Why to love isogenies in 2024

Chloe Martindale

University of Bristol

19th June 2024

Isogeny-based cryptography: why?

- Based on hard problems believed to (still) be hard for quantum computers

Isogeny-based cryptography: why?

- Based on hard problems believed to (still) be hard for quantum computers
- Low memory requirements
  - Lowest of all post-quantum candidates (by far)
  - Smallest options similar size to classical ECC

Isogeny-based cryptography: why?

- Based on hard problems believed to (still) be hard for quantum computers
- Low memory requirements
  - Lowest of all post-quantum candidates (by far)
  - Smallest options similar size to classical ECC
- Made up of ECC subroutines $\rightsquigarrow$ quite compatible with current small-device implementations

Isogeny-based cryptography: why?

- Based on hard problems believed to (still) be hard for quantum computers
- Low memory requirements
  - Lowest of all post-quantum candidates (by far)
  - Smallest options similar size to classical ECC
- Made up of ECC subroutines $\rightsquigarrow$ quite compatible with current small-device implementations
- Rich mathematical structure $\rightsquigarrow$ flexible post-quantum applications.

Isogeny-based cryptography: why?

- Based on hard problems believed to (still) be hard for quantum computers
- Low memory requirements
  - Lowest of all post-quantum candidates (by far)
  - Smallest options similar size to classical ECC
- Made up of ECC subroutines $\rightsquigarrow$ quite compatible with current small-device implementations
- Rich mathematical structure $\rightsquigarrow$ flexible post-quantum applications. Since 2018:
  - Only practical pq non-interactive key exchange
  - Fast, small key encapsulation mechanism
  - Two different signature schemes
  - Oblivious pseudorandom functions
  - Threshold schemes
  - ...

Isogeny-based cryptography: why not?

- Newest of all pq schemes $\rightsquigarrow$ less confidence in security.

Isogeny-based cryptography: why not?

- Newest of all pq schemes $\rightsquigarrow$ less confidence in security.
- Rich mathematical structure $\rightsquigarrow$ many attack avenues, maybe not all explored.

Isogeny-based cryptography: why not?

- Newest of all pq schemes $\rightsquigarrow$ less confidence in security.
- Rich mathematical structure $\rightsquigarrow$ many attack avenues, maybe not all explored.
- One Hard Problem admits a subexponential quantum attack; concrete complexity an active research topic.
  - Difficult to make concrete parameter choices.

Isogeny-based cryptography: why not?

- Newest of all pq schemes $\rightsquigarrow$ less confidence in security.
- Rich mathematical structure $\rightsquigarrow$ many attack avenues, maybe not all explored.
- One Hard Problem admits a subexponential quantum attack; concrete complexity an active research topic.
  - Difficult to make concrete parameter choices.
- Slow: Orders of magnitude slower than ECC or the fastest pq option (structured lattices).

Example: CRS (Couveignes '97 Rostostev-Stolbunov '04)

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

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x. \end{array}$$

Example: CRS (Couveignes '97 Rostostev-Stolbunov '04)

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

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x. \end{array}$$

$\rightsquigarrow$ Idea:

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

$$H \times S \to S.$$

Example: CRS (Couveignes '97 Rostostev-Stolbunov '04)

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

$$\begin{array}{ccc} \mathbb{Z} \times G & \to & G \\ (x, g) & \mapsto & g^x. \end{array}$$

$\rightsquigarrow$ Idea:

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

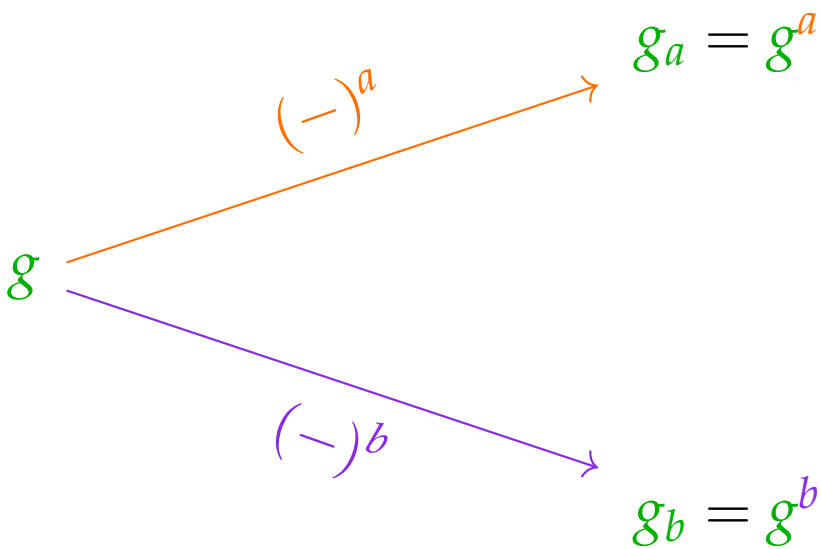
$$H \times S \to S.$$

- S is a (specially chosen) set of elliptic curves
- H acts via isogenies (maps between elliptic curves)


[ˈsiːˌsaɪd]

