RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: CRYP-F02

CODES AND ISOGENIES

David Jao

Professor
University of Waterloo

RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: CRYP-F02

REVOCABLE IDENTITY-BASED ENCRYPTION FROM CODES WITH RANK METRIC

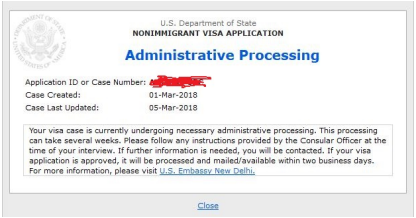
Somitra Kumar Sanadhya

Associate Professor
CSE, IIT Ropar, India

Joint work with Donghoon Chang (IIIT Delhi), Amit Kumar Chauhan (IIT Ropar), and Sandeep Kumar (IIIT Delhi & University of Delhi, India)

Date: 04/20/2018 Session: CRYP-F02 Time: 10:15 AM - 11:00 AM

Thanks to US Department of State for not providing Nonimmigrant Visa to Dr. Somitra Kumar Sanadhya.


U.S. Department of State
NONIMMIGRANT VISA APPLICATION

Administrative Processing

Application ID or Case Number: [redacted]
Case Created: 01-Mar-2018
Case Last Updated: 05-Mar-2018

Your visa case is currently undergoing necessary administrative processing. This processing can take several weeks. Please follow any instructions provided by the Consular Officer at the time of your interview. If further information is needed, you will be contacted. If your visa application is approved, it will be processed and mailed/available within two business days. For more information, please visit U.S. Embassy New Delhi.

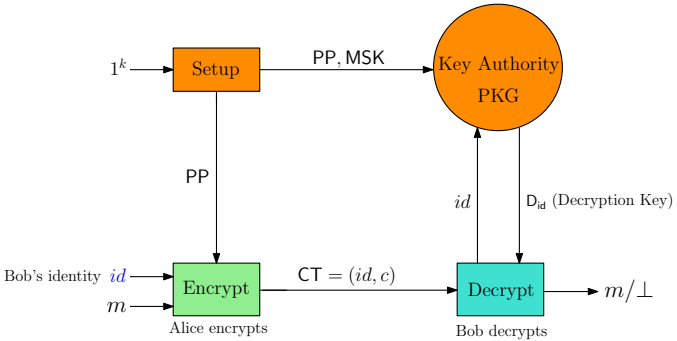
Close

Dr. Reza Azarderakhsh kindly agrees to present our work at CT-RSA 2018.

We would like to thank Dr. Reza Azarderakhsh for accepting our request.

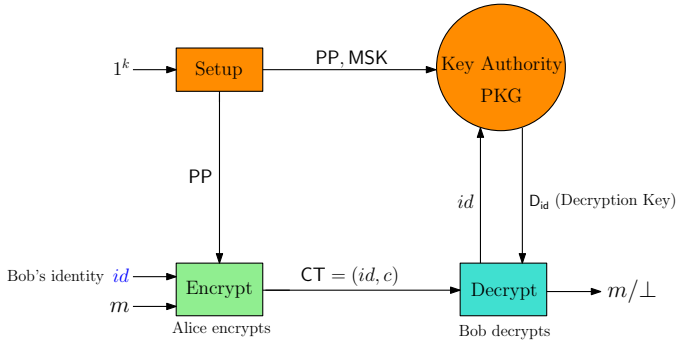
Motivation

Identity-based Crypto: To resolve the problem of key management in standard PKE.



Motivation

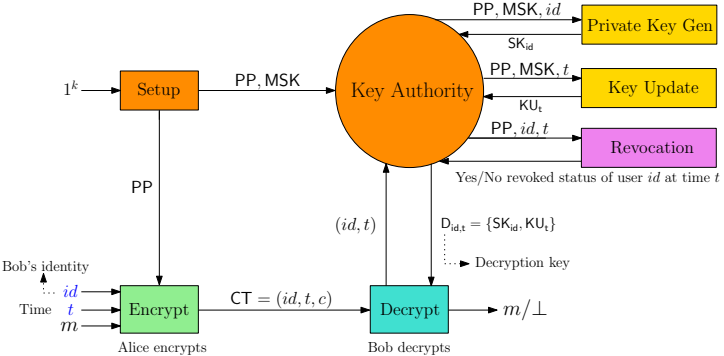
Identity-based Crypto: To resolve the problem of key management in standard PKE.



Problem: how to efficiently revoke past users in identity-based encryption (IBE) ?

