RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

AN EXPOSURE MODEL FOR SUPERSINGULAR ISOGENY DIFFIE-HELLMAN KEY EXCHANGE

Brian Koziel, Reza Azarderakhsh, David Jao

Assistant Professor
Florida Atlantic University

Current PKC is safe until large-scale quantum computers are available

- ECDH, ECDSA: Protected by the Elliptic curve discrete logarithm problem

Current PKC is safe until large-scale quantum computers are available

- ECDH, ECDSA: Protected by the Elliptic curve discrete logarithm problem
- RSA: Protected by the factorization and discrete logarithm problems

Current PKC is safe until large-scale quantum computers are available

- ECDH, ECDSA: Protected by the Elliptic curve discrete logarithm problem
- RSA: Protected by the factorization and discrete logarithm problems
- Large-scale quantum computers with Shor's algorithm will BREAK the security assumptions for these primitives

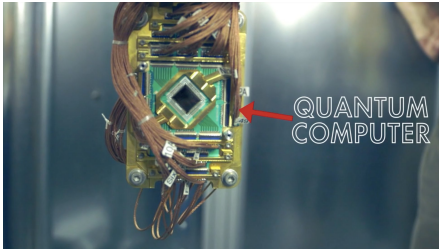


Figure: Quantum Computer (The Verge)

NIST has started a PQC standardization process

Primary Post-Quantum Cryptography (PQC) Candidates:

- Code-Based: McEliece

NIST has started a PQC standardization process

Primary Post-Quantum Cryptography (PQC) Candidates:

- Code-Based: McEliece
- Hash-based: Lamport, Merkle Signatures

NIST has started a PQC standardization process

Primary Post-Quantum Cryptography (PQC) Candidates:

- Code-Based: McEliece
- Hash-based: Lamport, Merkle Signatures
- Lattice-based: NTRU, LWE

NIST has started a PQC standardization process

Primary Post-Quantum Cryptography (PQC) Candidates:

- Code-Based: McEliece
- Hash-based: Lamport, Merkle Signatures
- Lattice-based: NTRU, LWE
- Multivariate: Rainbow Signature

NIST has started a PQC standardization process

Primary Post-Quantum Cryptography (PQC) Candidates:

- Code-Based: McEliece
- Hash-based: Lamport, Merkle Signatures
- Lattice-based: NTRU, LWE
- Multivariate: Rainbow Signature
- Isogeny-based: SIDH, SIKE

NIST has started a PQC standardization process

Primary Post-Quantum Cryptography (PQC) Candidates:

- Isogeny-based: SIDH, SIKE

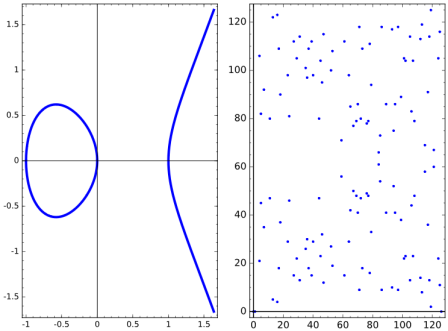


Figure: $E : y^2 = x^3 - x$ Left: $E/\mathbb{Z}$ Right: $E/\mathbb{F}_{127}$

SIDH offers the smallest key sizes of known quantum-resistant algorithms

Table: Comparison of different post-quantum key exchange and encryption algorithms at 128-bit quantum security level. Key sizes are in Bytes

Algorithm	NTRU	New Hope	McBits	SIDH	Compressed SIDH
Type	Lattice	Ring-LWE	Code	Isogeny	Isogeny
Public Key	6,130	2,048	1,046,739	576	336
Private Key	6,743	2,048	10,992	48	48
Performance	Slow	Very Fast	Slow	Very Slow	Very Slow

SIDH offers the smallest key sizes of known quantum-resistant algorithms

Table: Comparison of different post-quantum key exchange and encryption algorithms at 128-bit quantum security level. Key sizes are in Bytes

Algorithm	NTRU	New Hope	McBits	SIDH	Compressed SIDH
Type	Lattice	Ring-LWE	Code	Isogeny	Isogeny
Public Key	6,130	2,048	1,046,739	576	336
Private Key	6,743	2,048	10,992	48	48
Performance	Slow	Very Fast	Slow	Very Slow	Very Slow

- Small key sizes reduce transmission cost and storage requirement

Why would I want to use SIDH or SIKE as a quantum alternative?