Evolution of key exchange

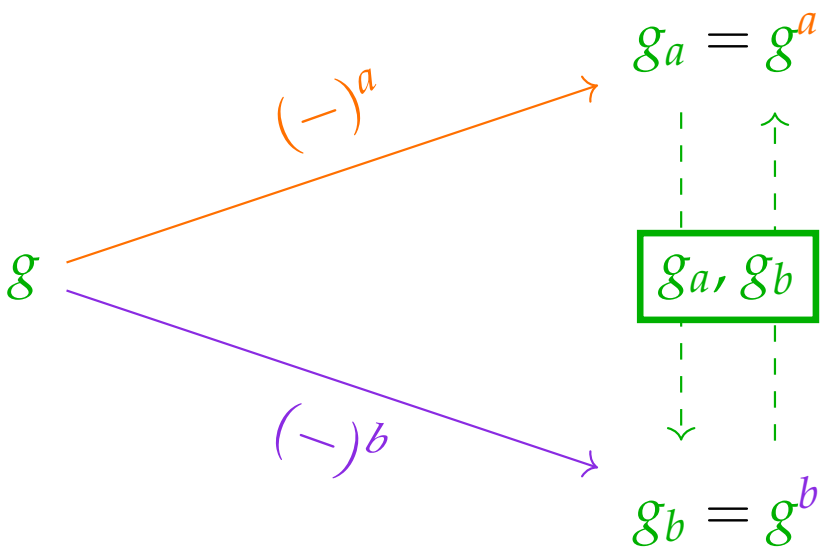
Diffie-Hellman



Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange

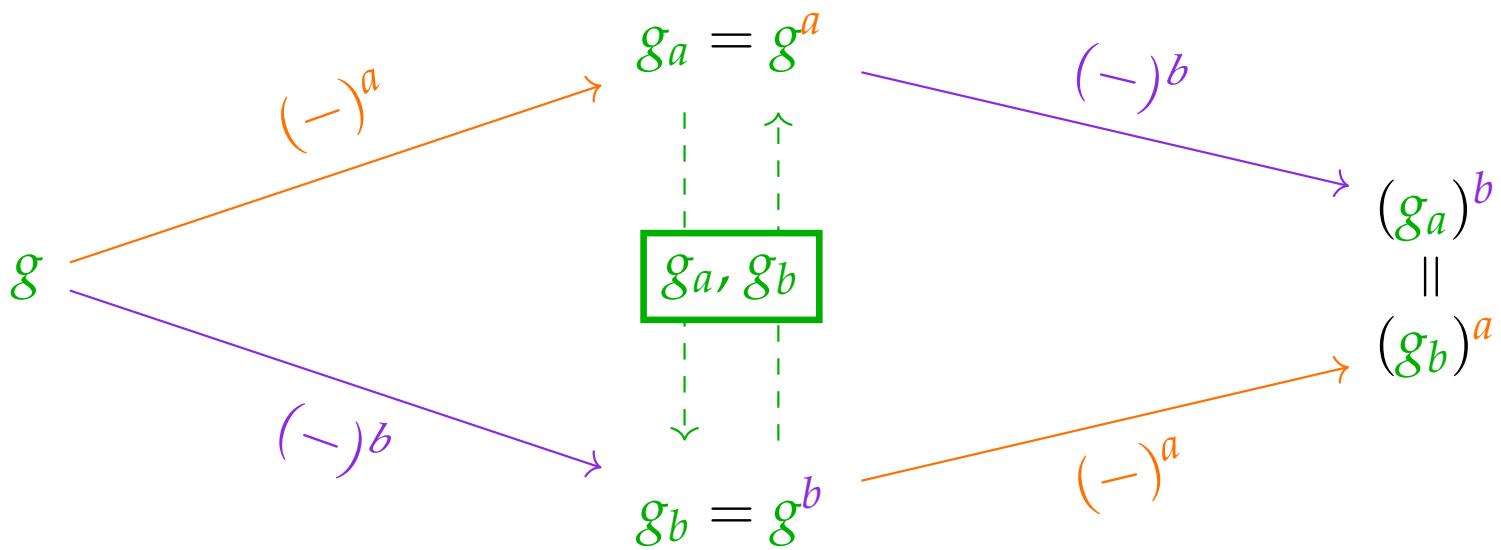
Diffie-Hellman



Colour code: Public, Alice's secret, Bob's secret

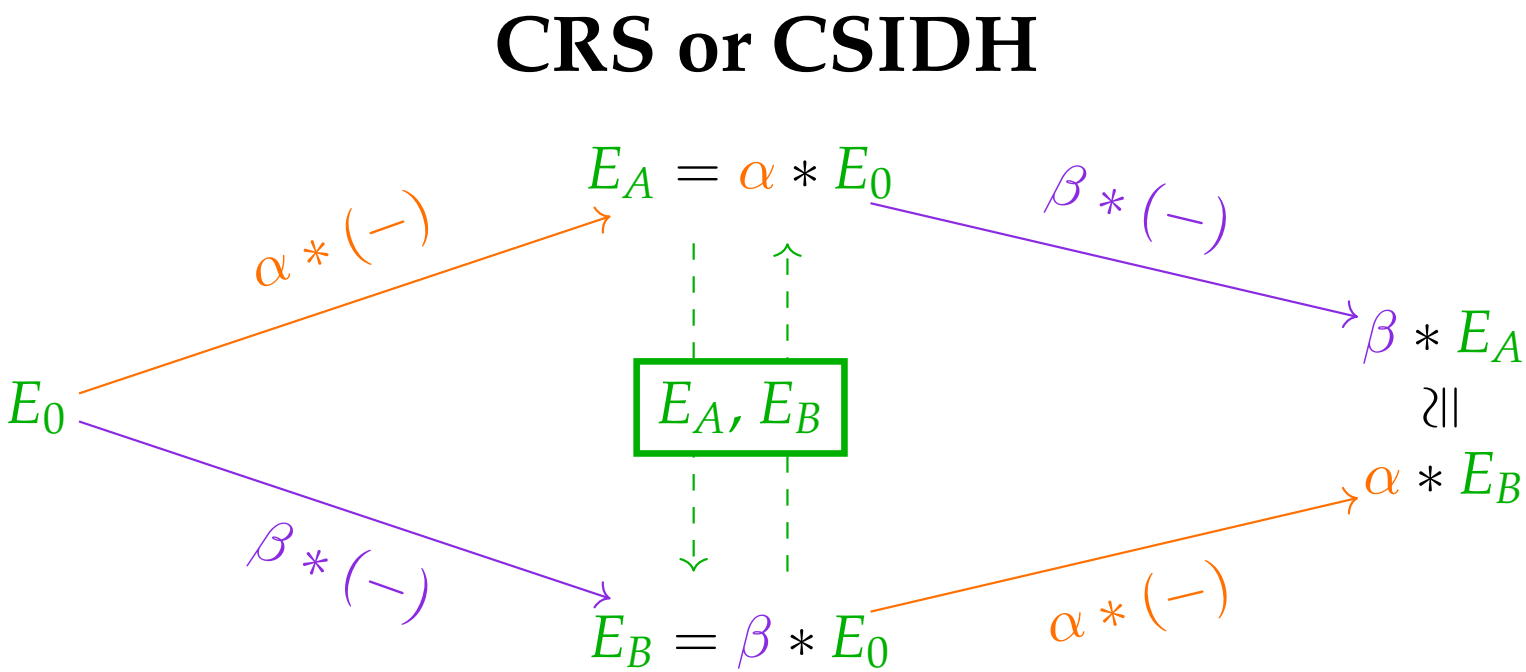
Evolution of key exchange

Diffie-Hellman



Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange



Colour code: Public, Alice's secret, Bob's secret

Signatures (S '06, DG '18, BKV '19, DFKLMPW '23)

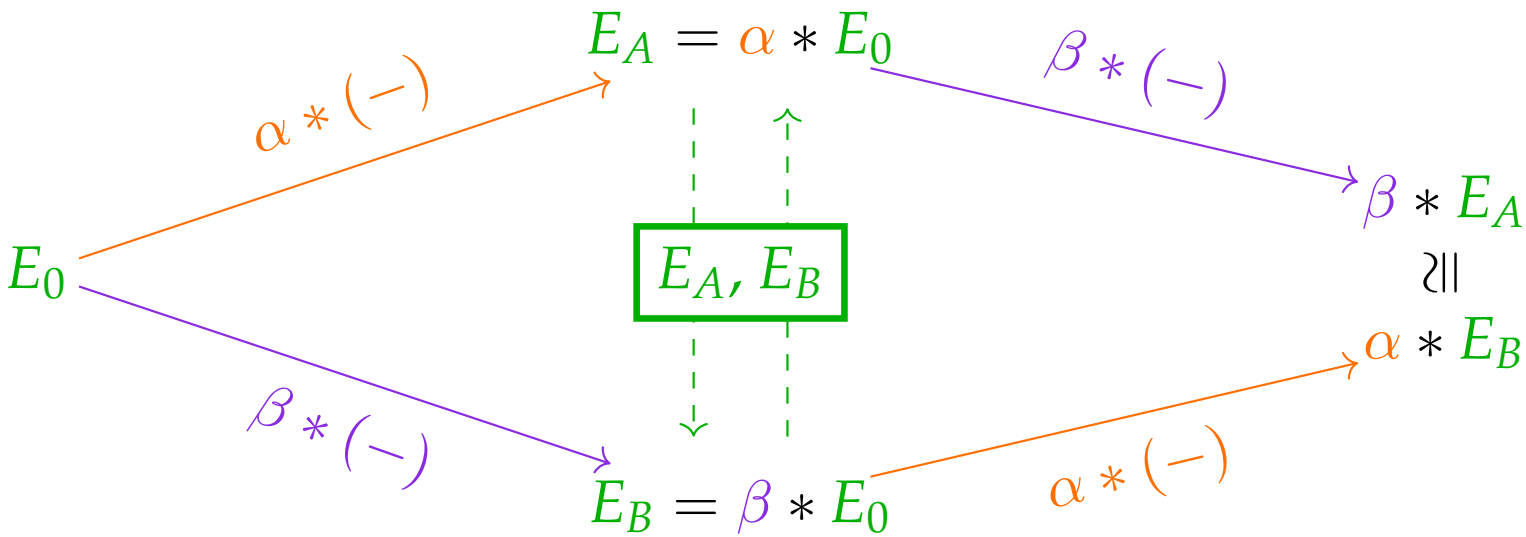
Identification scheme from $H \times S \to S$:

Prover	**Public**	**Verifier**
	$E \in S, \alpha_i \in H$	
$s_i \leftarrow \$\, \mathbb{Z}$		
$\mathsf{sk} = \prod \alpha_i{}^{s_i}$,		
$\mathsf{pk} = \mathsf{sk} * E$	$\xrightarrow{\mathsf{pk}}$	pk
	$\xleftarrow{c}$	$c \leftarrow \$\, \{0,1\}$
$t_i \leftarrow \$\, \mathbb{Z}$		
$\mathsf{esk} = \prod \alpha_i{}^{t_i}$,		
$\mathsf{epk}_1 = \mathsf{esk} * E$,		
$\mathsf{epk}_2 = \mathsf{esk} \cdot \mathsf{sk}^{-c}$	$\xrightarrow{\mathsf{pk},\mathsf{epk}_1,\mathsf{epk}_2}$	check:
		$\mathsf{epk}_1 = \mathsf{epk}_2 * ([\mathsf{sk}^c] * E)$.

After k challenges c, an imposter succeeds with prob 2^{-k}.

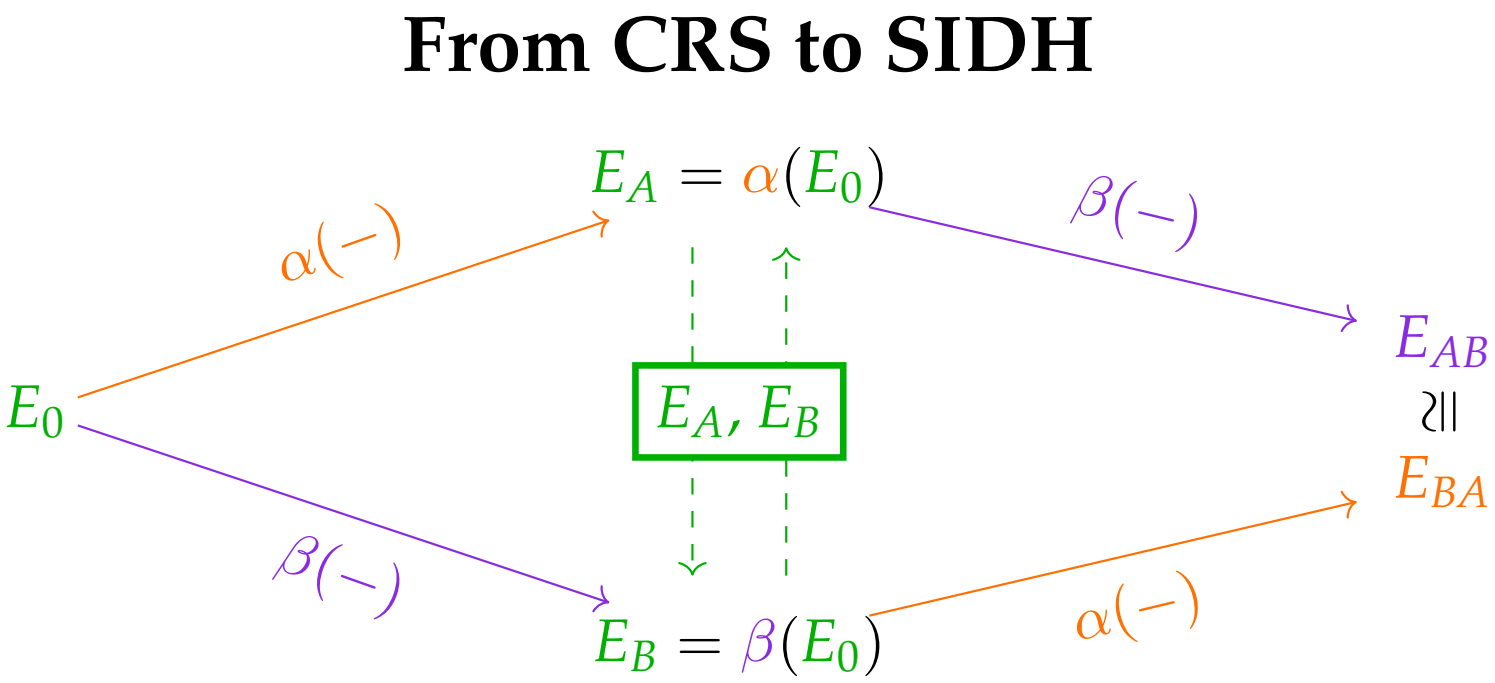
Evolution of key exchange

CRS or CSIDH



Colour code: Public, Alice's secret, Bob's secret

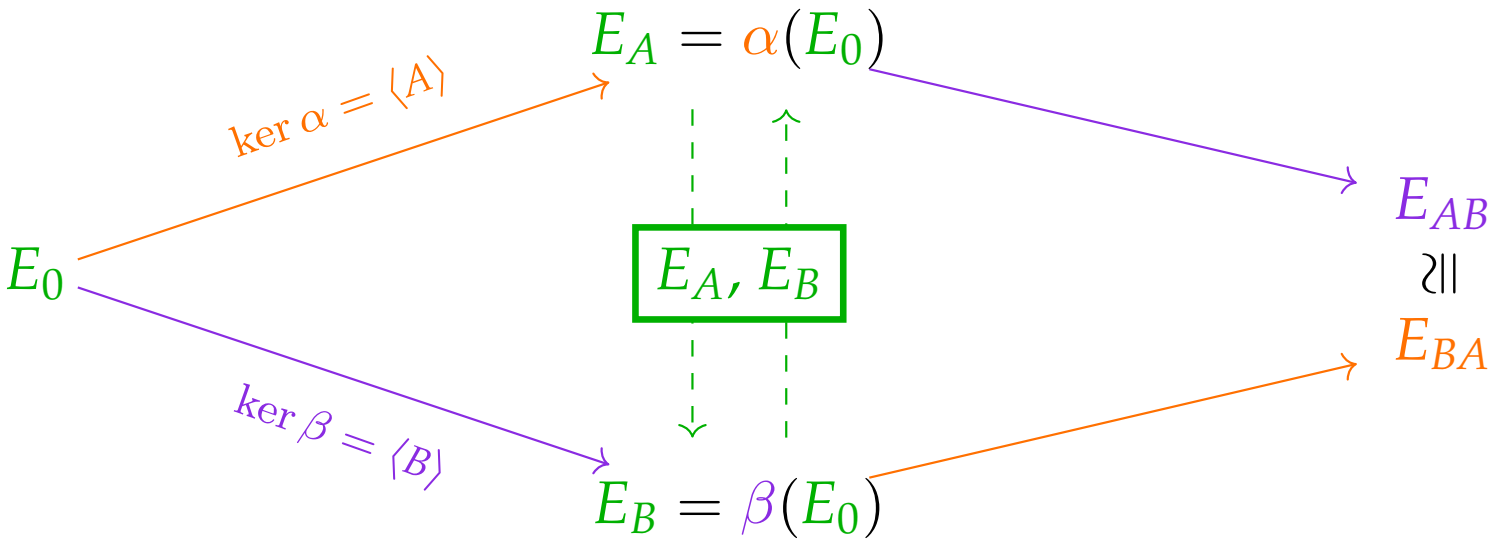
Evolution of key exchange



Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange

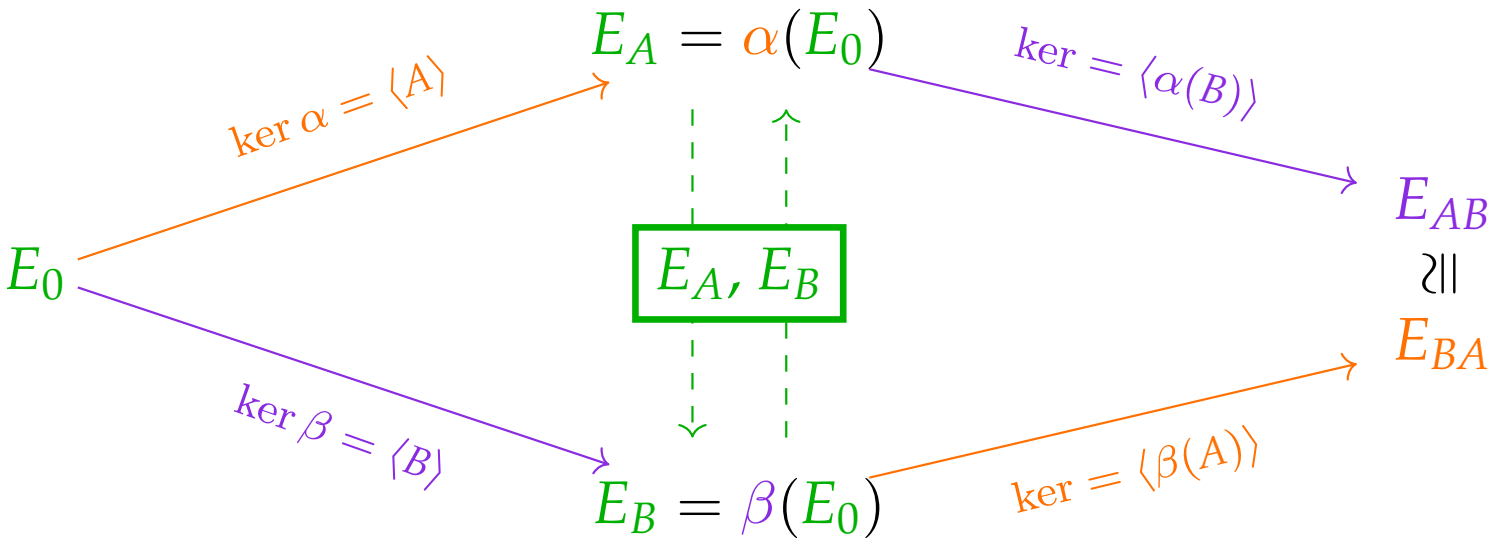
From CRS to SIDH



Colour code: Public, Alice's secret, Bob's secret

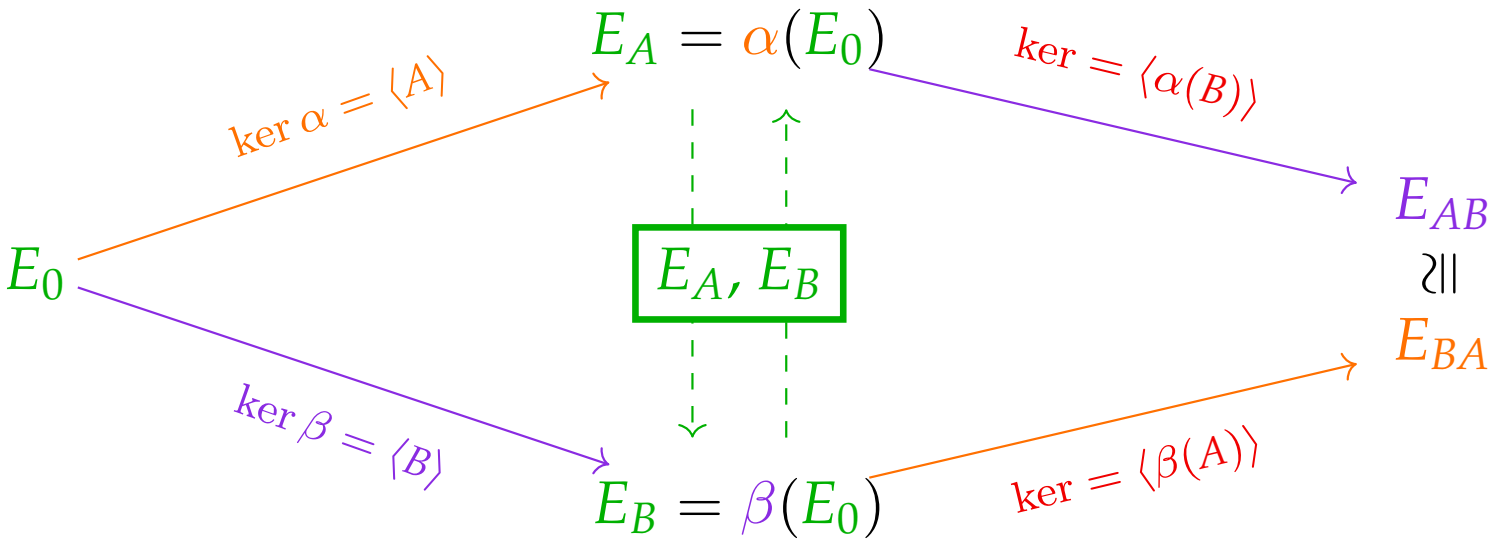
Evolution of key exchange

From CRS to SIDH



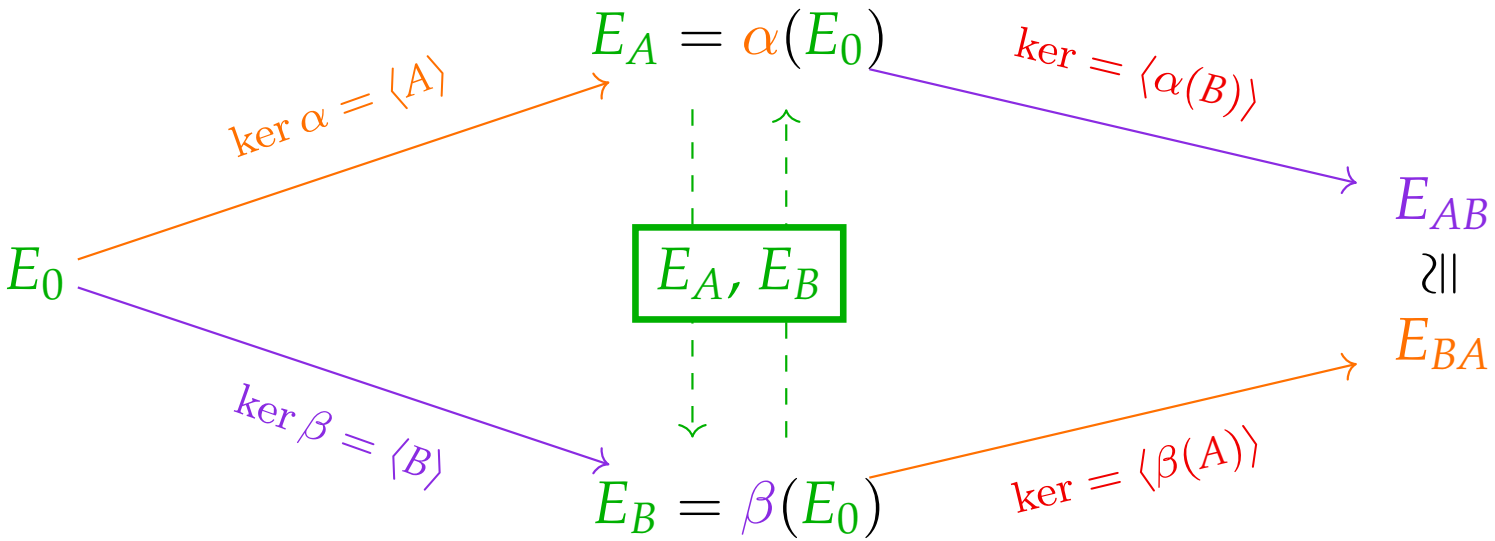
Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange



Colour code: Public, Alice's secret, Bob's secret, ?!