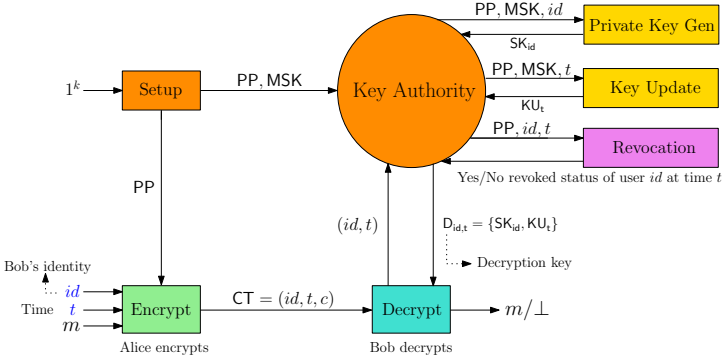
Revocable IBE (RIBE)

Adding efficient revocation procedure to IBE …



Revocable IBE (RIBE)

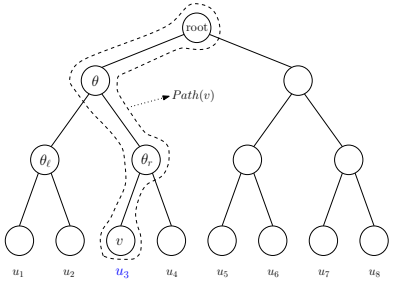
Adding efficient revocation procedure to IBE …



- *Initially, the key update is public to all users. At a later time, stop publishing the key updates for a revoked user !*

Key Generation on a Binary Tree

- Upon registration, the key authority provides the user u_3 with a set of distinct private keys for each node in Path(v) on a binary tree BT.



- To generate the key for a user u, one can choose a random value r
  - split r into two parts r_1 and r_2 for each node
  - generate private-key SK_{id} on (id, r_1) using *private key generation* algorithm
  - do key-updates KU_t on (t, r_2) using *key-updates* algorithm
- The decryption-key is: $\mathsf{DK}_{id,t} = \{\mathsf{SK}_{id}, \mathsf{KU}_t\}$ for a node on a path corresponding to a user u.

How to revoke a user ?

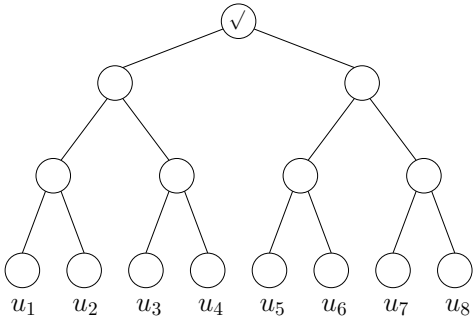
- Run the Revocation algorithm $\mathcal{R}(id^*, t^*, \mathrm{RL})$:
  - adds revoked user id^* to the revocation list RL at revoked time t^*.
- Do the key-updates for non-revoked users (the number of key-updates are logarithmic in the number of users).
- Stop sending key-updates for a revoked user for the time $t \geq t^*$.

How to do Key-Updates for a user ?

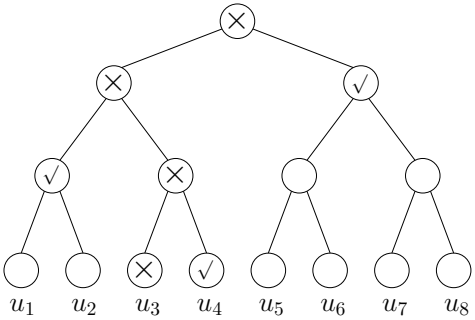
- Run the Key-Update-Nodes algorithm KUNodes(BT, RL, t) :
  - returns a minimal set $Y \subset$ BT of nodes for which key update needs to be published.

KUNodes(BT, RL, t)
$X, Y \leftarrow \phi$
$\forall(v_i, t_i) \in$ RL
if $t_i \leq t$ *then add* Path(v_i) *to* X
$\forall \theta \in X$
if $\theta_\ell \notin X$ *then add* θ_ℓ *to* Y
if $\theta_r \notin X$ *then add* θ_r *to* Y
If $Y = \phi$ *then add* root *to* Y
Return Y

Examples of Key-Updates Algorithm



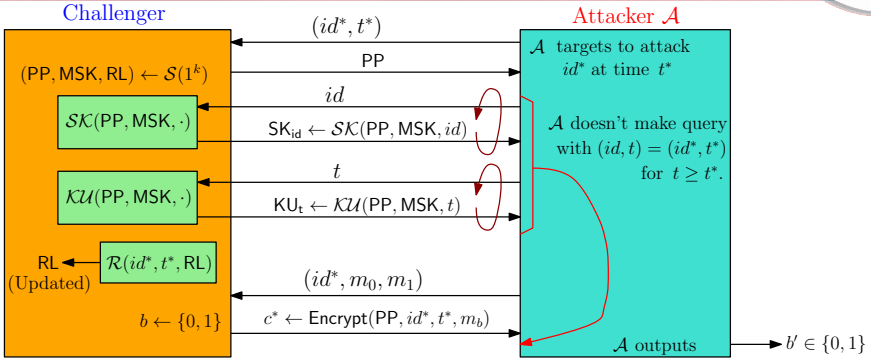
(a) No user is revoked



(b) User u_3 is revoked

✓ shows that the key-updates KU_t is published for those nodes at time t.

