A (hopefully) friendly introduction to Isogeny-Based Cryptography

Silvia Sconza

PhD Seminars
University of Genoa, DIMA

June 29, 2023

Table of Contents

Purpose and Terminology

The purpose of cryptography is to find ways (protocols) to communicate securely, assuming the presence of eavesdroppers (Eve).

Purpose and Terminology

The purpose of cryptography is to find ways (protocols) to communicate securely, assuming the presence of eavesdroppers (Eve).

We want to transform our messages (Encryption) in such a way that opponents will find it to be unintelligible text and only the predestined receiver will be able to trace the original message (Decryption).

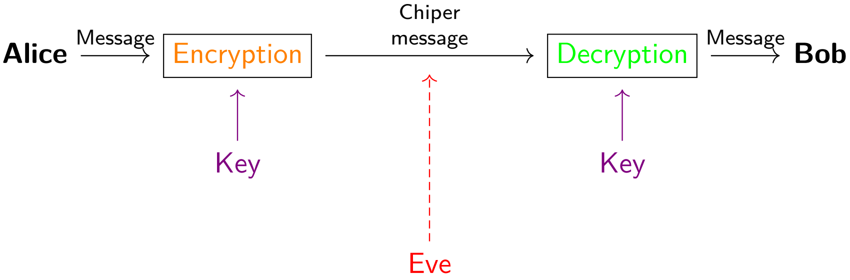
Purpose and Terminology

The purpose of cryptography is to find ways (protocols) to communicate securely, assuming the presence of eavesdroppers (Eve).

We want to transform our messages (Encryption) in such a way that opponents will find it to be unintelligible text and only the predestined receiver will be able to trace the original message (Decryption).

In order to carry out encryption and decryption, we need so-called cryptographic keys.

Purpose and Terminology



Types and keys

Two main types of cryptography:

Types and keys

Two main types of cryptography:

- Symmetric-Key Cryptography: same secret key to encrypt and decrypt the message;

Types and keys

Two main types of cryptography:

- Symmetric-Key Cryptography: same secret key to encrypt and decrypt the message;
- Public-Key Cryptography: two keys involved: a public one known to all and a private one known only to the owner.

Types and keys

Two main types of cryptography:

- Symmetric-Key Cryptography: same secret key to encrypt and decrypt the message;
- Public-Key Cryptography: two keys involved: a public one known to all and a private one known only to the owner.

• Key Exchange Problem: how can two parties exchange keys in such a way as to establish a secure communication channel?

Diffie-Hellman Key Exchange

Diffie-Hellman Key Exchange (DHKE), 1976

1. Alice and Bob publicly agree on a cyclic finite group G and a generator g.
2. Alice chooses $a \in \{1, \ldots, \mathrm{ord}(G)\}$, computes g^a and sends it to Bob. Her secret key is a.
3. Bob chooses $b \in \{1, \ldots, \mathrm{ord}(G)\}$, computes g^b and sends it to Alice. His secret key is b.
4. Alice computes $(g^b)^a = g^{ba}$.
5. Bob computes $(g^a)^b = g^{ab}$.

The secret common key is $g^{ba} = g^{ab}$.

Diffie-Hellman Key Exchange

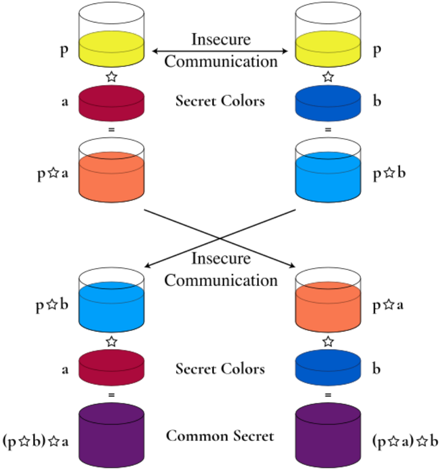
Diffie-Hellman Key Exchange (DHKE), 1976

1. Alice and Bob publicly agree on a cyclic finite group G and a generator g.
2. Alice chooses $a \in \{1, \dots, \mathrm{ord}(G)\}$, computes g^a and sends it to Bob. Her secret key is a.
3. Bob chooses $b \in \{1, \dots, \mathrm{ord}(G)\}$, computes g^b and sends it to Alice. His secret key is b.
4. Alice computes $(g^b)^a = g^{ba}$.
5. Bob computes $(g^a)^b = g^{ab}$.