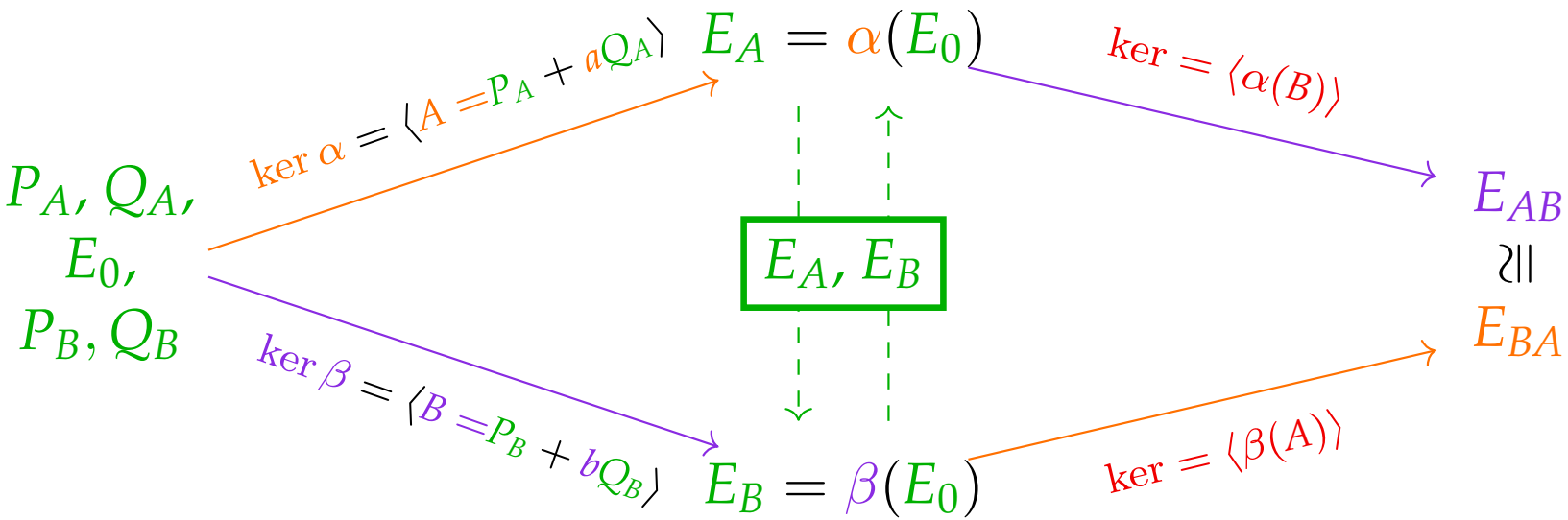
Evolution of key exchange



Colour code: Public, Alice's secret, Bob's secret, ?!