Selective-Revocable-ID Security of RIBE



The advantage of $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\mathsf{ind\text{-}srid\text{-}cpa}}_{\mathcal{A},\mathsf{RIBE}}(k) \quad := \quad \left| \Pr[b' = b] - \frac{1}{2} \right| .$$

A Bit of (R)IBE History

- Boneh and Franklin : *Identity-based Encryption from Weil Pairing*, CRYPTO 2001.
- Boldyreva, Goyal and Kumar : Identity-based Encryption (from Weil pairing) with Efficient Revocation, ACM CCS 2008.
- Agrawal, Boneh and Boyen : *Efficient Lattice (H)IBE in the Standard Model*, EUROCRYPT 2010.
- Chen, Lim, Ling, Wang and Nguyen : *Revocable Identity-Based Encryption from Lattices*, ACISP 2012.
- Gaborit, Hauteville, Phan and Tillich : *Identity-based Encryption from Codes with Rank Metric*, CRYPTO 2017.

Our Work

- We construct Revocable IBE (RIBE) from Codes with Rank Metric.
- Highlights of RIBE :
  - Constructed from Low Rank Parity-Check (LRPC) codes.
  - Master Secret Key (MSK) is defined as the “Trapdoor” generated through the RankSign algorithm[1].
  - Provides IND-sRID-CPA security which relies mainly on the Rank Syndrome Decoding (RSD) problem[1].
  - Binary-Tree data structure[2] is used to add efficient revocation procedure to IBE.

[1]Gaborit et al. *RankSign: An Efficient Signature Algorithm based on the Rank Metric*, PQCrypto 2014.
[2]Boldyreva, Goyal, and Kumar. *Identity-based Encryption with Efficient Revocation*, ACM CCS 2008.

Rank Metric over $\mathbb{F}_{q^m}^n$

- Let $\mathbb{F}_{q^m}$ be a m-dimensional vector space over the field $\mathbb{F}_q$ with prime q.
- Let $\boldsymbol{B} = (b_1, \ldots, b_m)$ be a basis of $\mathbb{F}_{q^m}$.
- Let $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$, then each coordinate x_j can be expressed in terms of basis $\boldsymbol{B}$ as :

$$x_j = \sum_{i=1}^{m} m_{ij}\, b_i \qquad \text{where } m_{ij} \in \mathbb{F}_q$$

- The $m \times n$ matrix associated with $\boldsymbol{x}$ is defined as $\boldsymbol{M}(\boldsymbol{x}) = (m_{ij})_{1 \le i \le m,\, 1 \le j \le n}$.
- *Rank weight* of $\boldsymbol{x}$ is defined as

$$\|\boldsymbol{x}\| = \mathit{Rank}\ \boldsymbol{M}(\boldsymbol{x})$$

- *Rank distance (metric)* $d(\boldsymbol{x}, \boldsymbol{y})$ between elements $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{F}_{q^m}^n$ is defined as

$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|$$

Rank Code

- A linear code $\mathcal{C}$ of dimension k and length n is a subspace of dimension k of $\mathbb{F}_{q^m}^n$ embedded with the rank metric.
- $\mathcal{C}$ can be represented by two ways :
  - by a generator matrix $\boldsymbol{G} \in \mathbb{F}_{q^m}^{k \times n}$. Each rows of $\boldsymbol{G}$ is an element of a basis of $\mathcal{C}$,

$$\mathcal{C} = \{\boldsymbol{x}\boldsymbol{G} \mid \boldsymbol{x} \in \mathbb{F}_{q^m}^k\}$$

  - by a parity-check matrix $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k) \times n}$. Each rows of $\boldsymbol{H}$ determines a parity-check equation verified by its elements of $\mathcal{C}$:

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{F}_{q^m}^n \mid \boldsymbol{H}\boldsymbol{x}^T = \boldsymbol{0}\}$$

Low Rank Parity-Check (LRPC) Codes

LRPC Codes

- Let $\boldsymbol{H} = (h_{ij})_{1 \le i \le n-k,\, 1 \le j \le n} \in \mathbb{F}_{q^m}^{(n-k) \times n}$ be a full rank matrix.
- All the elements of $\boldsymbol{H}$ generate an $\mathbb{F}_q$-subspace F of dimension d:

$$F = \langle h_{ij} \rangle_{\mathbb{F}_q}$$

- The code $\mathcal{C}[n, k, d]_{q^m}$ with parity-check matrix $\boldsymbol{H}$ is called a LRPC code of weight d.

Such a matrix $\boldsymbol{H}$ is called a homogeneous matrix of weight d.

Augmented Low Rank Parity-Check Codes

Augmented LRPC Codes

- Let $\boldsymbol{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a homogeneous matrix of full rank and of weight d.
- $\boldsymbol{R} \in \mathbb{F}_{q^m}^{(n-k)\times \ell}$ be a random matrix.
- $\boldsymbol{P} \in GL_{n-k}(\mathbb{F}_{q^m})$ and $Q \in GL_{n+\ell}(\mathbb{F}_q)$ be two invertible matrices.
- $\boldsymbol{H'} = \boldsymbol{P}(\boldsymbol{R}|\boldsymbol{H})\boldsymbol{Q}$ be the parity-check matrix of a code $\mathcal{C}$ of type $[n+\ell, k+\ell]$.

We call such a code, an LRPC$^+$ code. If $\ell = 0$, then $\mathcal{C}$ is a LRPC code.

Definition (LRPC$^+$ Problem)

Given an LRPC$^+$ code, distinguish it from a random code with the same parameters.