Pros :)

- Very small public/private keys
- Implementations resemble ECC
- Security based on supersingular isogeny problem
- SIKE: IND-CCA KEM alternative to SIDH $\rightarrow$ static keys can be reused!
- No possibility for decryption error
- No complicated error distributions, rejection sampling, etc.
- Conservative security analysis when assuming generic attacks

Cons :(

- Newest candidate for PQC applications
- Very slow
- SIDH has security concerns if keys are reused

Isogeny-Based Cryptography History

- 1996: Couveignes - First mention of isogenies in cryptography. Published in 2006

Isogeny-Based Cryptography History

- 1996: Couveignes - First mention of isogenies in cryptography. Published in 2006
- 1999: Galbraith - First published cryptanalysis of isogeny problem

Isogeny-Based Cryptography History

- 1996: Couveignes - First mention of isogenies in cryptography. Published in 2006
- 1999: Galbraith - First published cryptanalysis of isogeny problem
- 2009: Charles et al. - Hash functions from supersingular isogenies

Isogeny-Based Cryptography History

- 1996: Couveignes - First mention of isogenies in cryptography. Published in 2006
- 1999: Galbraith - First published cryptanalysis of isogeny problem
- 2009: Charles et al. - Hash functions from supersingular isogenies
- 2010: Stolbunov - First published isogeny-based public-key cryptosystem based on isogenies between ordinary curves

Isogeny-Based Cryptography History

- 1996: Couveignes - First mention of isogenies in cryptography. Published in 2006
- 1999: Galbraith - First published cryptanalysis of isogeny problem
- 2009: Charles et al. - Hash functions from supersingular isogenies
- 2010: Stolbunov - First published isogeny-based public-key cryptosystem based on isogenies between ordinary curves
- 2010: Childs et al. - Quantum subexponential attack on Stolbunov's public-key cryptosystem

Isogeny-Based Cryptography History

- 1996: Couveignes - First mention of isogenies in cryptography. Published in 2006
- 1999: Galbraith - First published cryptanalysis of isogeny problem
- 2009: Charles et al. - Hash functions from supersingular isogenies
- 2010: Stolbunov - First published isogeny-based public-key cryptosystem based on isogenies between ordinary curves
- 2010: Childs et al. - Quantum subexponential attack on Stolbunov's public-key cryptosystem
- 2011: Jao and De Feo - Supersingular Isogeny Diffie-Hellman (SIDH) proposed

Isogeny-Based Cryptography History

- 1996: Couveignes - First mention of isogenies in cryptography. Published in 2006
- 1999: Galbraith - First published cryptanalysis of isogeny problem
- 2009: Charles et al. - Hash functions from supersingular isogenies
- 2010: Stolbunov - First published isogeny-based public-key cryptosystem based on isogenies between ordinary curves
- 2010: Childs et al. - Quantum subexponential attack on Stolbunov's public-key cryptosystem
- 2011: Jao and De Feo - Supersingular Isogeny Diffie-Hellman (SIDH) proposed
- 2016: Galbraith et al. - Active attack against SIDH with static key re-use

Isogeny-Based Cryptography History

- 1996: Couveignes - First mention of isogenies in cryptography. Published in 2006
- 1999: Galbraith - First published cryptanalysis of isogeny problem
- 2009: Charles et al. - Hash functions from supersingular isogenies
- 2010: Stolbunov - First published isogeny-based public-key cryptosystem based on isogenies between ordinary curves
- 2010: Childs et al. - Quantum subexponential attack on Stolbunov's public-key cryptosystem
- 2011: Jao and De Feo - Supersingular Isogeny Diffie-Hellman (SIDH) proposed
- 2016: Galbraith et al. - Active attack against SIDH with static key re-use
- 2017: Jao et al. - Supersingular Isogeny Key Encapsulation (SIKE) submitted to NIST PQC process

Isogeny-Based Cryptography underlying security

Consider two supersingular elliptic curves defined over a large prime extension field

Isogeny-Based Cryptography underlying security

Consider two supersingular elliptic curves defined over a large prime extension field

- $E_1/\mathbb{F}_{p^2}$ and $E_2/\mathbb{F}_{p^2}$, where p is a large prime

Isogeny-Based Cryptography underlying security

Consider two supersingular elliptic curves defined over a large prime extension field

- $E_1/\mathbb{F}_{p^2}$ and $E_2/\mathbb{F}_{p^2}$, where p is a large prime
- There exists some isogeny $\phi : E_1 \to E_2$ with a fixed, smooth degree that is public

Isogeny-Based Cryptography underlying security

Consider two supersingular elliptic curves defined over a large prime extension field

- $E_1/\mathbb{F}_{p^2}$ and $E_2/\mathbb{F}_{p^2}$, where p is a large prime
- There exists some isogeny $\phi : E_1 \to E_2$ with a fixed, smooth degree that is public

Supersingular Isogeny Problem

Given $P, Q \in E_1$ and $\phi(P), \phi(Q) \in E_2$, compute the secret isogeny, ϕ

Isogeny-Based Cryptography underlying security

Consider two supersingular elliptic curves defined over a large prime extension field