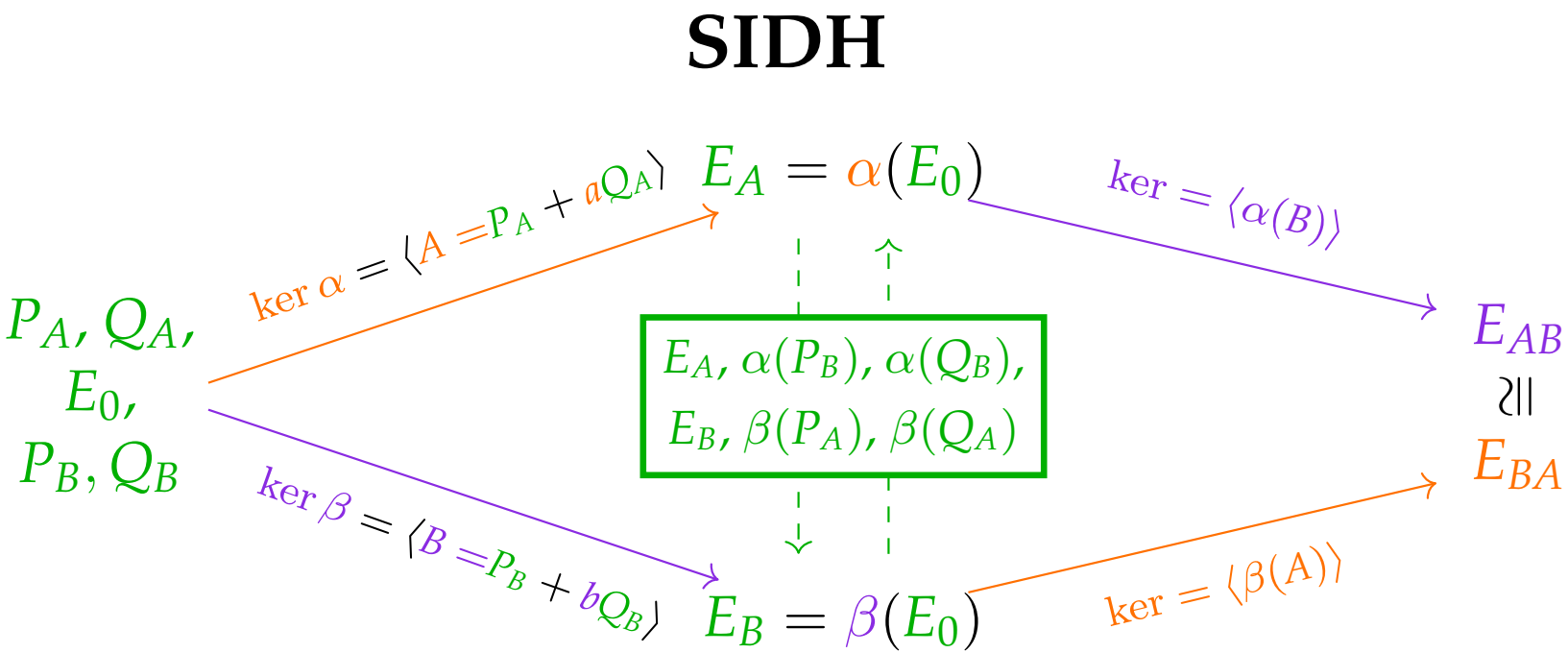
Evolution of key exchange

From CRS to SIDH



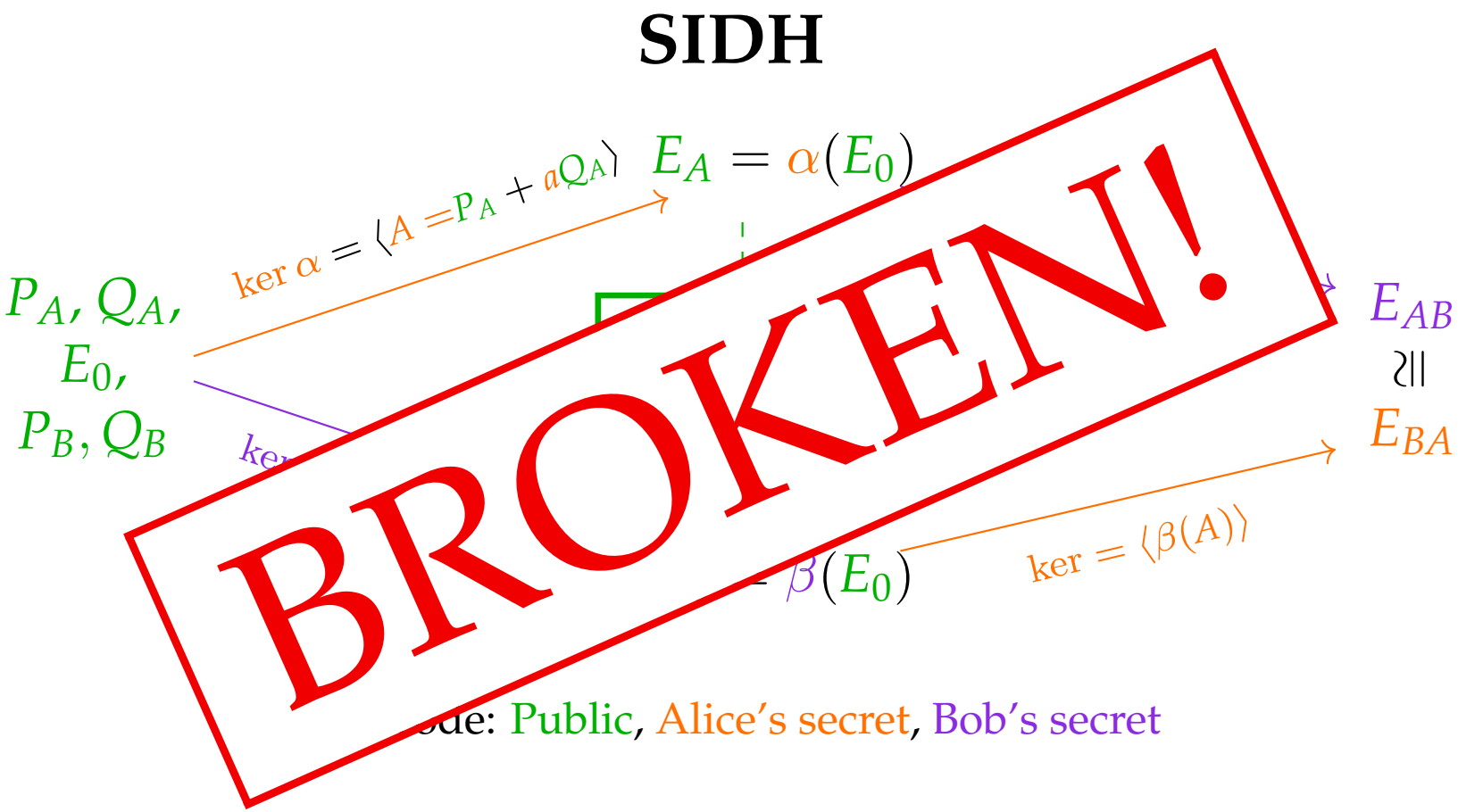
Colour code: Public, Alice's secret, Bob's secret, ?!

Evolution of key exchange



Colour code: Public, Alice's secret, Bob's secret

Evolution of key exchange



Summary of hard problems

- Diffie-Hellman – The Discrete Logarithm Problem, finding a given g and g^a.

Summary of hard problems

- Diffie-Hellman – The Discrete Logarithm Problem, finding a given g and g^a.
- CRS / CSIDH – Finding α given E and $\alpha * E$.

Summary of hard problems

- Diffie-Hellman – The Discrete Logarithm Problem, finding a given g and g^a.
- CRS / CSIDH – Finding α given E and $\alpha * E$.
- All isogeny-based schemes – Given elliptic curves E_0 and E_A, compute an isogeny $\alpha : E_0 \to E_A$ if it exists.

Summary of hard problems

- Diffie-Hellman – The Discrete Logarithm Problem, finding a given g and g^a.
- CRS / CSIDH – Finding α given E and $\alpha * E$.
- All isogeny-based schemes – Given elliptic curves E_0 and E_A, compute an isogeny $\alpha : E_0 \to E_A$ if it exists.
- SIDH –

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given the points P_B, Q_B on E_0 and $\alpha(P_B)$, $\alpha(Q_B)$, compute α. (modulo technical restrictions)*

*Details for the elliptic curve lovers:

p a large prime; $E_0/\mathbb{F}_{p^2}$ and $E_A/\mathbb{F}_{p^2}$ supersingular; $\deg(\alpha)$, B public large smooth coprime integers; points P_B, Q_B chosen such that $\langle P_B, Q_B \rangle = E_0[B]$.

Summary of hard problems

- Diffie-Hellman – The Discrete Logarithm Problem, finding a given g and g^a.
- CRS / CSIDH – Finding α given E and $\alpha * E$.
- All isogeny-based schemes – Given elliptic curves E_0 and E_A, compute an isogeny $\alpha : E_0 \to E_A$ if it exists.
- SIDH –

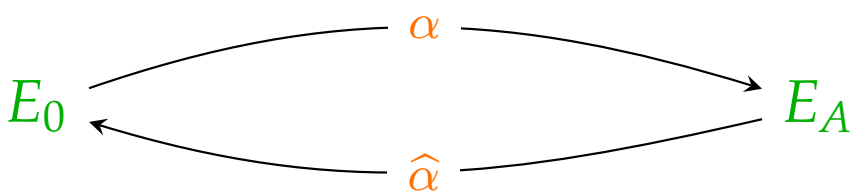
There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given $\alpha|_{E_0[B]}$, compute α. (modulo technical restrictions)*

*Details for the elliptic curve lovers:

p a large prime; $E_0/\mathbb{F}_{p^2}$ and $E_A/\mathbb{F}_{p^2}$ supersingular; $\deg(\alpha)$, B public large smooth coprime integers; points P_B, Q_B chosen such that $\langle P_B, Q_B \rangle = E_0[B]$.

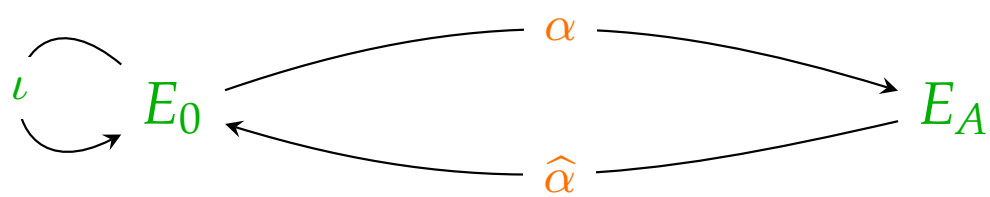
Petit's trick: torsion points to isogenies

Finding the secret isogeny α of known degree, given $\alpha|_{E_0[B]}$.



Petit's trick: torsion points to isogenies

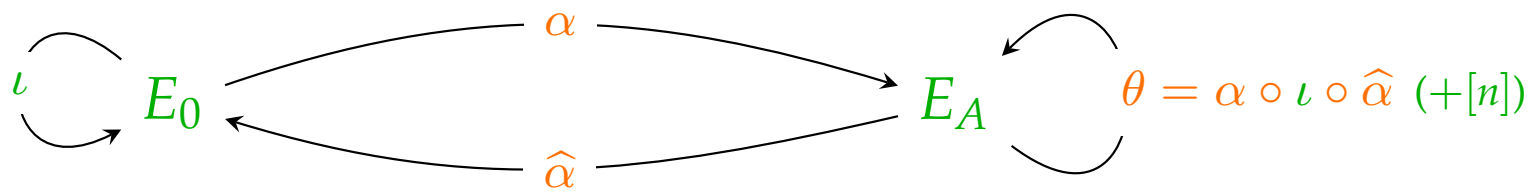
Finding the secret isogeny α of known degree, given $\alpha|_{E_0[B]}$.



- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.

Petit's trick: torsion points to isogenies

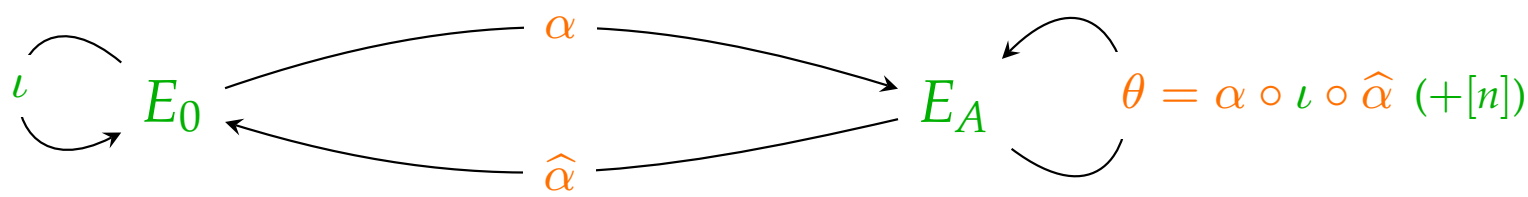
Finding the secret isogeny α of known degree, given $\alpha|_{E_0[B]}$.



- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.

Petit's trick: torsion points to isogenies

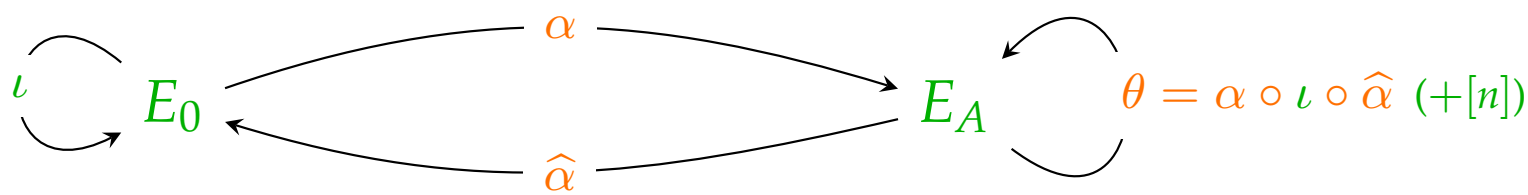
Finding the secret isogeny α of known degree, given $\alpha|_{E_0[B]}$.



- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.
- Know $\alpha|_{E_0[B]}$ (and $\widehat{\alpha}|_{E_A[B]}$) from public torsion points.

Petit's trick: torsion points to isogenies

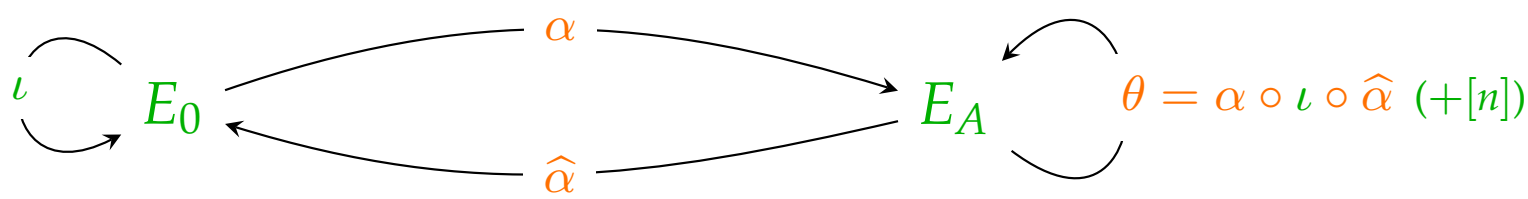
Finding the secret isogeny α of known degree, given $\alpha|_{E_0[B]}$.



- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.
- Know $\alpha|_{E_0[B]}$ (and $\widehat{\alpha}|_{E_A[B]}$) from public torsion points.
- Know $\deg(\theta) = \deg(\alpha)^2 \deg(\iota) + n^2$.

Petit's trick: torsion points to isogenies

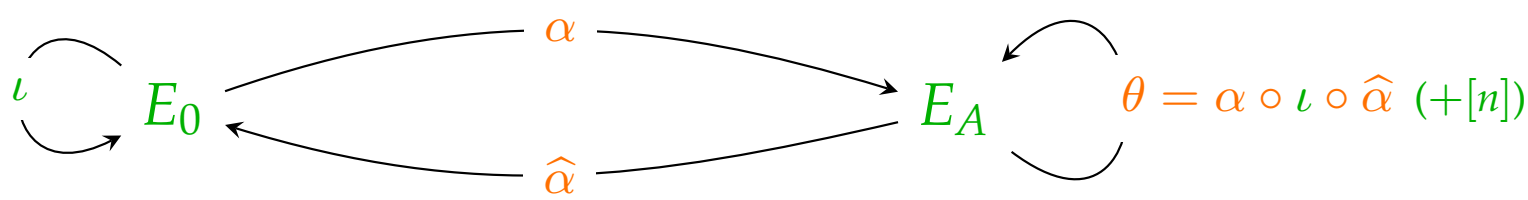
Finding the secret isogeny α of known degree, given $\alpha|_{E_0[B]}$.



- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.
- Know $\alpha|_{E_0[B]}$ (and $\widehat{\alpha}|_{E_A[B]}$) from public torsion points.
- Know $\deg(\theta) = \deg(\alpha)^2 \deg(\iota) + n^2$.
- Restriction # 2: If there exist ι, n such that $\deg(\theta) = B$, then can completely determine θ, and α, in polynomial-time.

Petit's trick: torsion points to isogenies

Finding the secret isogeny α of known degree, given $\alpha|_{E_0[B]}$.



- Restriction # 1: Assume we can choose $\iota : E_0 \to E_0$.
- Know $\alpha|_{E_0[B]}$ (and $\widehat{\alpha}|_{E_A[B]}$) from public torsion points.
- Know $\deg(\theta) = \deg(\alpha)^2 \deg(\iota) + n^2$.
- Restriction # 2: If there exist ι, n such that $\deg(\theta) = B$, then can completely determine θ, and α, in polynomial-time.
- Restriction # 2 rules out SIKE parameters, where $B \approx \deg(\alpha)$ (and $p \approx B \cdot \deg \alpha$).

Enter Kani

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given $\alpha|_{E_0[B]}$, compute α.*

Enter Kani

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given $\alpha|_{E_0[B]}$, compute α.*

Problem:
Not enough choices $\theta : E_A \to E_A$.
'No θ of degree B.'

Enter Kani

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given $\alpha|_{E_0[B]}$, compute α.*

Problem:
Not enough choices $\theta : E_A \to E_A$.
'No θ of degree B.'

Solution? $\theta : E_0 \times E_A \to E_0 \times E_A$?
$\rightsquigarrow$ still not enough.

Enter Kani

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given $\alpha|_{E_0[B]}$, compute α.*

Problem:
Not enough choices $\theta : E_A \to E_A$.
'No θ of degree B.'

Solution? $\theta : E_0 \times E_A \to E_0 \times E_A$?
$\rightsquigarrow$ still not enough. But!