Rank Syndrome Decoding Problem

Definition (Rank Syndrome Decoding (RSD) Problem)

Instance : a parity-check matrix $\boldsymbol{H}$ in $\mathbb{F}_{q^m}^{(n-k)\times n}$, a syndrome $\boldsymbol{s}$ in $\mathbb{F}_{q^m}^{n-k}$ and an integer w.

Question : does there exist $\boldsymbol{x} \in \mathbb{F}_{q^m}^{n}$ such that $\boldsymbol{H}.\boldsymbol{x}^{\mathsf{T}} = \boldsymbol{s}$ and $w_R(\boldsymbol{x}) \leq w$?

Definition (Decisional Rank Syndrome Decoding (DRSD) Problem)

Instance : a generator matrix $\boldsymbol{G}$ in $\mathbb{F}_{q^m}^{k\times n}$, $\boldsymbol{m} \in \mathbb{F}_{q^m}^{k}$ and $\boldsymbol{x} \in \mathbb{F}_{q^m}^{n}$ of weight w.

Question : can we distinguish the pair $(\boldsymbol{G}, \boldsymbol{mG} + \boldsymbol{x})$ from $(\boldsymbol{G}, \boldsymbol{y})$ with $\boldsymbol{y} \xleftarrow{\$} \mathbb{F}_{q^m}^{n}$?

Rank Support Learning Problem

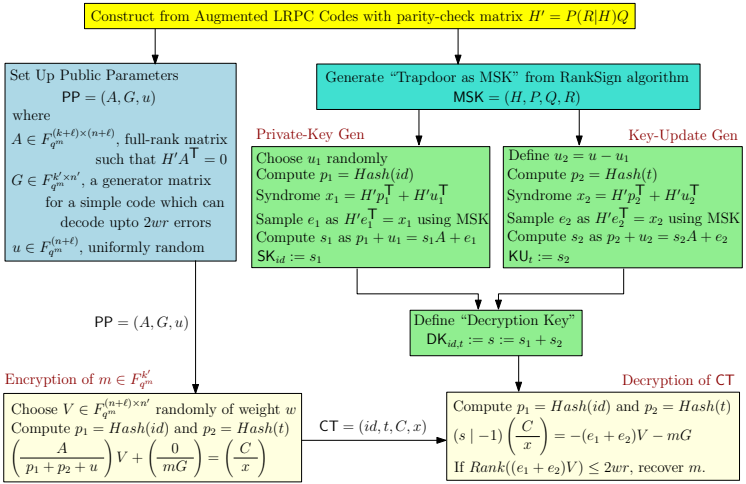
Definition (Rank Support Learning (RSL) Problem)

- Let $\boldsymbol{A}$ be a random full-rank matrix of size $(n-k) \times n$ over $\mathbb{F}_{q^m}$ and U be a subspace of $\mathbb{F}_{q^m}$ of dimension w.
- Let $\mathcal{O}$ be an oracle which gives samples of the form $(\boldsymbol{A}, \boldsymbol{Av})$, where $\boldsymbol{v} \xleftarrow{\$} U^n$.
- The RSL **problem** is to recover $\boldsymbol{V}$ given only access to the oracle. That is,

$$\Pr[\mathcal{A}(\boldsymbol{A}, \boldsymbol{AV}) = \boldsymbol{V}] \leq \epsilon, \quad \boldsymbol{V} \xleftarrow{\$} U^{n \times N}$$

  where N be the number of queries made to the oracle $\mathcal{O}$.
- The DRSL **problem** is to distinguish $(\boldsymbol{A}, \boldsymbol{AV})$ from $(\boldsymbol{A}, \boldsymbol{Y})$ with $\boldsymbol{Y} \xleftarrow{\$} \mathbb{F}_{q^m}^{(n-k) \times N}$.

Overview of RIBE from LRPC Codes



Security of RIBE

Theorem

Suppose there exists an adversary $\mathcal{A}$ against the IND-sRID-CPA security, who makes at most q_{H_1} and q_{H_2} distinct queries to the H_1 and H_2 random oracles, then the advantage of adversary $\mathcal{A}$ is given by the following expression

$$\epsilon_{\mathsf{ribe}} \leq \left(q_{H_1} + q_{H_2}\right) \cdot \left(\frac{2}{q} + \epsilon_{\mathsf{drsd}}\right) + \epsilon_{\mathsf{lrpc}^+} + \epsilon_{\mathsf{drsl}},$$

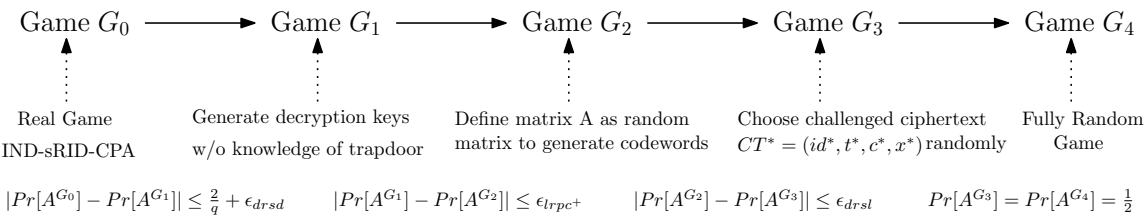
where ϵ_{ribe}, ϵ_{drsd}, ϵ_{drsl} and $\epsilon_{\mathsf{lrpc}^+}$ are respectively the bound on the advantage of the attacks against the RIBE system, the DRSD, DRSL and LRPC$^+$ problems.