- $E_1/\mathbb{F}_{p^2}$ and $E_2/\mathbb{F}_{p^2}$, where p is a large prime
- There exists some isogeny $\phi : E_1 \to E_2$ with a fixed, smooth degree that is public

Supersingular Isogeny Problem

Given $P, Q \in E_1$ and $\phi(P), \phi(Q) \in E_2$, compute the secret isogeny, ϕ

- The best known attack is based on the claw finding algorithm

Isogeny-Based Cryptography underlying security

Consider two supersingular elliptic curves defined over a large prime extension field

- $E_1/\mathbb{F}_{p^2}$ and $E_2/\mathbb{F}_{p^2}$, where p is a large prime
- There exists some isogeny $\phi : E_1 \to E_2$ with a fixed, smooth degree that is public

Supersingular Isogeny Problem

Given $P, Q \in E_1$ and $\phi(P), \phi(Q) \in E_2$, compute the secret isogeny, ϕ

- The best known attack is based on the claw finding algorithm
- For SIDH/SIKE:

Isogeny-Based Cryptography underlying security

Consider two supersingular elliptic curves defined over a large prime extension field

- $E_1/\mathbb{F}_{p^2}$ and $E_2/\mathbb{F}_{p^2}$, where p is a large prime
- There exists some isogeny $\phi : E_1 \to E_2$ with a fixed, smooth degree that is public

Supersingular Isogeny Problem

Given $P, Q \in E_1$ and $\phi(P), \phi(Q) \in E_2$, compute the secret isogeny, ϕ

- The best known attack is based on the claw finding algorithm
- For SIDH/SIKE:
  - Classical attack $O(p^{1/4})$

Isogeny-Based Cryptography underlying security

Consider two supersingular elliptic curves defined over a large prime extension field

- $E_1/\mathbb{F}_{p^2}$ and $E_2/\mathbb{F}_{p^2}$, where p is a large prime
- There exists some isogeny $\phi : E_1 \to E_2$ with a fixed, smooth degree that is public

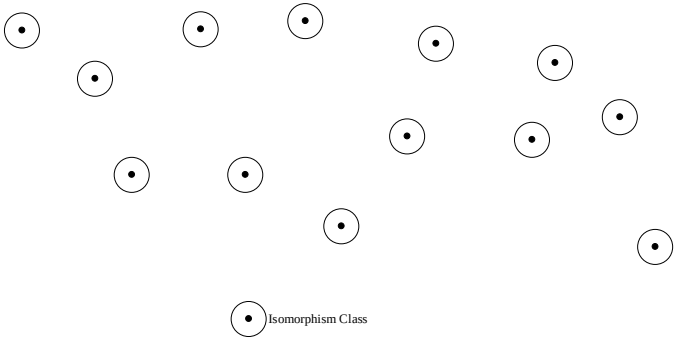
Supersingular Isogeny Problem

Given $P, Q \in E_1$ and $\phi(P), \phi(Q) \in E_2$, compute the secret isogeny, ϕ

- The best known attack is based on the claw finding algorithm
- For SIDH/SIKE:
  - Classical attack $O(p^{1/4})$
  - Quantum attack $O(p^{1/6})$

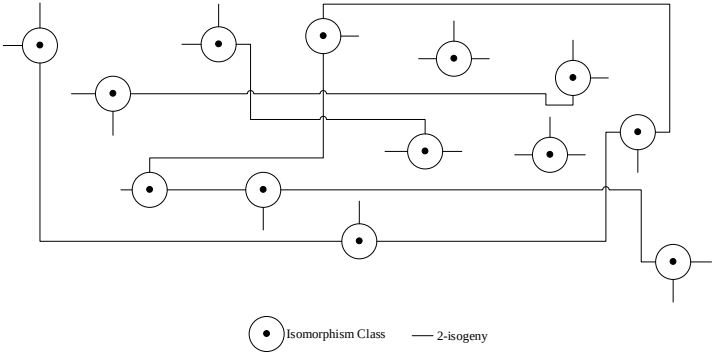
Visualizing the Supersingular Isogeny Problem

Consider a graph where each node represents supersingular isomorphism classes



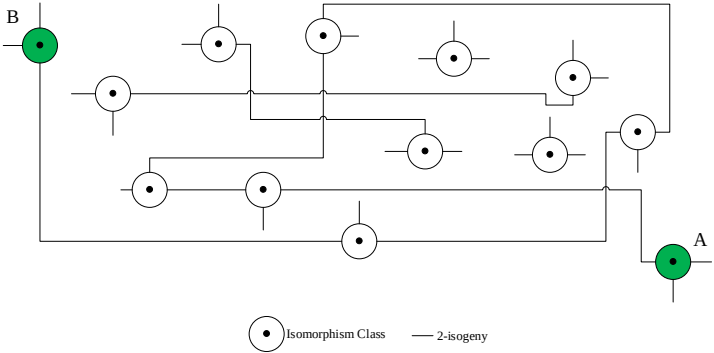
Visualizing the Supersingular Isogeny Problem

