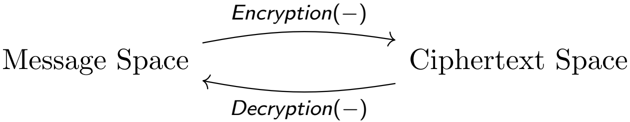
Quantum-resistant cryptography from supersingular elliptic curves

David Urbanik

July 12, 2016

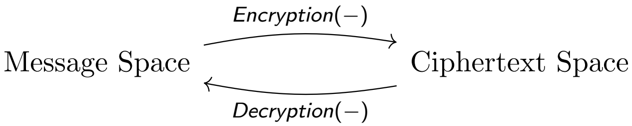
Symmetric Ciphers and Shared Secrets

To communicate cryptographically, we need to specify (at least) a function and its inverse:

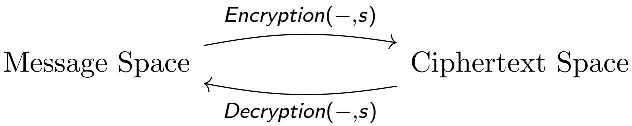


Symmetric Ciphers and Shared Secrets

To communicate cryptographically, we need to specify (at least) a function and its inverse:

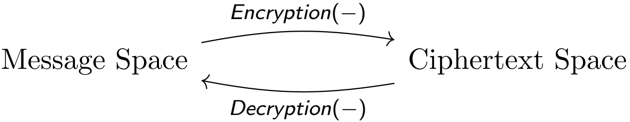


Just *one* pair of these functions is not enough; need (exponentially) many of them indexed by a parameter s:

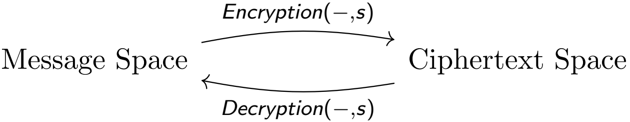


Symmetric Ciphers and Shared Secrets

To communicate cryptographically, we need to specify (at least) a function and its inverse:



Just *one* pair of these functions is not enough; need (exponentially) many of them indexed by a parameter s:



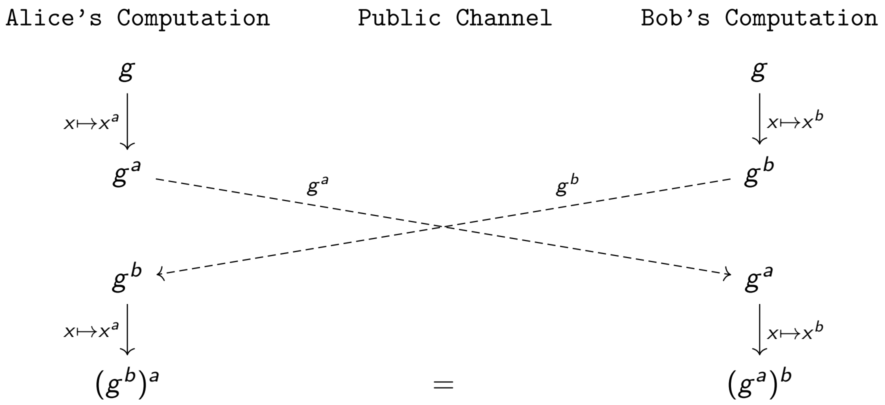
To do encryption, two parties need to agree on s. In 1976, Diffie and Hellman showed it can be done over a public channel.

The Diffie-Hellman Protocol

Setup: Fix a group G and $g \in G$.