Proof Sketch (Selective-Revocable-ID security of RIBE)

Suppose $\mathcal{A}$ be a probabilistic polynomial time adversary who wishes to break the security of RIBE in selective-revocable-ID security.

Goal : To show that a PPT adversary $\mathcal{A}$ has a negligible advantage in winning the original IND-sRID-CPA game.

Transitions of Games:



Practical set of Parameters

We choose the parameters of the simple code in such a way that it can decode up to $2wr$ errors and the decoding error with failure probability $\approx \frac{1}{q^{\ell'-2wr+1}}$ is small.

Scheme	n	$n-k$	m	q	d	ℓ	r	d_{GV}	d_{sign}
IBE[3]	100	20	96	2^{192}	5	12	16	11	20
RIBE	100	20	96	2^{192}	5	12	16	11	20

Scheme	Public Key Size (Bytes)	n'	k'	ℓ'	w
IBE[5]	4,239,360 of $\boldsymbol{A}$	96	9	66	4
RIBE	4,497,408 of $(\boldsymbol{A}, \boldsymbol{u})$	96	9	66	2

With the above parameters, one can achieve decoding failure probability $\approx 2^{-576}$.

[3] Gaborit et al.: *Identity-based Encryption from Codes with Rank Metric*, CRYPTO 2017.

Conclusion and Future Work

- We built the revocable IBE from codes with rank metric.
- We formally proved the selective-ID CPA security of revocable IBE in the random oracle model.

- As an extension of this work, one might think of constructing Revocable IBE with adaptive-ID CCA security, which is a strongest security model.
- Another extension could be a scheme secure in standard model.

THANK YOU
FOR YOUR
ATTENTION

RSA Conference2018
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

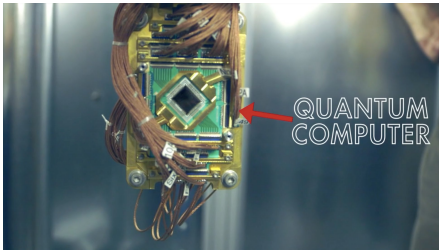


Figure: Quantum Computer (The Verge)

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

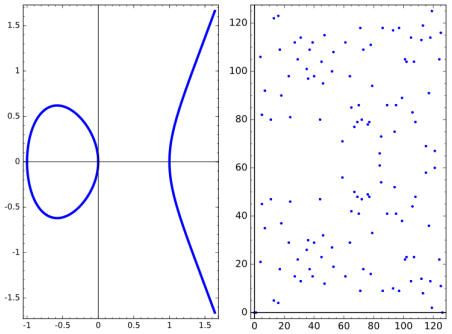


Figure: $E : y^2 = x^3 - x$ Left: $E/\mathbb{Z}$ Right: $E/\mathbb{F}_{127}$

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 $\phi : E_1 \rightarrow E_2$ 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 $\phi : E_1 \rightarrow E_2$ 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 $\phi : E_1 \rightarrow E_2$ 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 $\phi : E_1 \rightarrow E_2$ 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 $\phi : E_1 \rightarrow E_2$ 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 $\phi : E_1 \rightarrow E_2$ with a fixed, smooth degree that is public

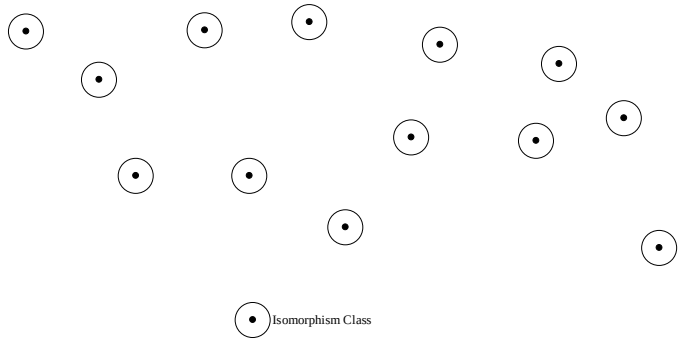
Supersingular Isogeny Problem

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 $O(p^{1/6})$

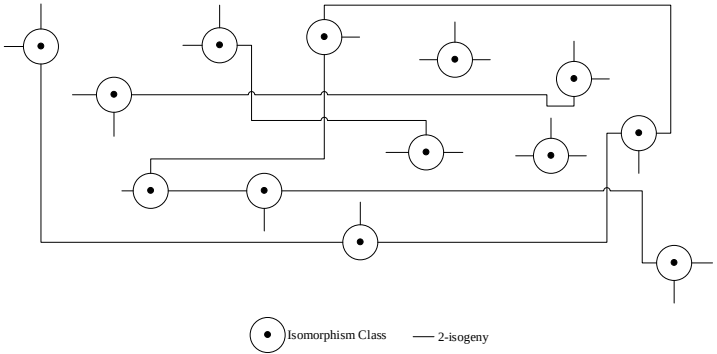
Visualizing the Supersingular Isogeny Problem