The secret common key is $g^{ba} = g^{ab}$.

- Diffie-Hellman Problem (DHP): Let G be a finite cyclic group and let g be a generator. Given g^a and g^b, find g^{ab}.

Diffie-Hellman Key Exchange



[Picture from Borradaile, G. "Defend Dissent." Corvallis: Oregon State University, 2021.]

1994: The security of current cryptosystems is based on the difficulty of integer factorisation and the discrete logarithm. Both problems can be solved in polynomial time using Shor's algorithm for a sufficiently large quantum computer.

1994: The security of current cryptosystems is based on the difficulty of integer factorisation and the discrete logarithm. Both problems can be solved in polynomial time using Shor's algorithm for a sufficiently large quantum computer.

2016: The National Institute of Standards and Technology (NIST) opens a call for standardization asking for post-quantum cryptographic algorithms.

1994: The security of current cryptosystems is based on the difficulty of integer factorisation and the discrete logarithm. Both problems can be solved in polynomial time using Shor's algorithm for a sufficiently large quantum computer.

2016: The National Institute of Standards and Technology (NIST) opens a call for standardization asking for post-quantum cryptographic algorithms.

Proposals:

1994: The security of current cryptosystems is based on the difficulty of integer factorisation and the discrete logarithm. Both problems can be solved in polynomial time using Shor's algorithm for a sufficiently large quantum computer.

2016: The National Institute of Standards and Technology (NIST) opens a call for standardization asking for post-quantum cryptographic algorithms.

Proposals:

- Lattice-based Crypto

Post-Quantum Cryptography

1994: The security of current cryptosystems is based on the difficulty of integer factorisation and the discrete logarithm. Both problems can be solved in polynomial time using Shor's algorithm for a sufficiently large quantum computer.

2016: The National Institute of Standards and Technology (NIST) opens a call for standardization asking for post-quantum cryptographic algorithms.

Proposals:

- Lattice-based Crypto
- Code-based Crypto

Post-Quantum Cryptography

1994: The security of current cryptosystems is based on the difficulty of integer factorisation and the discrete logarithm. Both problems can be solved in polynomial time using Shor's algorithm for a sufficiently large quantum computer.

2016: The National Institute of Standards and Technology (NIST) opens a call for standardization asking for post-quantum cryptographic algorithms.

Proposals:

- Lattice-based Crypto
- Code-based Crypto
- Multivariate Crypto

Post-Quantum Cryptography

1994: The security of current cryptosystems is based on the difficulty of integer factorisation and the discrete logarithm. Both problems can be solved in polynomial time using Shor's algorithm for a sufficiently large quantum computer.

2016: The National Institute of Standards and Technology (NIST) opens a call for standardization asking for post-quantum cryptographic algorithms.

Proposals:

- Lattice-based Crypto
- Code-based Crypto
- Multivariate Crypto
- Isogeny-based Crypto

Post-Quantum Cryptography

1994: The security of current cryptosystems is based on the difficulty of integer factorisation and the discrete logarithm. Both problems can be solved in polynomial time using Shor's algorithm for a sufficiently large quantum computer.

2016: The National Institute of Standards and Technology (NIST) opens a call for standardization asking for post-quantum cryptographic algorithms.

Proposals:

- Lattice-based Crypto
- Code-based Crypto
- Multivariate Crypto
- Isogeny-based Crypto
- Hash-based Crypto

1994: The security of current cryptosystems is based on the difficulty of integer factorisation and the discrete logarithm. Both problems can be solved in polynomial time using Shor's algorithm for a sufficiently large quantum computer.

2016: The National Institute of Standards and Technology (NIST) opens a call for standardization asking for post-quantum cryptographic algorithms.