For $\ell = 2$, each node is connected by three unique 2-isogenies



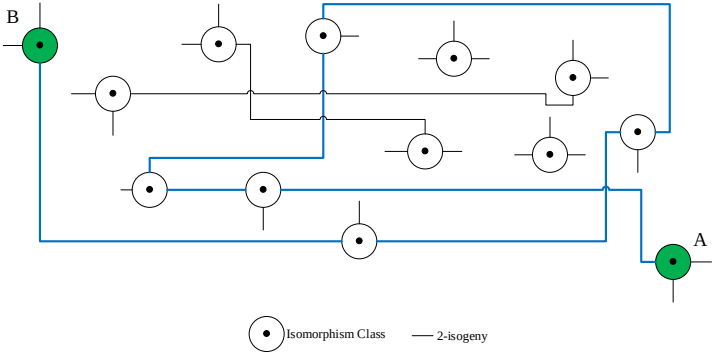
Visualizing the Supersingular Isogeny Problem

Consider finding an isogeny from Class A to Class B when $\ell = 2$



Visualizing the Supersingular Isogeny Problem

For large isogeny graphs (i.e., *p* is 512 bits or more), finding an isogeny path is HARD.



The SIDH protocol resembles standard Diffie-Hellman

Diffie-Hellman

Alice		Bob
A	g, p	B
$g^A \bmod p$		$g^B \bmod p$
$(g^B)^A \bmod p$		$(g^A)^B \bmod p$

$$g^{BA} \bmod p \equiv g^{AB} \bmod p$$

The SIDH protocol resembles standard Diffie-Hellman

Diffie-Hellman

Alice		Bob
A	g, p	B
$g^A \bmod p$		$g^B \bmod p$
$(g^B)^A \bmod p$		$(g^A)^B \bmod p$
$g^{BA} \bmod p$	$\equiv$	$g^{AB} \bmod p$

Supersingular Isogeny Diffie-Hellman

Alice		Bob
ϕ_A	$E_0/\mathbb{F}_{p^2}$	ϕ_B
$\phi_A : E_0 \to E_A$		$\phi_B : E_0 \to E_B$
$E_0/\langle A \rangle$		$E_0/\langle B \rangle$
$\phi'_A : E_B \to E_{BA}$		$\phi'_B : E_A \to E_{AB}$
$E_{BA} = E_0/\langle B, A \rangle$		$E_{AB} = E_0/\langle A, B \rangle$
$j(E_{BA}) \bmod p$	$\equiv$	$j(E_{AB}) \bmod p$

SIDH Protocol

$$E_0 \xrightarrow[E_0/\langle [m_A]P_A + [n_A]Q_A \rangle]{\phi_A} E_A = E_0/\langle A \rangle$$

