e(S,P) \qquad e^{-\gamma} = e(S,Q)$$

such that $P = \alpha R + \beta S$ and $Q = \gamma R + \delta S$

$e(R,S)$ $e(R,P)$ $e(R,Q)$ $e(S,P)$ $e(S,Q)$

---

$f_0 \leftarrow f_{n,R}$ $f_3 \leftarrow f_{n,S}$

$f_0 \leftarrow f_0(S)$ $f_1 \leftarrow f_0(P)$ $f_2 \leftarrow f_0(Q)$ $f_3 \leftarrow f_3(P)$ $f_4 \leftarrow f_3(Q)$

$\vdots$ $\vdots$ $\vdots$ $\vdots$ $\vdots$

# Transferring to $\mu_n$ via reduced Tate pairing

Transfer the discrete logs to $\mu_n$

$$e = e(R,S) \qquad e^{\beta} = e(R,P) \qquad e^{\delta} = e(R,Q)$$
$$e^{-\alpha} = e(S,P) \qquad e^{-\gamma} = e(S,Q)$$

such that $P = \alpha R + \beta S$ and $Q = \gamma R + \delta S$

$e(R,S)$ $e(R,P)$ $e(R,Q)$ $e(S,P)$ $e(S,Q)$

---

$f_0 \leftarrow f_{n,R}$ $f_3 \leftarrow f_{n,S}$

$f_0 \leftarrow f_0(S)$ $f_1 \leftarrow f_0(P)$ $f_2 \leftarrow f_0(Q)$ $f_3 \leftarrow f_3(P)$ $f_4 \leftarrow f_3(Q)$

$\vdots$ $\vdots$ $\vdots$ $\vdots$ $\vdots$

**Optimized formulas for $f_{n,R}$ and $f_{n,S}$!**

# Efficient discrete logarithms (Pohlig-Hellman)

For $e_0, e_1, e_2, e_3, e_4 \in \mu_{\ell^e}$, compute $\alpha, \beta, \gamma, \delta$ such that

$$e_1 = e_0^{-\alpha}, \qquad e_2 = e_0^{\beta}, \qquad e_3 = e_0^{-\gamma}, \qquad e_4 = e_0^{\delta}$$

As $\mu_{\ell^e} \subset G_{p+1} \subset \mathbb{F}_{p^2}$, $\mathbf{I} \approx \mathbf{M}$, $\mathbf{S} \approx 2\mathbf{s}$, $\mathbf{C} \approx 2\mathbf{m} + 1\mathbf{s}$

# Efficient discrete logarithms (Pohlig-Hellman)

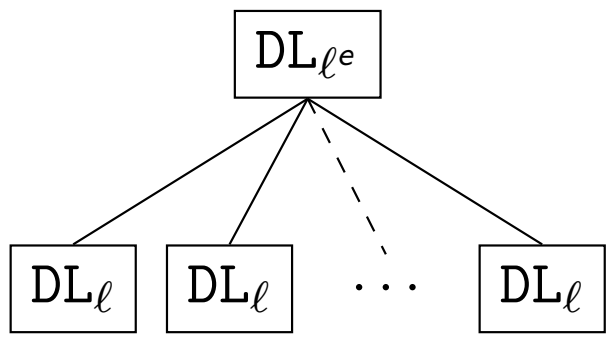
For $e_0, e_1, e_2, e_3, e_4 \in \mu_{\ell^e}$, compute $\alpha, \beta, \gamma, \delta$ such that

$$e_1 = e_0^{-\alpha}, \qquad e_2 = e_0^{\beta}, \qquad e_3 = e_0^{-\gamma}, \qquad e_4 = e_0^{\delta}$$

As $\mu_{\ell^e} \subset G_{p+1} \subset \mathbb{F}_{p^2}$, $\mathbf{I} \approx \mathbf{M}$, $\mathbf{S} \approx 2\mathbf{s}$, $\mathbf{C} \approx 2\mathbf{m} + 1\mathbf{s}$