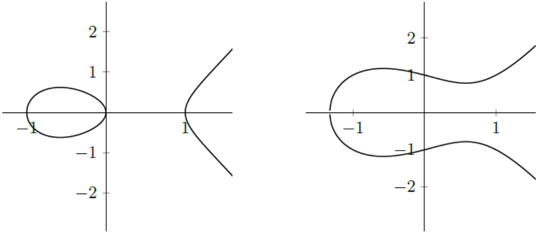
Proposals:

- Lattice-based Crypto
- Code-based Crypto
- Multivariate Crypto
- Isogeny-based Crypto
- Hash-based Crypto
- Others

Table of Contents

Elliptic Curves

An *elliptic curve* is a pair (E, O_E), where E is a nonsingular projective curve of genus 1 and $O_E \in E$ is a fixed point.

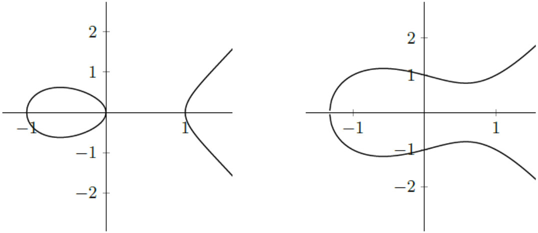


(a) $y^2 = x^3 - x$

(b) $y^2 = x^3 - x + 1$

Elliptic Curves

An *elliptic curve* is a pair (E, O_E), where E is a nonsingular projective curve of genus 1 and $O_E \in E$ is a fixed point.



(a) $y^2 = x^3 - x$ (b) $y^2 = x^3 - x + 1$

Weierstrass form: $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$

If $\mathrm{char}(k) \neq 2, 3$: $y^2 = x^3 + Ax + B$

The *discriminant* of E is $\Delta(E) = -(4A^3 + 27B^2)$.

Elliptic curves

University of Zurich[UZH]

The *discriminant* of E is $\Delta(E) = -(4A^3 + 27B^2)$.
The *j-invariant* of E is

$$j(E) = 1728 \frac{4A^3}{4A^3 + 27B^2}.$$

Elliptic curves

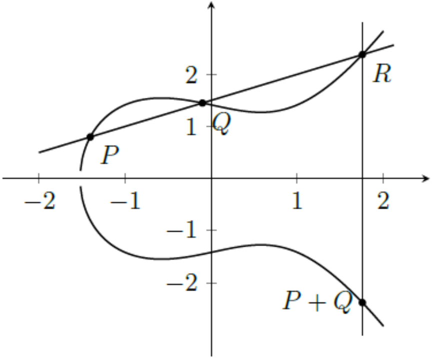
The *discriminant* of E is $\Delta(E) = -(4A^3 + 27B^2)$.
The *j-invariant* of E is

$$j(E) = 1728 \frac{4A^3}{4A^3 + 27B^2}.$$

Properties

- A curve given by a Weierstrass equation is nonsingular if and only if $\Delta(E) \neq 0$.
- Two elliptic curves are isomorphic over $\overline{k}$ if and only if they have the same j-invariant.
- Let $j_0 \in \overline{k}$. There exists an elliptic curve defined over $k(j_0)$ whose j-invariant is j_0.

Group law:



Isogenies

An *isogeny* between two elliptic curves E_1 and E_2 is a morphism $\phi\colon E_1 \to E_2$ such that $\phi(O_{E_1}) = O_{E_2}$.

Isogenies

An *isogeny* between two elliptic curves E_1 and E_2 is a morphism $\phi \colon E_1 \to E_2$ such that $\phi(O_{E_1}) = O_{E_2}$.
An isogeny is a group homomorphism.

An *isogeny* between two elliptic curves E_1 and E_2 is a morphism $\phi\colon E_1 \to E_2$ such that $\phi(O_{E_1}) = O_{E_2}$.
An isogeny is a group homomorphism.
We indicate the set (group) of such isogenies with $\mathrm{Hom}(E_1, E_2)$. Moreover $\mathrm{End}(E) = \mathrm{Hom}(E, E)$ has a ring structure.