SIDH Protocol

$$E_A = E_0/\langle A \rangle$$

$$\phi_A$$

$$E_0/\langle [m_A]P_A + [n_A]Q_A \rangle$$

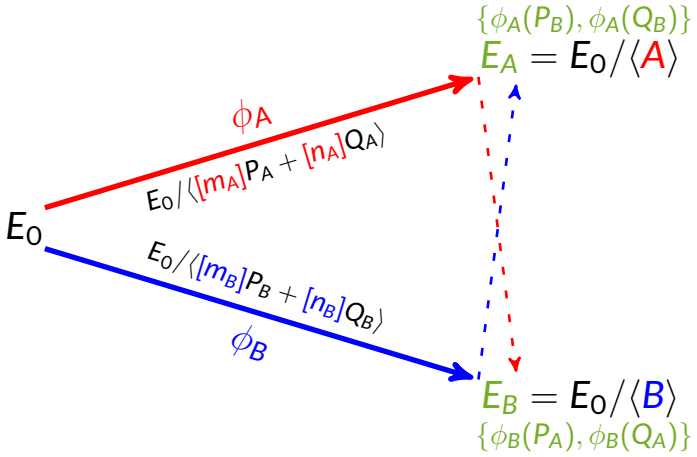
$$E_0$$

$$E_0/\langle [m_B]P_B + [n_B]Q_B \rangle$$

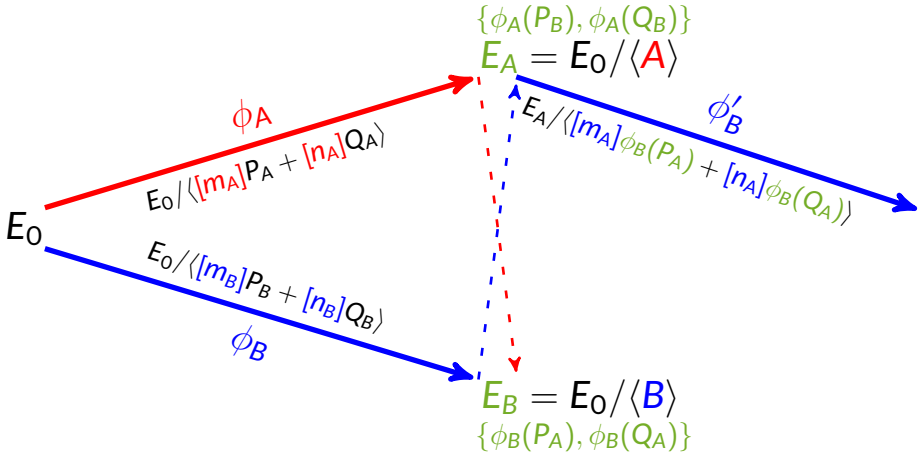
$$\phi_B$$

$$E_B = E_0/\langle B \rangle$$

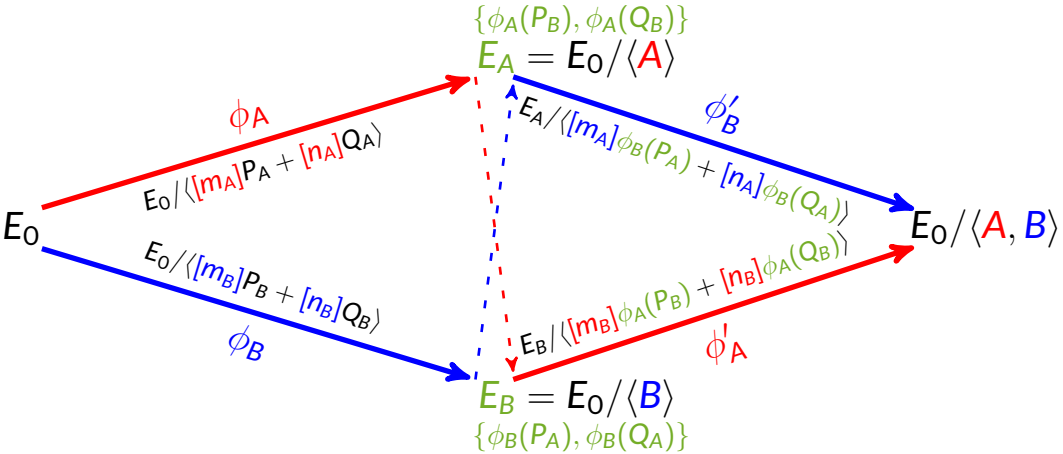
SIDH Protocol



SIDH Protocol



SIDH Protocol



SIDH Computations

Secret Kernel Generation

- Inputs:
  - Supersingular elliptic curve $E(\mathbb{F}_{p^2})$, torsion basis $\{P,Q\}$, private keys m, n
- Compute $R = \langle [m]P + [n]Q \rangle$

Large-Degree Isogeny

- Inputs:
  - Supersingular elliptic curve E, secret kernel point R
- Compute $\phi : E \to E/\langle R \rangle$ by iteratively computing isogenies

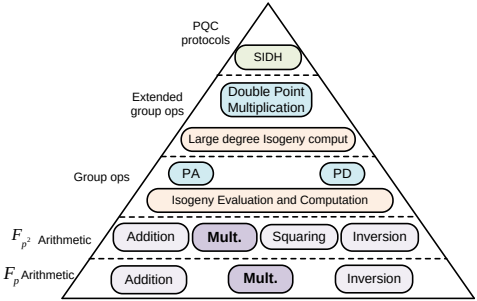
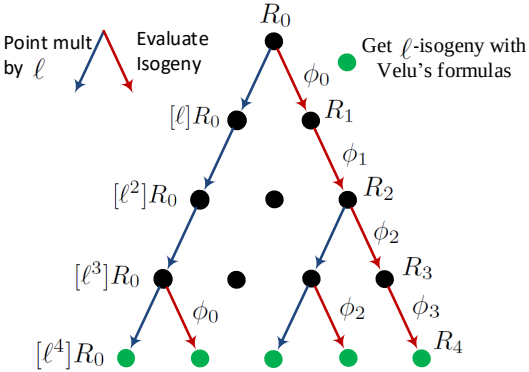


Figure: Breakdown of supersingular isogeny computations

Visualizing the large-degree isogeny computation

- Large-degree isogenies can be computed by iteratively computing small-degree isogenies
- Set $E_0 = E$ and $R_0 = ker(\phi)$
- Find kernel point $\ker(\phi_i) = \ell^{e-i-1}R_i$
- Compute ith isogeny $\phi_i : E_i \to E_i/\langle \ell^{e-i-1}R_i \rangle = E_{i+1}$
- Push kernel point to new curve $R_{i+1} = \phi_i(R_i)$