Enter Kani

There are public elliptic curves E_0 and E_A, and a secret isogeny $\alpha : E_0 \to E_A$. Given $\alpha|_{E_0[B]}$, compute α.*

Problem:
Not enough choices $\theta : E_A \to E_A$.
'No θ of degree B.'

Solution? $\theta : E_0 \times E_A \to E_0 \times E_A$?
$\rightsquigarrow$ still not enough. But! Kani's theorem:

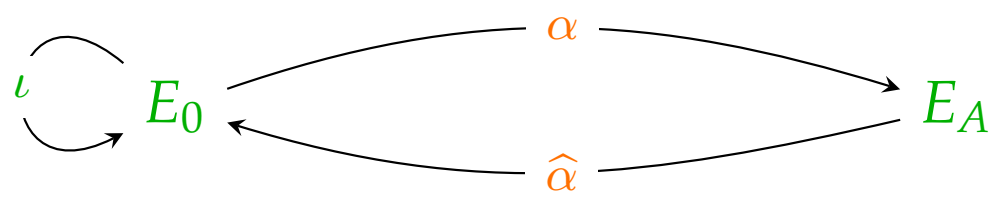
- Constructs E_1, E_2 such that there exists a (structure-preserving) isogeny

$$E_1 \times E_A \to E_0 \times E_2$$

  of the right degree, B^2.
- Petit's trick then applies.

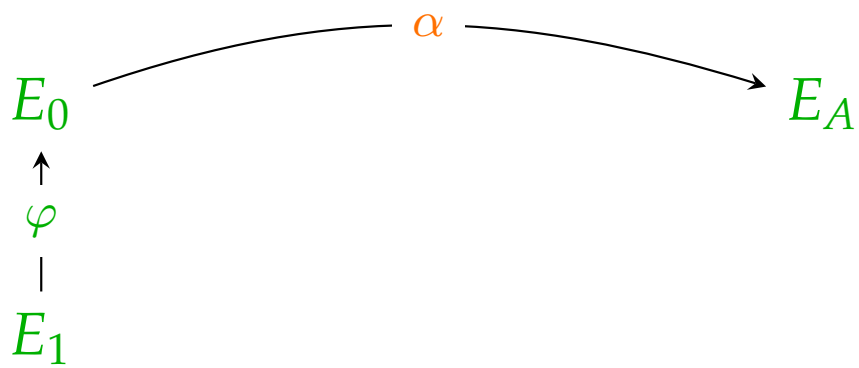
Recovering the secret

Finding the secret isogeny α of known degree given $\alpha|_{E_0[B]}$.



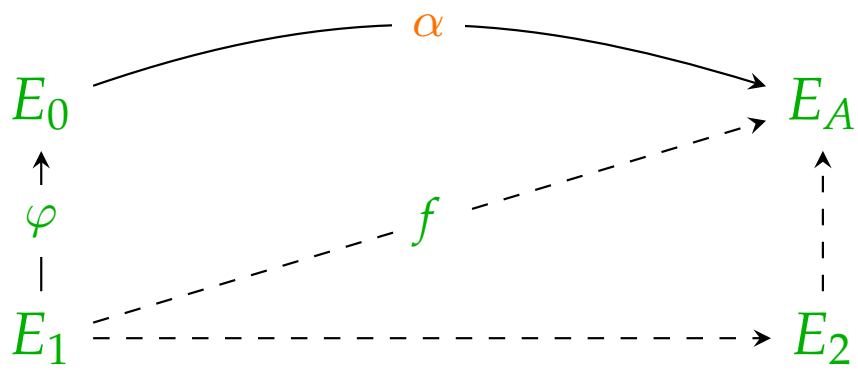
Recovering the secret

Finding the secret isogeny α of known degree given $\alpha|_{E_0[B]}$.



Recovering the secret

Finding the secret isogeny α of known degree given $\alpha|_{E_0[B]}$.



Kani's theorem constructs the above such that

$$\Phi = \begin{pmatrix} \varphi & -\widehat{\alpha} \\ * & * \end{pmatrix} : E_1 \times E_A \to E_0 \times E_2$$

is a structure preserving isogeny of degree B^2, and

$$\ker(\Phi) = \{(\deg(\alpha)P, f(P)) : P \in E_1[B]\}$$

$\rightsquigarrow$ can compute Φ and read off secret α!

Selected applications

- (Q)FESTA: New KEM. Fast and small as SIKE was?
- SQISign2D: Small, fast signatures with clean security reduction.

Selected applications

- (Q)FESTA: New KEM. Fast and small as SIKE was?
- SQISign2D: Small, fast signatures with clean security reduction.

Selected applications

- (Q)FESTA: New KEM. Fast and small as SIKE was?
- SQISign2D: Small, fast signatures with clean security reduction.

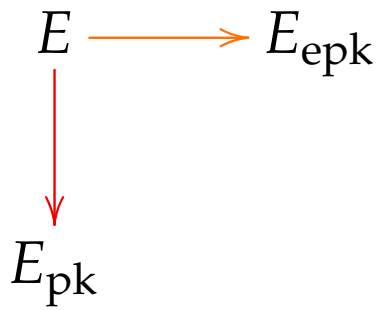
$$E$$

$$E_{\mathrm{pk}}$$

public, secret, eph. secret, public challenge, public proof

Selected applications

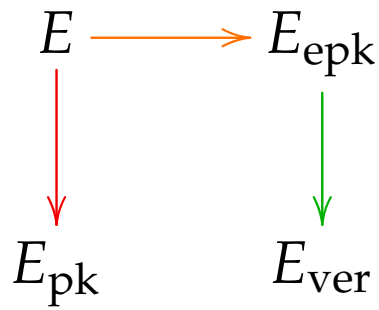
- (Q)FESTA: New KEM. Fast and small as SIKE was?
- SQISign2D: Small, fast signatures with clean security reduction.



public, secret, eph. secret, public challenge, public proof

Selected applications

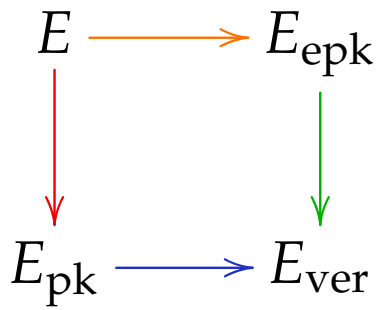
- (Q)FESTA: New KEM. Fast and small as SIKE was?
- SQISign2D: Small, fast signatures with clean security reduction.



public, secret, eph. secret, public challenge, public proof

Selected applications

- (Q)FESTA: New KEM. Fast and small as SIKE was?
- SQISign2D: Small, fast signatures with clean security reduction.

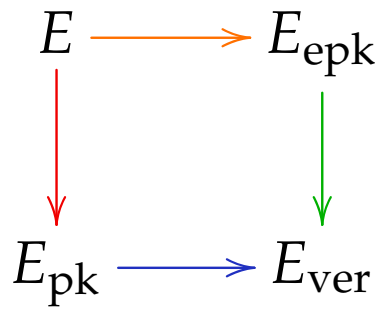


public, secret, eph. secret, public challenge, public proof

- Quaternion algorithms for $\to$ $\rightsquigarrow$ slow, ad hoc
- Kani $\rightsquigarrow$ faster, safer

Selected applications

- (Q)FESTA: New KEM. Fast and small as SIKE was?
- SQISign2D: Small, fast signatures with clean security reduction.



public, secret, eph. secret, public challenge, public proof

- Quaternion algorithms for $\to$ $\rightsquigarrow$ slow, ad hoc
- Kani $\rightsquigarrow$ faster, safer

Thank you!