Isogenies

An *isogeny* between two elliptic curves E_1 and E_2 is a morphism $\phi\colon E_1 \to E_2$ such that $\phi(O_{E_1}) = O_{E_2}$.
An isogeny is a group homomorphism.
We indicate the set (group) of such isogenies with $\mathrm{Hom}(E_1, E_2)$. Moreover $\mathrm{End}(E) = \mathrm{Hom}(E, E)$ has a ring structure.

An example of isogeny is the *multiplication-by-m* with $m \in \mathbb{Z}$:

$$\begin{aligned}[m]\colon E &\to E \\ P &\mapsto P + \cdots + P\end{aligned}$$

Definition

Two elliptic curves E, E' are *ℓ-isogenous* if there exists an isogeny $\varphi\colon E \to E'$ of degree ℓ.
An isogeny of degree ℓ is called *ℓ-isogeny*.

Definition

Two elliptic curves E, E' are *ℓ-isogenous* if there exists an isogeny $\varphi\colon E \to E'$ of degree ℓ.
An isogeny of degree ℓ is called *ℓ-isogeny*.

Theorem

Let $\varphi\colon E \to E'$ be an isogeny of degree ℓ. Then there exists an isogeny $\widehat{\varphi}\colon E' \to E$ of degree ℓ, called *dual isogeny*, such that

$$\varphi \circ \widehat{\varphi} = [\ell] \quad \text{and} \quad \widehat{\varphi} \circ \varphi = [\ell].$$

Ordinary and Supersingular EC

Elliptic curves can be partitioned into two families: the ordinary EC and the supersingular EC.

Ordinary and Supersingular EC

Elliptic curves can be partitioned into two families: the ordinary EC and the supersingular EC.

Properties:

Ordinary and Supersingular EC

Elliptic curves can be partitioned into two families: the ordinary EC and the supersingular EC.

Properties:

- If $\mathrm{char}(k) = 0$, then all the elliptic curves are ordinary.

Ordinary and Supersingular EC

Elliptic curves can be partitioned into two families: the ordinary EC and the supersingular EC.

Properties:

- If $\mathrm{char}(k) = 0$, then all the elliptic curves are ordinary.
- If $\mathrm{char}(k) = p$ and E is a supersingular elliptic curve, then $j(E) \in \mathbb{F}_{p^2}$.

Ordinary and Supersingular EC

Elliptic curves can be partitioned into two families: the ordinary EC and the supersingular EC.

Properties:

- If $\mathrm{char}(k) = 0$, then all the elliptic curves are ordinary.
- If $\mathrm{char}(k) = p$ and E is a supersingular elliptic curve, then $j(E) \in \mathbb{F}_{p^2}$.
- **Tate's Theorem:** If two elliptic curves are isogenous, then they are of the same type.

Ordinary and Supersingular EC

Elliptic curves can be partitioned into two families: the ordinary EC and the supersingular EC.

Properties:

- If $\mathrm{char}(k) = 0$, then all the elliptic curves are ordinary.
- If $\mathrm{char}(k) = p$ and E is a supersingular elliptic curve, then $j(E) \in \mathbb{F}_{p^2}$.
- **Tate's Theorem:** If two elliptic curves are isogenous, then they are of the same type.
- The endomorphism ring of an ordinary elliptic curve is commutative. The endomorphism ring of a supersingular elliptic curve is noncommutative.

Table of Contents

Underlying Problems

In order to well-define a cryptosystem, we need to base it on a hard mathematical problem.

Underlying Problems

In order to well-define a cryptosystem, we need to base it on a hard mathematical problem.

- General Isogeny Problem: Given two isogenous elliptic curves, find an isogeny between them.

Underlying Problems

In order to well-define a cryptosystem, we need to base it on a hard mathematical problem.

- General Isogeny Problem: Given two isogenous elliptic curves, find an isogeny between them.
- ℓ-Isogeny Problem: Given two ℓ-isogenous elliptic curves, find an ℓ-isogeny between them.