Creating an exposure model for SIDH

- Exposure Model $\rightarrow$ Assessing the security of a cryptosystem if certain pieces of information are divulged

Creating an exposure model for SIDH

- Exposure Model $\rightarrow$ Assessing the security of a cryptosystem if certain pieces of information are divulged
- Necessary to account for weak implementations or new attacks

Creating an exposure model for SIDH

- Exposure Model $\rightarrow$ Assessing the security of a cryptosystem if certain pieces of information are divulged
- Necessary to account for weak implementations or new attacks
- Looking specifically at the large-degree isogeny

Why might intermediate values be exposed?

- Poor implementation
- New attacks on large-degree isogeny
- Cache prime and probe
- Spectre and Meltdown
- Intermediate values not cleared
- Unexpected reset
- etc. etc.

What are the exposure classes?

- CLASS 1: Intermediate curve is exposed

What are the exposure classes?

- CLASS 1: Intermediate curve is exposed
  - An attacker can now split the large-degree isogeny into two separate isogenies

What are the exposure classes?

- CLASS 1: Intermediate curve is exposed
  - An attacker can now split the large-degree isogeny into two separate isogenies
  - Worst case scenario could cut the security assumption in half

What are the exposure classes?

- CLASS 1: Intermediate curve is exposed
  - An attacker can now split the large-degree isogeny into two separate isogenies
  - Worst case scenario could cut the security assumption in half
- CLASS 2: Intermediate kernel point is exposed

What are the exposure classes?

- CLASS 1: Intermediate curve is exposed
  - An attacker can now split the large-degree isogeny into two separate isogenies
  - Worst case scenario could cut the security assumption in half
- CLASS 2: Intermediate kernel point is exposed
  - Some variant of the secret kernel point is leaked

What are the exposure classes?

- CLASS 1: Intermediate curve is exposed
  - An attacker can now split the large-degree isogeny into two separate isogenies
  - Worst case scenario could cut the security assumption in half
- CLASS 2: Intermediate kernel point is exposed
  - Some variant of the secret kernel point is leaked
  - If corresponding curve can be found, then remaining isogenies can be computed $\rightarrow$ this is very bad

What are the exposure classes?

- CLASS 1: Intermediate curve is exposed
  - An attacker can now split the large-degree isogeny into two separate isogenies
  - Worst case scenario could cut the security assumption in half
- CLASS 2: Intermediate kernel point is exposed
  - Some variant of the secret kernel point is leaked
  - If corresponding curve can be found, then remaining isogenies can be computed $\rightarrow$ this is very bad
- CLASS 3: Intermediate basis point is exposed

What are the exposure classes?

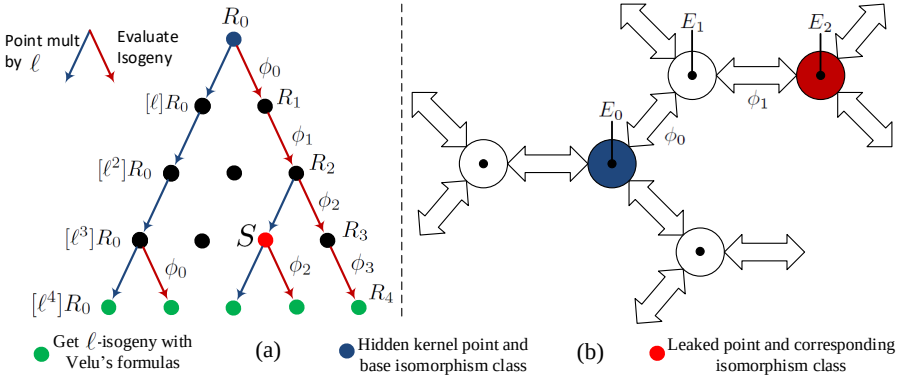
- CLASS 1: Intermediate curve is exposed
  - An attacker can now split the large-degree isogeny into two separate isogenies
  - Worst case scenario could cut the security assumption in half
- CLASS 2: Intermediate kernel point is exposed
  - Some variant of the secret kernel point is leaked
  - If corresponding curve can be found, then remaining isogenies can be computed $\rightarrow$ this is very bad
- CLASS 3: Intermediate basis point is exposed
  - If corresponding curve is found, then isogeny decisions are revealed

What are the exposure classes?