Consider a graph where each node represents supersingular isomorphism classes



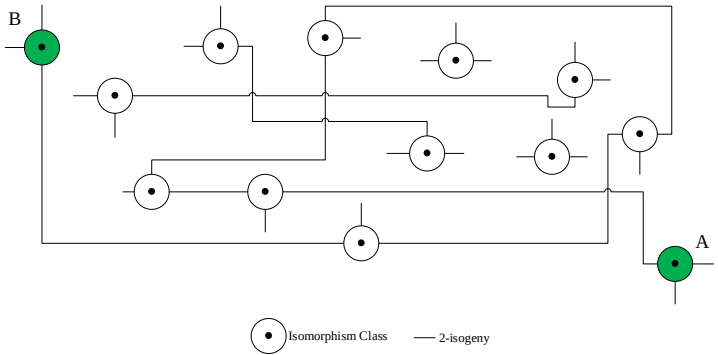
Visualizing the Supersingular Isogeny Problem

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



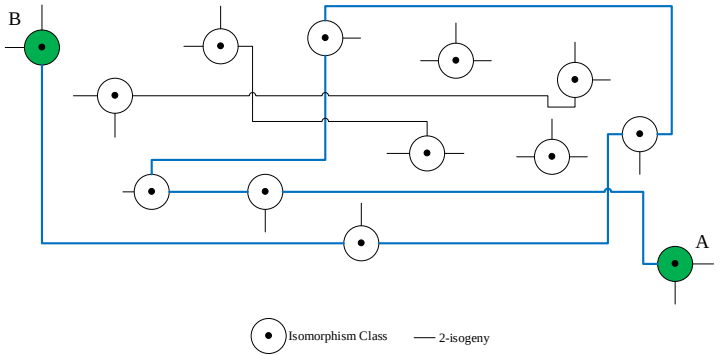
Visualizing the Supersingular Isogeny Problem

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

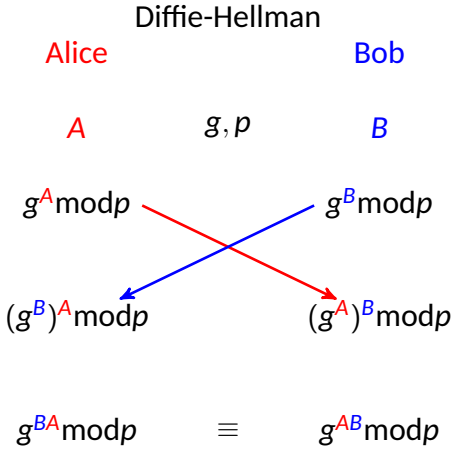


Visualizing the Supersingular Isogeny Problem

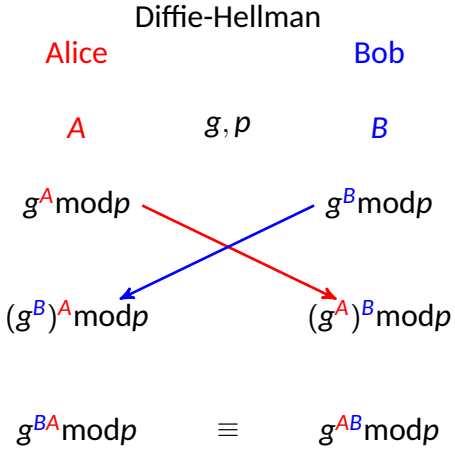
For large isogeny graphs (i.e., *p* is 512 bits or more), finding an isogeny path is HARD.