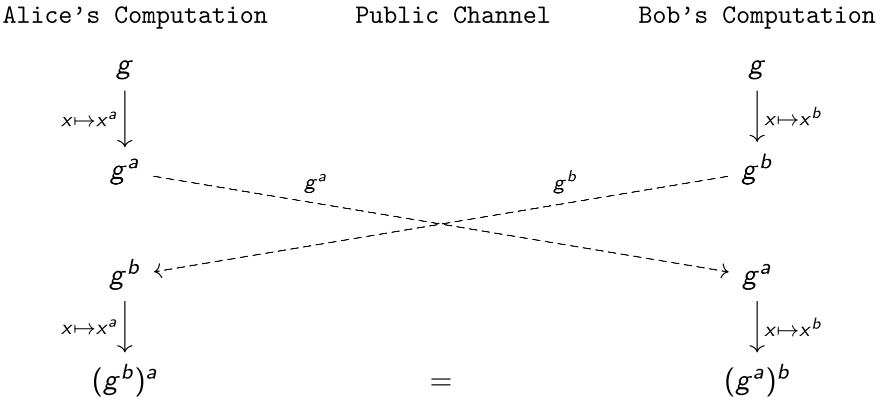
The Diffie-Hellman Protocol

Setup: Fix a group G and $g \in G$.



The Diffie-Hellman Protocol

Setup: Fix a group G and $g \in G$.



Hard problem: Given g, g^a, and g^b, determine g^{ab}.

Cryptography Under Attack

Cryptography Under Attack

- Nearly all cryptosystems which use only an open channel are broken by quantum computers.

Cryptography Under Attack

- Nearly all cryptosystems which use only an open channel are broken by quantum computers.
- Quantum computers can factor integers (breaks RSA), and compute discrete logarithms (breaks all forms of Diffie-Hellman).

Cryptography Under Attack

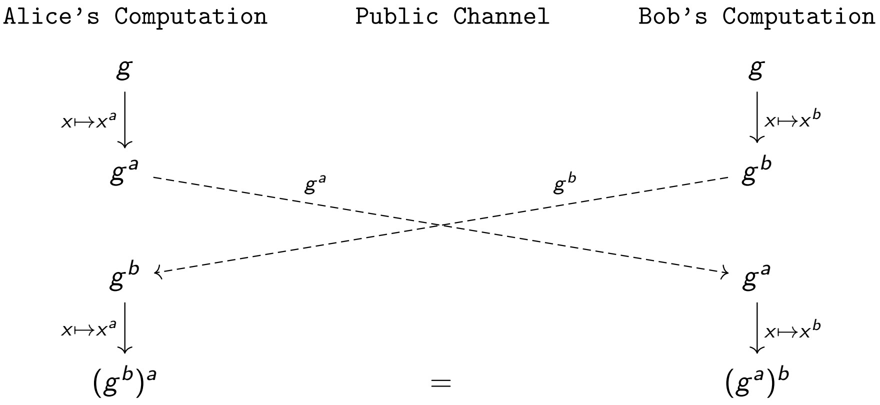
- Nearly all cryptosystems which use only an open channel are broken by quantum computers.
- Quantum computers can factor integers (breaks RSA), and compute discrete logarithms (breaks all forms of Diffie-Hellman).
- Both are special cases of the Abelian Hidden subgroup problem:

Abelian Hidden Subgroup Problem

Input: Abelian group G, a set X, $H \leq G$, and $f : G \to X$ where $f(g_1) = f(g_2)$ iff $g_1 H = g_2 H$.
Output: A generating set for H.

The Diffie-Hellman Protocol (again)

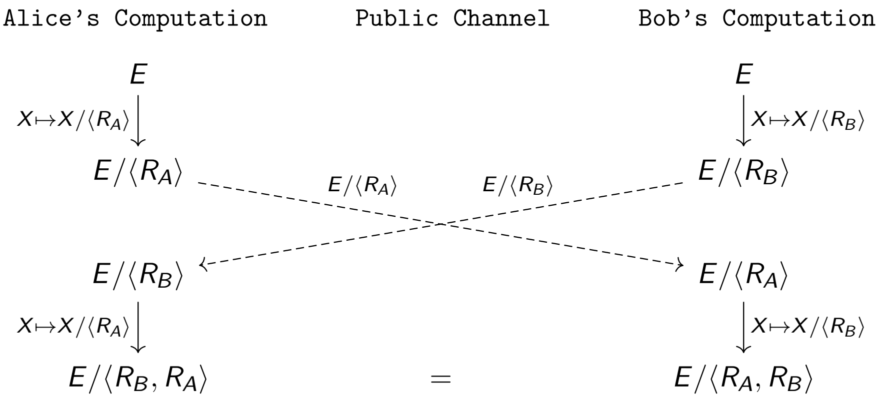
Setup: Fix a group G and $g \in G$.