- CLASS 1: Intermediate curve is exposed
  - An attacker can now split the large-degree isogeny into two separate isogenies
  - Worst case scenario could cut the security assumption in half
- CLASS 2: Intermediate kernel point is exposed
  - Some variant of the secret kernel point is leaked
  - If corresponding curve can be found, then remaining isogenies can be computed $\rightarrow$ this is very bad
- CLASS 3: Intermediate basis point is exposed
  - If corresponding curve is found, then isogeny decisions are revealed
  - After the corresponding curve has been found, this situation resembles the intermediate curve scenario

Visualization of an exposed kernel point

- With an exposed kernel point (CLASS 2), an attacker can find the corresponding curve and compute the remaining isogenies that compose the secret isogeny



An exposed kernel point is a disaster as it can be used to retrieve private keys

General attack procedure for point after k ℓ-isogenies and j point multiplications by ℓ
Intermediate kernel point is of the form $S = \phi_{k-1:0}([\ell^j m]P + [\ell^j n]Q)$ on curve E_k (CLASS 2). Original secret kernel point is of the form $R = [m]P + [n]Q$

An exposed kernel point is a disaster as it can be used to retrieve private keys

General attack procedure for point after k ℓ-isogenies and j point multiplications by ℓ
Intermediate kernel point is of the form $S = \phi_{k-1:0}([\ell^j m]P + [\ell^j n]Q)$ on curve E_k (CLASS 2). Original secret kernel point is of the form $R = [m]P + [n]Q$

1. Find isogenous curve E_k to find first k isogenies (difficulty $O(\ell^k)$)

An exposed kernel point is a disaster as it can be used to retrieve private keys

General attack procedure for point after k ℓ-isogenies and j point multiplications by ℓ
Intermediate kernel point is of the form $S = \phi_{k-1:0}([\ell^j m]P + [\ell^j n]Q)$ on curve E_k (CLASS 2). Original secret kernel point is of the form $R = [m]P + [n]Q$

1. Find isogenous curve E_k to find first k isogenies (difficulty $O(\ell^k)$)
2. Push torsion basis through k isogenies
   - $\{\phi_{k-1:0}(P), \phi_{k-1:0}(Q)\}$

An exposed kernel point is a disaster as it can be used to retrieve private keys

General attack procedure for point after k ℓ-isogenies and j point multiplications by ℓ
Intermediate kernel point is of the form $S = \phi_{k-1:0}([\ell^j m]P + [\ell^j n]Q)$ on curve E_k (CLASS 2). Original secret kernel point is of the form $R = [m]P + [n]Q$

1. Find isogenous curve E_k to find first k isogenies (difficulty $O(\ell^k)$)
2. Push torsion basis through k isogenies
   - $\{\phi_{k-1:0}(P), \phi_{k-1:0}(Q)\}$
3. Perform generalized elliptic curve discrete log (simple for SIDH curves) $\rightarrow$ result is $k - j$ bits of private key
   - $\phi_{k-1:0}([\ell^j m]P + [\ell^j n]Q) = \phi_{k-1:0}([m']P) + \phi_{k-1:0}([n']Q)$

An exposed kernel point is a disaster as it can be used to retrieve private keys

General attack procedure for point after k ℓ-isogenies and j point multiplications by ℓ
Intermediate kernel point is of the form $S = \phi_{k-1:0}([\ell^j m]P + [\ell^j n]Q)$ on curve E_k (CLASS 2). Original secret kernel point is of the form $R = [m]P + [n]Q$

1. Find isogenous curve E_k to find first k isogenies (difficulty $O(\ell^k)$)
2. Push torsion basis through k isogenies
   - $\{\phi_{k-1:0}(P), \phi_{k-1:0}(Q)\}$
3. Perform generalized elliptic curve discrete log (simple for SIDH curves) $\rightarrow$ result is $k - j$ bits of private key
   - $\phi_{k-1:0}([\ell^j m]P + [\ell^j n]Q) = \phi_{k-1:0}([m']P) + \phi_{k-1:0}([n']Q)$
4. Perform exhaustive search on the point multiples j for the rest of the key (difficulty $O(\ell^j)$)
   - Remaining secret key bits is some secret multiple of the point order: $m' \equiv x \times m \bmod \ell^j$

A random pre-isogeny isomorphism protects against all but an exposed curve

- Vélu's formulas for isogenies are deterministic for a given kernel point and curve

A random pre-isogeny isomorphism protects against all but an exposed curve

- Vélu's formulas for isogenies are deterministic for a given kernel point and curve
- An isomorphism moves from one curve to another within an isomorphism class