$\#G_1 = \ell^e$

$\#G_1 = \ell$



# Nested Pohlig-Hellman

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

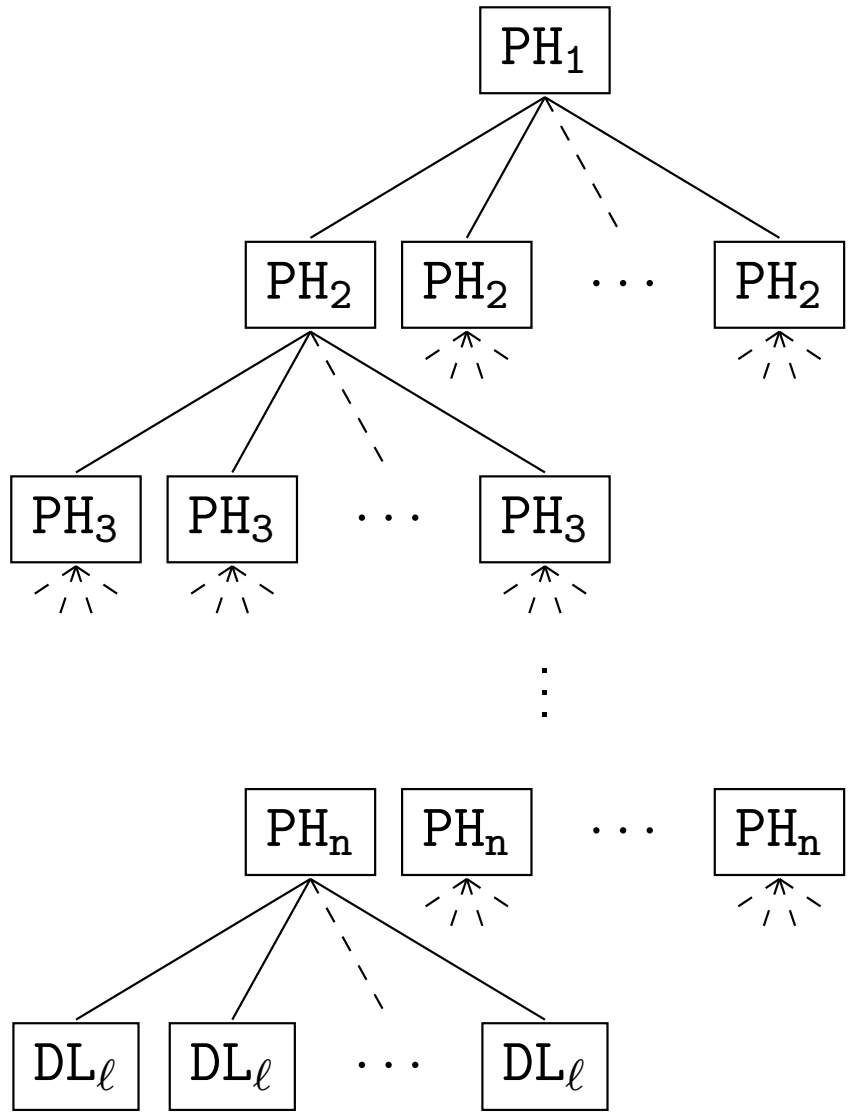
$$\#G_2 = \ell^{e_2}$$

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

$$\vdots$$

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

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



# Comparison

| # | 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 |

Options for different time-memory trade-offs [Sut11]

# Signature size reduction

- The quadruple $(\alpha, \beta, \gamma, \delta) \in \mathbb{Z}_{\ell^e}^4$ determines

$$P = \alpha R + \beta S, \quad Q = \gamma R + \delta S.$$

  These determine $\langle P + \lambda Q \rangle$, for some $\lambda \in \mathbb{Z}_{\ell^e}^*$
- Thus we only need $P, Q$ up to scalar, and compress to

$$[\alpha : \beta : \gamma : \delta] .$$

  As $P, Q$ form a basis of $E[\ell^e]$, either $\alpha$ or $\beta$ is invertible
- Normalizing, we represent it in $\mathbb{Z}_{\ell^e}^3 \times \mathbb{Z}_2$

# Benchmarks (for $\ell = 2$)

| | **This work** | [Aza+16] | Speed-up |
|---|---|---|---|
| Key size (bytes) | 328 | 385 | – |
| SIDH (cc $\times$ $10^6$) | 80 | – | – |
| Compression (cc $\times$ $10^6$) | 109 | 6 081 | 56$\times$ |
| Decompression (cc $\times$ $10^6$) | 42 | 539 | 13$\times$ |
| Full no comp. (cc $\times$ $10^6$) | 192 | 535 | 2.8$\times$ |
| Full comp. (cc $\times$ $10^6$) | 469 | 15 395 | 31$\times$ |

Software available at

https://github.com/Microsoft/PQCrypto-SIDH

# Thanks!

Questions?

# References I


[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.

[JF11] David Jao and Luca De Feo. "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies". In: *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings*. 2011, pp. 19–34. DOI: 10.1007/978-3-642-25405-5_2. URL: http://dx.doi.org/10.1007/978-3-642-25405-5_2.

# References II

[Sut11] Andrew V. Sutherland. "Structure computation and discrete logarithms in finite abelian $p$-groups". In: *Math. Comput.* 80.273 (2011), pp. 477–500. DOI: 10.1090/S0025-5718-10-02356-2. URL: http://dx.doi.org/10.1090/S0025-5718-10-02356-2.