Hard problem: Given g, g^a, and g^b, determine g^{ab}.

The Supersingular Isogeny Diffie-Hellman Protocol

Setup: Fix a supersingular isogeny class $\mathcal{C}$ and $E \in \mathcal{C}$.



Hard problem: Given E, $E/\langle R_A \rangle$, $E/\langle R_B \rangle$ *, determine $E/\langle R_A, R_B \rangle$.

* Some extra information is also available.

Elliptic Curves

A set of solutions $\{(x, y)\}$ over a field $\mathbb{k}$ to an equation of the form

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad .$$

Elliptic Curves

A set of solutions $\{(x, y)\}$ over a field $\Bbbk$ to an equation of the form

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad .$$

After a change of coordinates:

- **Weierstrass Form**

$$y^2 = x^3 + ax + b$$

- **Montgomery Form**

$$by^2 = x^3 + ax^2 + x$$

- **Legendre Form**

$$y^2 = x(x-1)(x-\lambda)$$

Elliptic Curves

A set of solutions $\{(x, y)\}$ over a field $\mathbb{k}$ to an equation of the form

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad .$$

After a change of coordinates:

- **Weierstrass Form**
$$y^2 = x^3 + ax + b$$
- **Montgomery Form**
$$by^2 = x^3 + ax^2 + x$$
- **Legendre Form**
$$y^2 = x(x-1)(x-\lambda)$$

Elliptic curves also need to be non-singular, i.e. there is a unique tangent line at every point.

Elliptic Curve Examples

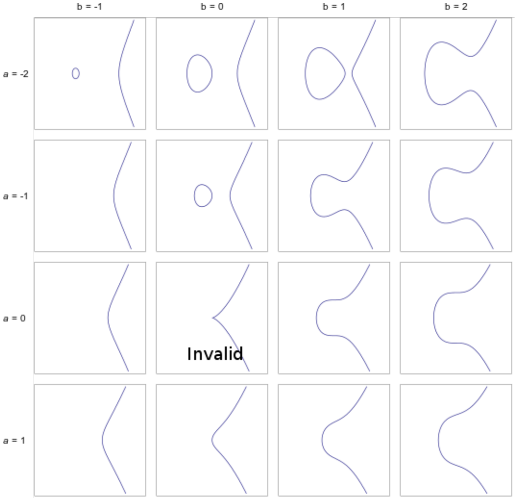


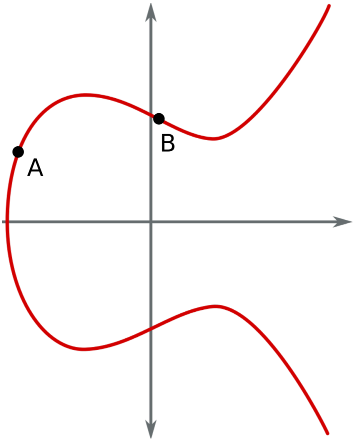
Figure: $y^2 = x^3 + ax + b, \qquad a \in \{-2, -1, 0, 1\}$ and $b \in \{-1, 0, 1, 2\}$.