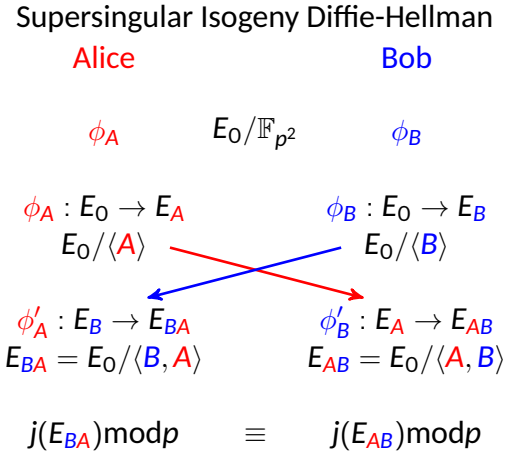


The SIDH protocol resembles standard Diffie-Hellman

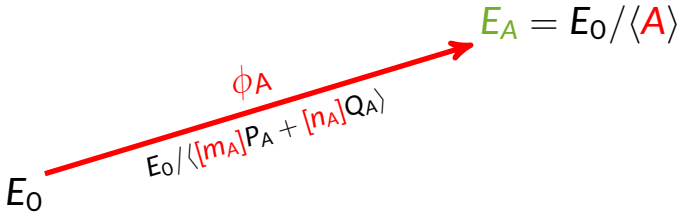


The SIDH protocol resembles standard Diffie-Hellman

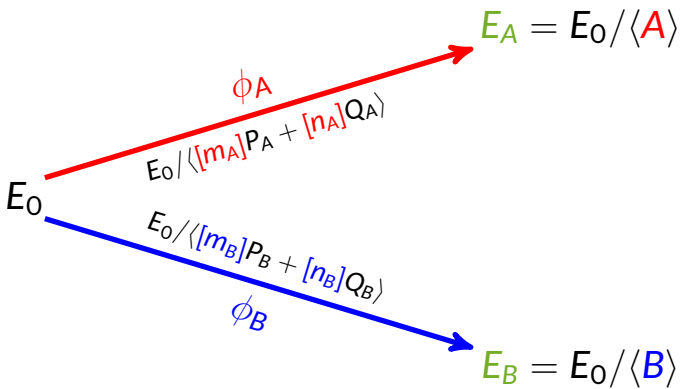




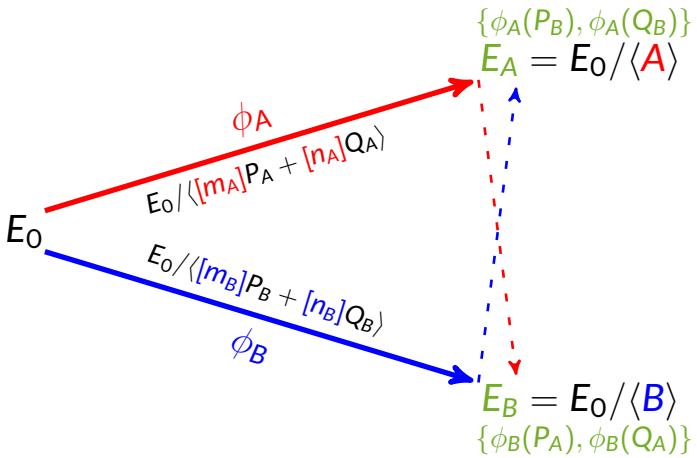
SIDH Protocol



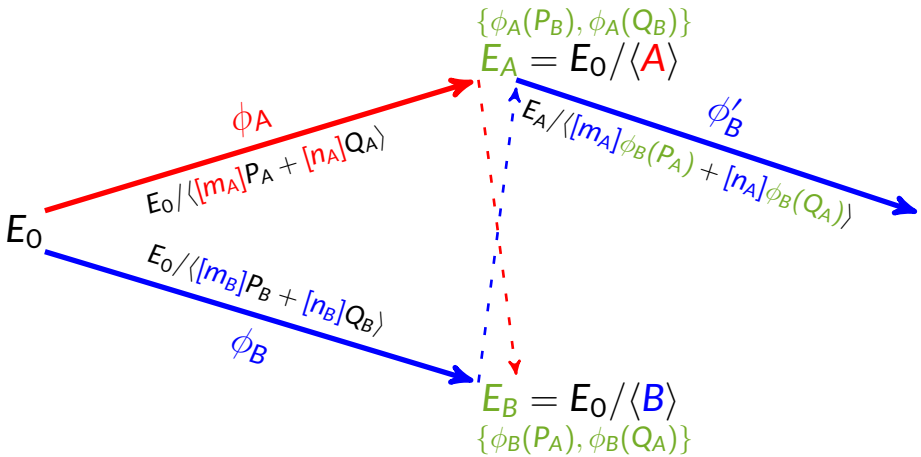
SIDH Protocol



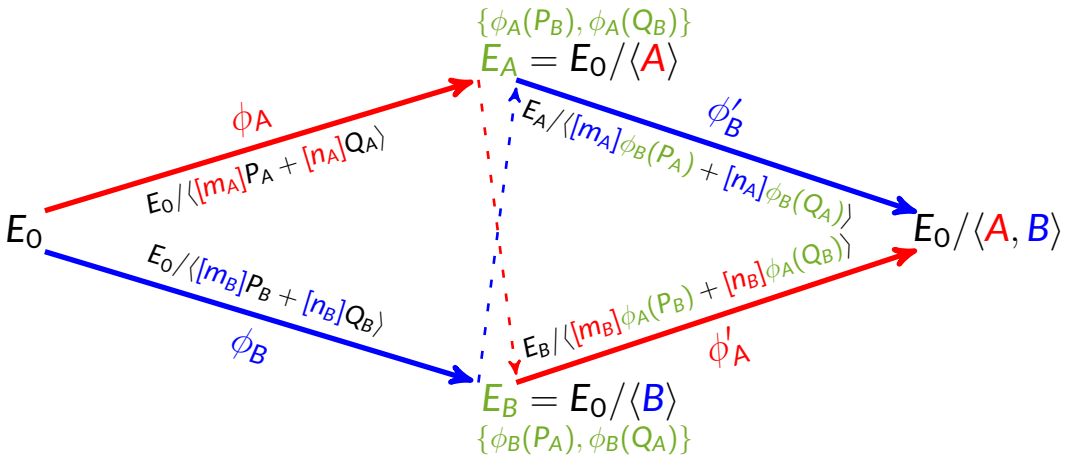
SIDH Protocol



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

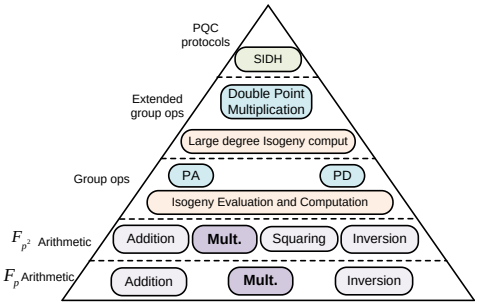
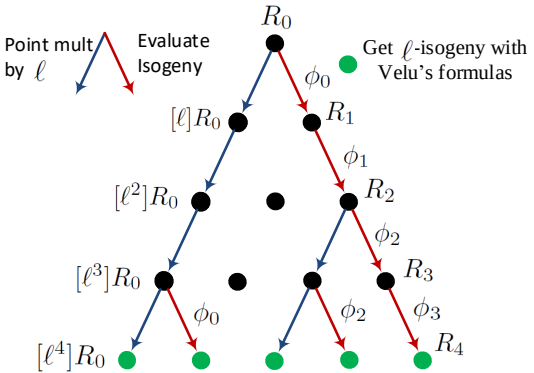


Figure: Breakdown of supersingular isogeny computations

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 → 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 → this is very bad
- CLASS 3: Intermediate basis point is exposed

What are the exposure classes?