Isogeny graphs

Definition

Let ℓ be a prime number such that $\ell \neq \mathrm{char}(k)$.
An *ℓ-isogeny graph $G_\ell(k)$* is a graph whose vertices are j-invariants of elliptic curves defined over k and whose edges are ℓ-isogenies defined over k between them.

Isogeny graphs

Definition

Let ℓ be a prime number such that $\ell \neq \mathrm{char}(k)$.
An *ℓ-isogeny graph $G_\ell(k)$* is a graph whose vertices are j-invariants of elliptic curves defined over k and whose edges are ℓ-isogenies defined over k between them.

Thanks to the existence of dual isogeny, we can see this graph as undirected.

Isogeny graphs

Definition

Let ℓ be a prime number such that $\ell \neq \mathrm{char}(k)$.
An *ℓ-isogeny graph $G_\ell(k)$* is a graph whose vertices are j-invariants of elliptic curves defined over k and whose edges are ℓ-isogenies defined over k between them.

Thanks to the existence of dual isogeny, we can see this graph as undirected.

It follows from Tate's theorem that the graph $G_\ell(k)$ can always be partitioned into ordinary and supersingular components.

Ordinary case

Given an ℓ-isogeny of two ordinary elliptic curves, it could be horizontal, ascending or descending, depending on the relation between the endomorphism rings of the two curves.

Ordinary case

Given an ℓ-isogeny of two ordinary elliptic curves, it could be horizontal, ascending or descending, depending on the relation between the endomorphism rings of the two curves. Thanks to David Kohel, we know exactly how many ℓ-isogenies of each type we have.

Ordinary case

Given an ℓ-isogeny of two ordinary elliptic curves, it could be horizontal, ascending or descending, depending on the relation between the endomorphism rings of the two curves. Thanks to David Kohel, we know exactly how many ℓ-isogenies of each type we have.

Definition

An *ℓ-volcano* is a connected undirected graph whose vertices are partitioned into one or more *levels* $V_0, \ldots, V_d$ such that:

(i) the subgraph on V_0 (the *surface*) is a regular graph of degree at most 2;

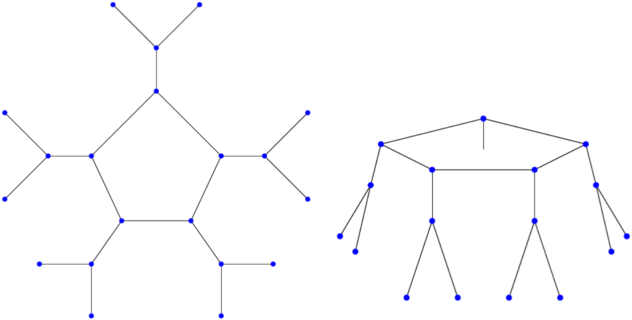
(ii) for $i > 0$, each vertex in V_i has exactly one neighbor in level V_{i-1};

(iii) for $i < d$, each vertex in V_i has degree $\ell + 1$.

We call d the *depth* of the volcano and we call V_d the *floor*.

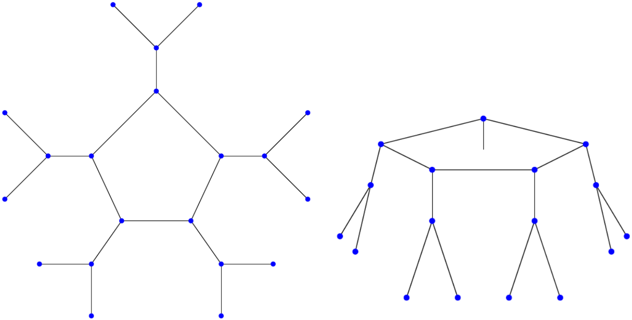
Ordinary Case

- V_0 regular graph of degree at most 2;
- each vertex in V_i has exactly one neighbor in V_{i-1}, for $i > 0$;
- each vertex in V_i has degree $\ell + 1$, for $i < d$.