The Group of an Elliptic Curve E

$$G = \{(x, y) \in \mathbb{k}^2 : (x, y) \text{ is a point on } E\}$$

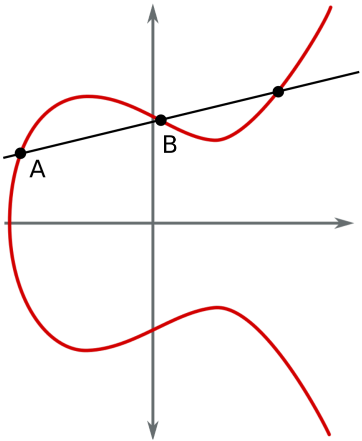
The Group of an Elliptic Curve E

$$G = \{(x, y) \in \mathbb{k}^2 : (x, y) \text{ is a point on } E\}$$



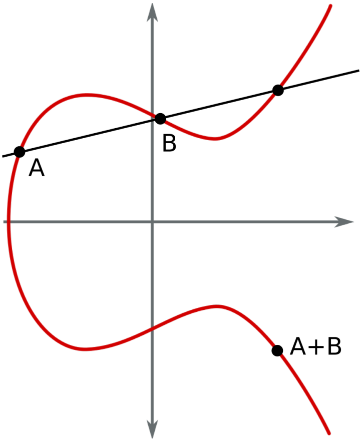
The Group of an Elliptic Curve E

$$G = \{(x, y) \in \mathbb{k}^2 : (x, y) \text{ is a point on } E\}$$



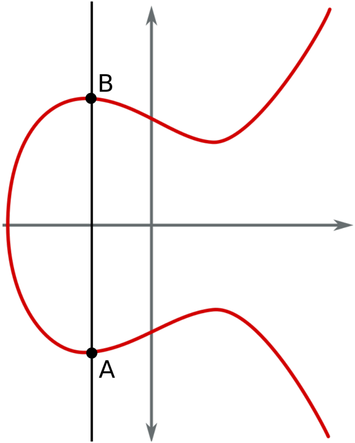
The Group of an Elliptic Curve E

$$G = \{(x, y) \in \mathbb{k}^2 : (x, y) \text{ is a point on } E\}$$



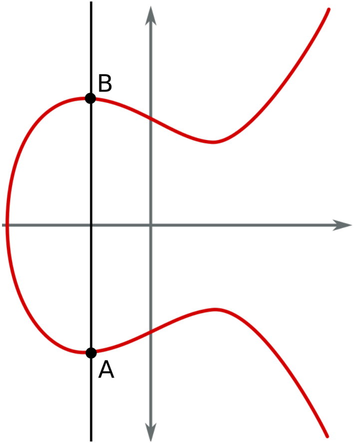
The Group of an Elliptic Curve E

$$G = \{(x, y) \in \mathbb{k}^2 : (x, y) \text{ is a point on } E\}$$



The Group of an Elliptic Curve E

$$G = \{(x, y) \in \mathbb{k}^2 : (x, y) \text{ is a point on } E\} \cup \{\infty\}$$

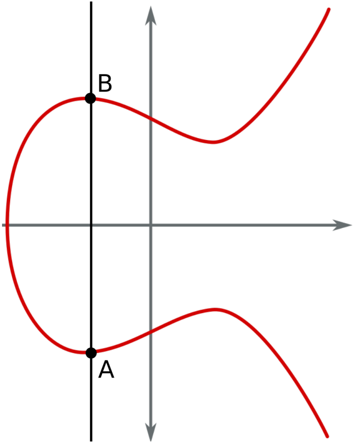


Define a point "at ∞" such that

$$A + B = \infty \quad .$$

The Group of an Elliptic Curve E

$$G = \{(x, y) \in \Bbbk^2 : (x, y) \text{ is a point on } E\} \cup \{\infty\}$$



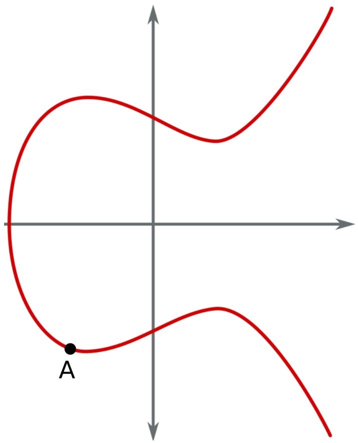
Define a point "at ∞" such that

$$A + B = \infty \quad .$$

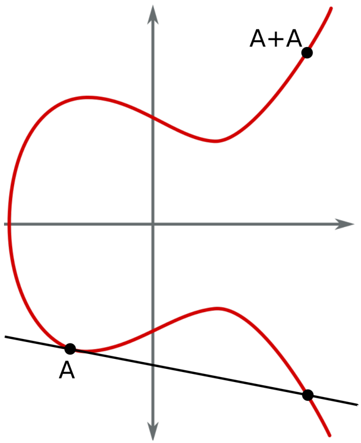
This way we get an identity element, and also inverses.