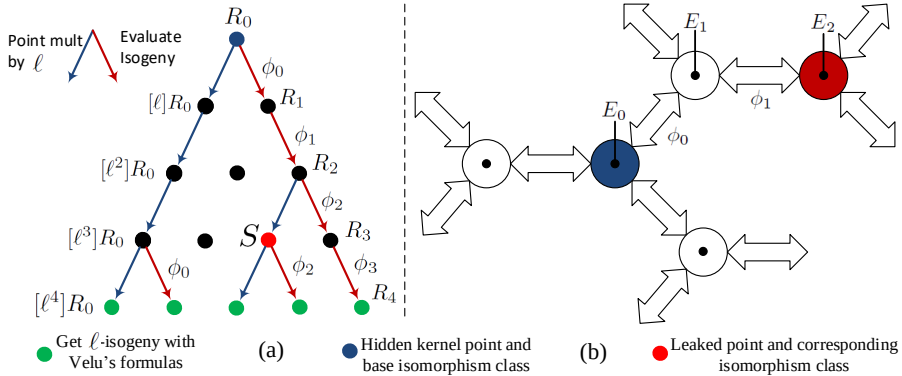
- CLASS 1: Intermediate curve is exposed
  - An attacker can now split the large-degree isogeny into two separate isogenies
  - Worst case scenario could cut the security assumption in half
- CLASS 2: Intermediate kernel point is exposed
  - Some variant of the secret kernel point is leaked
  - If corresponding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

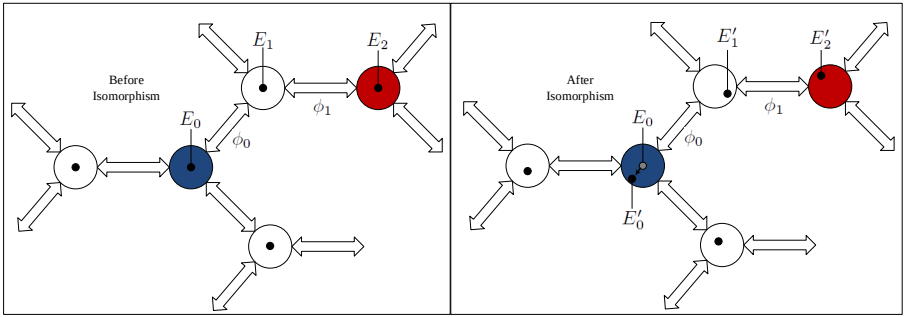
- Vélu's formulas for isogenies are deterministic for a given kernel point and curve
- An isomorphism moves from one curve to another within an isomorphism class
- A random isomorphism will scale the curve

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

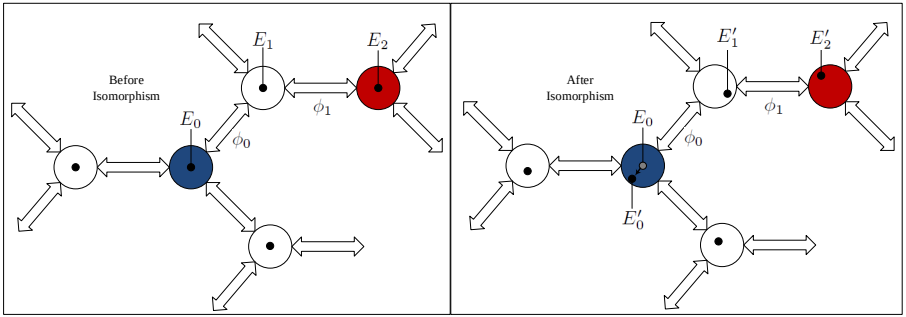
Visualization of a pre-isogeny isomorphism

- 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?