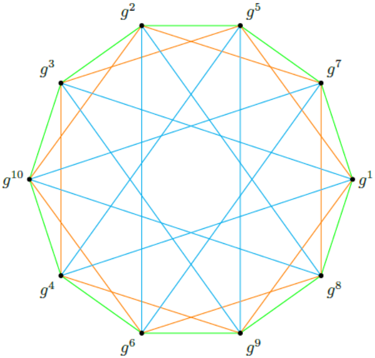
Ordinary Case

- V_0 regular graph of degree at most 2;
- each vertex in V_i has exactly one neighbor in V_{i-1}, for $i > 0$;
- each vertex in V_i has degree $\ell + 1$, for $i < d$.



- An ordinary component of $G_\ell(\mathbb{F}_q)$ is an ℓ-volcano.

An example



- V = {set of generators of a cyclic group of order 11};
- $S = \{3, 5, 7, 3^{-1}, 5^{-1}, 7^{-1}\} \subseteq (\mathbb{Z}/11\mathbb{Z})^\times$.

An example

Key exchange protocol (Couveignes, 2006)

- Public parameters
  - A group G of prime order p and a generator g;

An example

Key exchange protocol (Couveignes, 2006)

- Public parameters
  - A group G of prime order p and a generator g;
  - A generating set $D \subseteq (\mathbb{Z}/p\mathbb{Z})^\times$ such that $\sigma \in D \Rightarrow \sigma^{-1} \notin D$.

Key exchange protocol (Couveignes, 2006)

- Public parameters
  - A group G of prime order p and a generator g;
  - A generating set $D \subseteq (\mathbb{Z}/p\mathbb{Z})^\times$ such that $\sigma \in D \Rightarrow \sigma^{-1} \notin D$.
- Protocol
  1. Alice chooses a random succession ρ_A of elements in D and Bob chooses a random succession ρ_B of elements in D;

An example

Key exchange protocol (Couveignes, 2006)

- Public parameters
  - A group G of prime order p and a generator g;
  - A generating set $D \subseteq (\mathbb{Z}/p\mathbb{Z})^\times$ such that $\sigma \in D \Rightarrow \sigma^{-1} \notin D$.
- Protocol
  1. Alice chooses a random succession ρ_A of elements in D and Bob chooses a random succession ρ_B of elements in D;
  2. Alice computes $g_A = \rho_A(g)$ and sends it to Bob;

An example

Key exchange protocol (Couveignes, 2006)

- Public parameters
  - A group G of prime order p and a generator g;
  - A generating set $D \subseteq (\mathbb{Z}/p\mathbb{Z})^\times$ such that $\sigma \in D \Rightarrow \sigma^{-1} \notin D$.
- Protocol
  1. Alice chooses a random succession ρ_A of elements in D and Bob chooses a random succession ρ_B of elements in D;
  2. Alice computes $g_A = \rho_A(g)$ and sends it to Bob;
  3. Bob computes $g_B = \rho_B(g)$ and sends it to Alice;

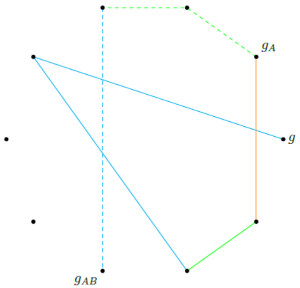
Key exchange protocol (Couveignes, 2006)

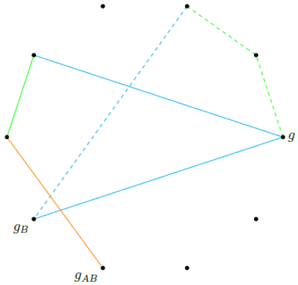
- Public parameters
  - A group G of prime order p and a generator g;
  - A generating set $D \subseteq (\mathbb{Z}/p\mathbb{Z})^\times$ such that $\sigma \in D \Rightarrow \sigma^{-1} \notin D$.
- Protocol
  1. Alice chooses a random succession ρ_A of elements in D and Bob chooses a random succession ρ_B of elements in D;
  2. Alice computes $g_A = \rho_A(g)$ and sends it to Bob;
  3. Bob computes $g_B = \rho_B(g)$ and sends it to Alice;
  4. Alice computes $g_{AB} = \rho_A(g_B)$ and Bob computes $g_{AB} = \rho_B(g_A)$.