The Group of an Elliptic Curve E

$$G = \{(x, y) \in \Bbbk^2 : (x, y) \text{ is a point on } E\} \cup \{\infty\}$$



$$G = \{(x, y) \in \Bbbk^2 : (x, y) \text{ is a point on } E\} \cup \{\infty\}$$



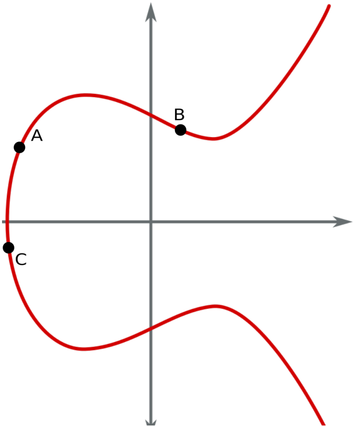
The Group of an Elliptic Curve E

The Group of an Elliptic Curve E

Why is this group operation associative?

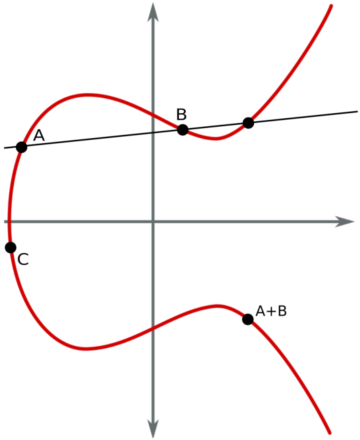
The Group of an Elliptic Curve E

Why is this group operation associative?



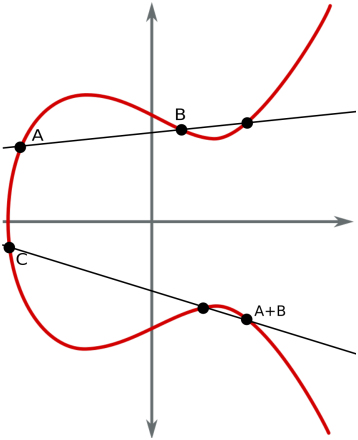
The Group of an Elliptic Curve E

Why is this group operation associative?



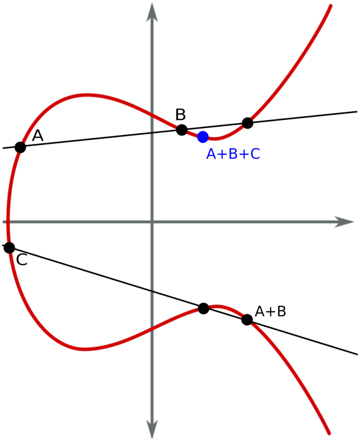
The Group of an Elliptic Curve E

Why is this group operation associative?



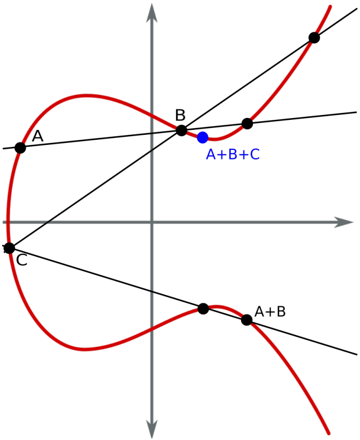
The Group of an Elliptic Curve E

Why is this group operation associative?



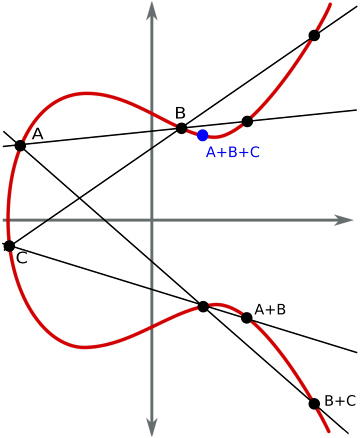
The Group of an Elliptic Curve E

Why is this group operation associative?



The Group of an Elliptic Curve E

Why is this group operation associative?