A random pre-isogeny isomorphism protects against all but an exposed curve

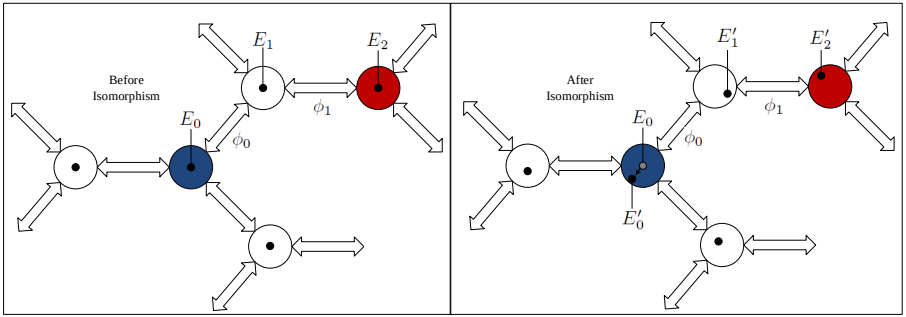
- Vélu's formulas for isogenies are deterministic for a given kernel point and curve
- An isomorphism moves from one curve to another within an isomorphism class
- A random isomorphism will scale the curve

A random pre-isogeny isomorphism protects against all but an exposed curve

- Vélu's formulas for isogenies are deterministic for a given kernel point and curve
- An isomorphism moves from one curve to another within an isomorphism class
- A random isomorphism will scale the curve
  - This scaling changes the output curves of isogenies and obfuscates any points that are exposed

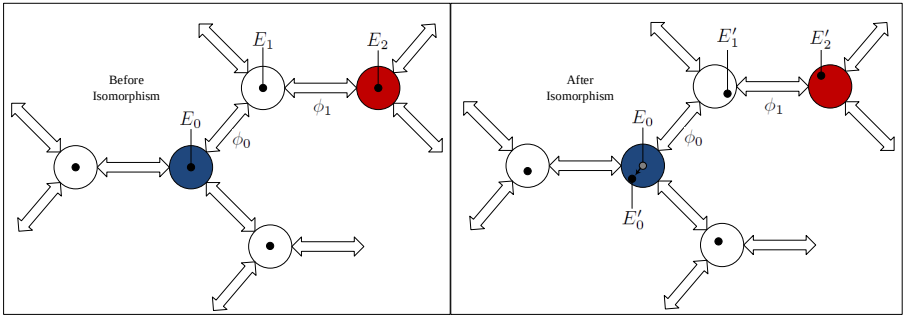
Visualization of a pre-isogeny isomorphism

- The pre-isogeny isomorphism obfuscates any exposed points throughout the isogeny operation



Visualization of a pre-isogeny isomorphism

- The pre-isogeny isomorphism obfuscates any exposed points throughout the isogeny operation
- Protects against CLASS 2 and CLASS 3 exposures



A pre-isogeny isomorphism is a computationally cheap countermeasure

- For short Weierstrass curves, this requires a random number and several field operations

A pre-isogeny isomorphism is a computationally cheap countermeasure

- For short Weierstrass curves, this requires a random number and several field operations
- Major cost is generating random numbers

Table: Cost of Pre-isogeny Isomorphism

Protocol	r	δ	I	M	S
SIDH Round 1	1	0	1	9	3
SIDH Round 2	1	0	1	5	3
SIDH Indirect Key Validation (Kirkwood et al. Validiation)	1	4	1	15	5

- Let r be the cost to generate a random number, I be a finite-field inversion, M be a finite-field multiplication, S be a finite-field squaring, and δ be a finite-field comparison

Conclusion

- SIDH/SIKE are up and coming candidates for PQC standardization

Conclusion

- SIDH/SIKE are up and coming candidates for PQC standardization
  - These primitives feature the smallest known public key sizes for quantum-resistant PKC

Conclusion

- SIDH/SIKE are up and coming candidates for PQC standardization
  - These primitives feature the smallest known public key sizes for quantum-resistant PKC
- We created an exposure model for the large-degree isogeny computation

Conclusion

- SIDH/SIKE are up and coming candidates for PQC standardization
  - These primitives feature the smallest known public key sizes for quantum-resistant PKC
- We created an exposure model for the large-degree isogeny computation
- We showed that an exposed intermediate kernel point (CLASS 2) can reveal a party's private key

Conclusion

- SIDH/SIKE are up and coming candidates for PQC standardization
  - These primitives feature the smallest known public key sizes for quantum-resistant PKC
- We created an exposure model for the large-degree isogeny computation
- We showed that an exposed intermediate kernel point (CLASS 2) can reveal a party's private key
- We proposed a pre-isogeny isomorphism as a cheap countermeasure to protect against exposing intermediate points in the future

Conclusion

- SIDH/SIKE are up and coming candidates for PQC standardization
  - These primitives feature the smallest known public key sizes for quantum-resistant PKC
- We created an exposure model for the large-degree isogeny computation
- We showed that an exposed intermediate kernel point (CLASS 2) can reveal a party's private key
- We proposed a pre-isogeny isomorphism as a cheap countermeasure to protect against exposing intermediate points in the future
- Thank you very much for your attention. Questions?