In the figure, Alice's route is represented by continuous lines, Bob's route by dashed lines.

An example

In the figure, Alice's route is represented by continuous lines, Bob's route by dashed lines.





The order of the steps in a route does not matter: what counts is only how many times each element of D appears in the route.

Ordinary Case

Key exchange protocol (Rostovtsev-Stolbunov, 2006)

- Public parameters
  - A large finite field $\mathbb{F}_q$ and an ordinary elliptic curve E over $\mathbb{F}_q$;

Ordinary Case

Key exchange protocol (Rostovtsev-Stolbunov, 2006)

- Public parameters
  - A large finite field $\mathbb{F}_q$ and an ordinary elliptic curve E over $\mathbb{F}_q$;
  - A set $L = \{\ell_1, \ldots, \ell_m\}$ of prime numbers;

Ordinary Case

Key exchange protocol (Rostovtsev-Stolbunov, 2006)

- Public parameters
  - A large finite field $\mathbb{F}_q$ and an ordinary elliptic curve E over $\mathbb{F}_q$;
  - A set $L = \{\ell_1, \dots, \ell_m\}$ of prime numbers;
  - For each prime number ℓ_i, a positive direction chosen at random.

Ordinary Case

Key exchange protocol (Rostovtsev-Stolbunov, 2006)

- Public parameters
  - A large finite field $\mathbb{F}_q$ and an <u>ordinary</u> elliptic curve E over $\mathbb{F}_q$;
  - A set $L = \{\ell_1, \ldots, \ell_m\}$ of prime numbers;
  - For each prime number ℓ_i, a positive direction chosen at random.
- Protocol
  1. Alice chooses a random succession ρ_A of elements in L and Bob chooses a random succession ρ_B of elements in L;

Ordinary Case

Key exchange protocol (Rostovtsev-Stolbunov, 2006)

- Public parameters
  - A large finite field $\mathbb{F}_q$ and an ordinary elliptic curve E over $\mathbb{F}_q$;
  - A set $L = \{\ell_1, \ldots, \ell_m\}$ of prime numbers;
  - For each prime number ℓ_i, a positive direction chosen at random.
- Protocol
  1. Alice chooses a random succession ρ_A of elements in L and Bob chooses a random succession ρ_B of elements in L;
  2. Alice computes $E_A = \rho_A(E)$ and sends it to Bob;

Key exchange protocol (Rostovtsev-Stolbunov, 2006)

- Public parameters
  - A large finite field $\mathbb{F}_q$ and an ordinary elliptic curve E over $\mathbb{F}_q$;
  - A set $L = \{\ell_1, \ldots, \ell_m\}$ of prime numbers;
  - For each prime number ℓ_i, a positive direction chosen at random.
- Protocol
  1. Alice chooses a random succession ρ_A of elements in L and Bob chooses a random succession ρ_B of elements in L;
  2. Alice computes $E_A = \rho_A(E)$ and sends it to Bob;
  3. Bob computes $E_B = \rho_B(E)$ and sends it to Alice;

Ordinary Case

University of Zurich[UZH]

Key exchange protocol (Rostovtsev-Stolbunov, 2006)

- Public parameters
  - A large finite field $\mathbb{F}_q$ and an ordinary elliptic curve E over $\mathbb{F}_q$;
  - A set $L = \{\ell_1, \ldots, \ell_m\}$ of prime numbers;
  - For each prime number ℓ_i, a positive direction chosen at random.
- Protocol
  1. Alice chooses a random succession ρ_A of elements in L and Bob chooses a random succession ρ_B of elements in L;
  2. Alice computes $E_A = \rho_A(E)$ and sends it to Bob;
  3. Bob computes $E_B = \rho_B(E)$ and sends it to Alice;
  4. Alice computes $E_{AB} = \rho_A(E_B)$ and Bob computes $E_{AB} = \rho_B(E_A)$.