Supersingular elliptic curves

Supersingular elliptic curves

Many equivalent definitions:

Supersingular elliptic curves

Many equivalent definitions:

- Elliptic curves for which the endomorphism ring has rank 4 (is an order in a quaternion algebra); normal elliptic curves have rank 2 or 1.

Supersingular elliptic curves

Many equivalent definitions:

- Elliptic curves for which the endomorphism ring has rank 4 (is an order in a quaternion algebra); normal elliptic curves have rank 2 or 1.
- The group of p-torsion points of E is trivial, where $\mathrm{char}(\Bbbk) = p$.

Supersingular elliptic curves

Many equivalent definitions:

- Elliptic curves for which the endomorphism ring has rank 4 (is an order in a quaternion algebra); normal elliptic curves have rank 2 or 1.
- The group of p-torsion points of E is trivial, where $\mathrm{char}(\mathbb{k}) = p$.
- Writing E in Legendre form, $E : y^2 = x(x-1)(x-\lambda)$ is supersingular iff λ is a root of

$$f(x) = \sum_{i=0}^{\frac{p-1}{2}} \binom{n}{i}^2 x^i \quad .$$

Supersingular elliptic curves

Many equivalent definitions:

- Elliptic curves for which the endomorphism ring has rank 4 (is an order in a quaternion algebra); normal elliptic curves have rank 2 or 1.
- The group of p-torsion points of E is trivial, where $\mathrm{char}(\Bbbk) = p$.
- Writing E in Legendre form, $E : y^2 = x(x-1)(x-\lambda)$ is supersingular iff λ is a root of

$$f(x) = \sum_{i=0}^{\frac{p-1}{2}} \binom{n}{i}^2 x^i \quad .$$

- If we write E as a cubic homogeneous polynomial $f(x, y, z)$ in the projective plane, then E is supersingular iff the coefficient of $(xyz)^{p-1}$ in $f(x, y, z)^{p-1}$ is zero.

Quotients of elliptic curves

Quotients of elliptic curves

- Quotients are generally associated to a surjective quotient map; in this case, the map is called an *isogeny*.

Quotients of elliptic curves

- Quotients are generally associated to a surjective quotient map; in this case, the map is called an *isogeny*.
- Isogenies are given by non-constant rational functions which fix the identity point.

Quotients of elliptic curves

- Quotients are generally associated to a surjective quotient map; in this case, the map is called an *isogeny*.
- Isogenies are given by non-constant rational functions which fix the identity point.
- It is a theorem that such a map is always a group homomorphism.

Quotients of elliptic curves

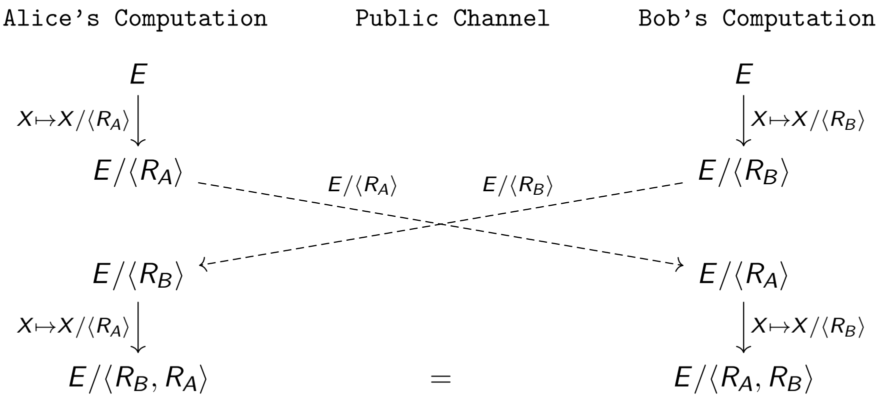
- Quotients are generally associated to a surjective quotient map; in this case, the map is called an *isogeny*.
- Isogenies are given by non-constant rational functions which fix the identity point.
- It is a theorem that such a map is always a group homomorphism.
- If Φ is a subgroup of the elliptic curve group of E, then there is (up to isomorphism) a unique isogeny with kernel Φ (comes from Velu's formulas).