Ordinary Case

Key exchange protocol (Rostovtsev-Stolbunov, 2006)

- Public parameters
  - A large finite field $\mathbb{F}_q$ and an ordinary elliptic curve E over $\mathbb{F}_q$;
  - A set $L = \{\ell_1, \ldots, \ell_m\}$ of prime numbers;
  - For each prime number ℓ_i, a positive direction chosen at random.
- Protocol
  1. Alice chooses a random succession ρ_A of elements in L and Bob chooses a random succession ρ_B of elements in L;
  2. Alice computes $E_A = \rho_A(E)$ and sends it to Bob;
  3. Bob computes $E_B = \rho_B(E)$ and sends it to Alice;
  4. Alice computes $E_{AB} = \rho_A(E_B)$ and Bob computes $E_{AB} = \rho_B(E_A)$.

N.B. The cryptosystem works because we are in a commutative environment.

Supersingular Case

In the supersingular case:

- The j-invariants (and so the vertices of the isogeny-graph) are elements in $\mathbb{F}_{p^2}$;

Supersingular Case

In the supersingular case:

- The j-invariants (and so the vertices of the isogeny-graph) are elements in $\mathbb{F}_{p^2}$;
- The isogeny graph is a Ramanujan graph.

In the supersingular case:

- The j-invariants (and so the vertices of the isogeny-graph) are elements in $\mathbb{F}_{p^2}$;
- The isogeny graph is a Ramanujan graph.

N.B. Since we are in a noncommutative environment, Rostovtsev-Stolbunov protocol **does not work**.

Supersingular Case

In the supersingular case:

- The j-invariants (and so the vertices of the isogeny-graph) are elements in $\mathbb{F}_{p^2}$;
- The isogeny graph is a Ramanujan graph.

N.B. Since we are in a noncommutative environment, Rostovtsev-Stolbunov protocol **does not work**.

> 2011: De Feo and Jao propose Supersingular Isogeny Diffie Hellman (SIDH), but to make it work they need to make extra information public.

Supersingular Case

In the supersingular case:

- The j-invariants (and so the vertices of the isogeny-graph) are elements in $\mathbb{F}_{p^2}$;
- The isogeny graph is a Ramanujan graph.

N.B. Since we are in a noncommutative environment, Rostovtsev-Stolbunov protocol **does not work**.

2011: De Feo and Jao propose Supersingular Isogeny Diffie Hellman (SIDH), but to make it work they need to make extra information public.

2022: Castryck and Decru use these extra information to broke the cryptosystem.

Supersingular Case

Restriction to $\mathbb{F}_p$

If we consider just the j-invariants in $\mathbb{F}_p$ and the ℓ-isogenies defined over $\mathbb{F}_p$, then the corresponding isogeny graph is a volcano.

Supersingular Case

Restriction to $\mathbb{F}_p$

If we consider just the j-invariants in $\mathbb{F}_p$ and the ℓ-isogenies defined over $\mathbb{F}_p$, then the corresponding isogeny graph is a volcano.

In particular, under this restriction, we can apply the Rostovsev-Stolbunov protocol!

Supersingular Case

Restriction to $\mathbb{F}_p$

If we consider just the j-invariants in $\mathbb{F}_p$ and the ℓ-isogenies defined over $\mathbb{F}_p$, then the corresponding isogeny graph is a volcano.

In particular, under this restriction, we can apply the Rostovsev-Stolbunov protocol!

N.B. The problem with the RS protocol on ordinary elliptic curves is that it takes several minutes per key exchange.

Supersingular Case

Restriction to $\mathbb{F}_p$

If we consider just the j-invariants in $\mathbb{F}_p$ and the ℓ-isogenies defined over $\mathbb{F}_p$, then the corresponding isogeny graph is a volcano.

In particular, under this restriction, we can apply the Rostovsev-Stolbunov protocol!

N.B. The problem with the RS protocol on ordinary elliptic curves is that it takes several minutes per key exchange. In the supersingular case this efficiency problem does not occur!

Thanks for your attention!