Quotients of elliptic curves

- Quotients are generally associated to a surjective quotient map; in this case, the map is called an *isogeny*.
- Isogenies are given by non-constant rational functions which fix the identity point.
- It is a theorem that such a map is always a group homomorphism.
- If Φ is a subgroup of the elliptic curve group of E, then there is (up to isomorphism) a unique isogeny with kernel Φ (comes from Velu's formulas).
- The image curve under the isogeny is the quotient curve.

The Supersingular Isogeny Diffie-Hellman Protocol

Setup: Fix a supersingular isogeny class $\mathcal{C}$ and $E \in \mathcal{C}$.



Hard problem: Given E, $E/\langle R_A\rangle$, $E/\langle R_B\rangle$ *, determine $E/\langle R_A, R_B\rangle$.

* Some extra information is also available.

Computing the Second Isogeny

- Suppose Alice receives the curve $E/\langle R_B \rangle$ from Bob and she wants to compute $(E/\langle R_B \rangle)/\langle R_A \rangle = E/\langle R_B, R_A \rangle$.

Computing the Second Isogeny

- Suppose Alice receives the curve $E/\langle R_B \rangle$ from Bob and she wants to compute $(E/\langle R_B \rangle)/\langle R_A \rangle = E/\langle R_B, R_A \rangle$.
- To use Velu's formulas, we need to know a subgroup of the domain curve, but R_A is a subgroup of the original curve E.

Computing the Second Isogeny

- Suppose Alice receives the curve $E/\langle R_B \rangle$ from Bob and she wants to compute $(E/\langle R_B \rangle)/\langle R_A \rangle = E/\langle R_B, R_A \rangle$.
- To use Velu's formulas, we need to know a subgroup of the domain curve, but R_A is a subgroup of the original curve E.
- Bob could compute the action of his isogeny ϕ_B on R_A for Alice, but this would require exchanging R_A over the public channel.

Computing the Second Isogeny

- Suppose Alice receives the curve $E/\langle R_B \rangle$ from Bob and she wants to compute $(E/\langle R_B \rangle)/\langle R_A \rangle = E/\langle R_B, R_A \rangle$.
- To use Velu's formulas, we need to know a subgroup of the domain curve, but R_A is a subgroup of the original curve E.
- Bob could compute the action of his isogeny ϕ_B on R_A for Alice, but this would require exchanging R_A over the public channel.
- The solution is to choose two special points P_A and Q_A on E, and then choose $R_A = m_A P_A + n_A Q_A$ for some integers m_A and n_A.

Computing the Second Isogeny

- Suppose Alice receives the curve $E/\langle R_B \rangle$ from Bob and she wants to compute $(E/\langle R_B \rangle)/\langle R_A \rangle = E/\langle R_B, R_A \rangle$.
- To use Velu's formulas, we need to know a subgroup of the domain curve, but R_A is a subgroup of the original curve E.
- Bob could compute the action of his isogeny ϕ_B on R_A for Alice, but this would require exchanging R_A over the public channel.
- The solution is to choose two special points P_A and Q_A on E, and then choose $R_A = m_A P_A + n_A Q_A$ for some integers m_A and n_A.
- Bob then sends $\phi_B(P_A)$ and $\phi_B(Q_A)$ to Alice, and then Alice can compute

$$\langle m_A \phi_B(P_A) + n_A \phi_B(Q_A) \rangle = \langle \phi_B(m_A P_A + n_A Q_A) \rangle = \langle \phi_B(R_A) \rangle \quad .$$

My Work

- The current leading implementation of SIDH was developed by researchers at Microsoft and released in April of 2016. I've been optimizing the finite field arithmetic used for 64-bit ARM architectures. Because the original algorithm used for finite field operations on this platform was very generic, using hand-coded 64-bit ARM assembly I was able to improve the performance by about a factor of 10.
- Of all the quantum-resistant key-exchange protocols, SIDH has by far the smallest key sizes, which can be made smaller with compression. I am currently working on implementing key compression and decompression algorithms which can make the key sizes of SIDH comparable with those of existing quantum-vulnerable algorithms.