The Report Committee for Rachelle H. David
certifies that this is the approved version of the following report:

Lattice Based and Isogeny Based Post-Quantum Cryptography

SUPERVISING COMMITTEE:

Vijay Garg, Supervisor

Michael Orshansky

Lattice Based and Isogeny Based Post-Quantum Cryptography

by

Rachelle H. David

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December **2023**

Abstract

Lattice Based and Isogeny Based Post-Quantum Cryptography

Rachelle H. David, MSE
The University of Texas at Austin, 2023

SUPERVISOR: Vijay Garg

Modern cryptography involving public-keys relies on mathematically difficult problems, such as factoring large numbers into prime factors and finding rational points on elliptic curves. A typical public-key scheme involves two communicators who each have a public and a private key. Each party publishes their public key so that others may use it to encrypt directed messages while only the individual party uses their private key to decrypt those messages.

As computers advance, so too do the methods of breaking cryptographic systems. This drives the development of increasingly difficult cryptographic schemes. While these schemes are more difficult to break and are faster than historic encryption methods, quantum computers threaten the security of classical cryptography. These highly efficient computers will break existing classical ciphers, such as RSA (Rivest Shamir Adleman), ECC (elliptic curve cryptography) and others. It is therefore necessary and urgent to improve cryptographic algorithms to make them resistant to quantum computers. These modifications improve the security of cryptographic schemes and make it more challenging for adversaries to intercept, modify, or decrypt confidential messages.

There are currently several areas of research for potential post-quantum cryptographic algorithms. Two such areas are isogeny based cryptography and lattice based cryptography. This kind of cryptography relies on isogenies of elliptic curves as well as lattices and works by each communicating party taking random walks on isogeny graphs. In this report we explain in detail how to find isogenies of elliptic curves, how we can compute isogeny graphs from isogenies of elliptic curves and practical applications of isogeny-based cryptography in the Diffie-Helman key exchange, and lastly analyze the security of the quantum-resistant cryptographic technique. Since this is a new area of research there is no available book which covers in detail all the tools used in isogeny based and lattice based cryptography, so this report is an initiative toward formalization.

As quantum computers become more realistic threats to classical cryptography, there is a clear need to develop practical quantum-resistant algorithms. By better understanding and improving cryptographic schemes, our communications in public channels will be better protected from adversaries.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Modern communication relies on the ability to send secure, private messages in public channels. This need for secure network traffic calls for mathematically verified cryptography algorithms. Cryptography is by definition the study of secure communication techniques under the assumption that an adversaries and other third parties are present.

The two major types of cryptosystems are symmetric and asymmetric key cryptography. Symmetric involves a single private key. It requires two communicators to determine their secret key prior to communicating publicly, and often incorporates a meetup. Asymmetric cryptography involves multiple public-keys that allow for two parties to communicate confidentially on a public channel without a prior meeting. In this report we will focus on public-key cryptography.

The basis of public-key cryptography is mathematically difficult problems (J. Hoffstein and Silverman (2010)). Some such problems are factoring large numbers into two prime factors and finding rational points on elliptic curves. A typical public-key scheme involves two communicators who each have a public and a private key. Each party publishes their public key so that others may use it to encrypt directed messages while only the individual party uses their private key to decrypt those messages.

The type of asymmetric cryptography in which messages are computed across elliptic curves in finite fields is known as Elliptic Curve Cryptography (ECC) and it relies on the fact that elliptic curves over finite fields form an Abelian group via the addition operator (Cohen and Frey (2006)). Isogenies of elliptic curves provide another complexity to the algorithm: isogeny-based cryptography (Costello (2016)). In contrast, another asymmetric cryptography protocol called Lattice-Based Cryptography leverages the hardness of the shortest vector problem (SVP) (Alwen (2018)).

While isogeny based cryptography is based on elliptic curves and is believed to be an appropriate cryptosystem for key exchange, lattice based cryptography is better for digital signatures. These concepts will be further explained and explored in this report.

Computers are becoming increasingly advanced and the age of Quantum Computing is approaching, threatening current cryptographic schemes. In response, software engineers and mathematicians are collaborating on post-quantum cryptography. That is, cryptographic schemes on classical computers that are resistant to quantum computer attacks. This report will explore multiple methods of post-quantum cryptography including isogeny based cryptography and lattice based cryptography.

Chapter 2: Historic Context

Governments and militaries are particularly interested in securing their communications against adversaries. Although the government most uses Advanced Encryption Standard (AES), a symmetric block based cipher, which is believed to be quantum resistant (with a little modification of the size of the key), public key cryptography is used extensively (Morris Dworkin and Dray (2001)). In order to meet the needs of military security measures, cryptographic schemes must be reliable, rapid, flexible, and economical (of Defense (2022)).

With military messages potentially disclosing sensitive information, reliability is critical. The decrypted message must be accurate to the original message. Additionally, the security of the system must not rely in the secrecy of the key, but rather in the system complexity itself. Military communications are often time-sensitive so the encrypting and decrypting computational speed must be rapid. Since military needs are situationally dependent, the communication security must be able to adapt to all circumstances and message types. Finally, since the military must operate at a large scale, the cryptograhpic scheme must be economical to deploy.

As militaries and governments need reliable communication systems, they are very interested in maintaining the security of those systems and must remain at the forefront of those technological innovations. In order to assist with understanding this development, table 2.1 shows a timeline depicting the progress of modern encryption mechanisms (David (2019), Damico (2009)).

Since World War II, cryptographic schemes have become more complex and reliant on computational power (Damico (2009)). The government even has an organization that approves cryptographic schemes, the National Institute of Standards and Technology (NIST). As a case study, NIST endorses the use of AES (Morris Dworkin and Dray (2001)). Currently, the only method of breaking AES is a brute force at-

Year	Event
1976	Diffie-Helman asymmetric cryptography article published
1982	First theoretical framework for quantum computer published (Benioff)
1985	Koblitz and Miller independently propose elliptic curves in cryptography
1986	First universal quantum computer mathematically described (Deutsch)
1995	US Army launches first workshop on quantum computing and quantum cryptography
2001	Advanced Encryption Standard (AES) published in NIST
2004	Elliptic curve cryptography widely deployed
2014	NSA begins to develop quantum capabilities specific for cryptography
2015	NSA announces plans to transition to quantum-resistant algorithms
2016	NIST calls for quantum-secure submissions
2017	Isogeny based cryptography first explored as a post quantum scheme
2018	National Quantum Initiative Act signed into US law
2019	IBM releases first commercial quantum computer, the IBM Q System One

Table 2.1: Timeline of modern encryption mechanisms with a focus on post-quantum cryptography.

tack. On a classical computer, a brute force method would take $2n$ computational time, where n is the key-space. On quantum computers, it has been proven in Grover (1996) that applying Grover's algorithm to breaking a symmetric key algorithm by brute force requires $n/2$ computational time. This potential attack against AES effectively halves the key size as compared to the classical computing power. As a result, with appropriate key length, the AES algorithm resists attacks launched from quantum computers. The impact of specific quantum computing algorithms will be further explored in this paper.

Although AES is still classified as quantum-resistant, the threat of quantum computers breaking common classical ciphers is real. In recent years, many deployed cryptographic systems were analytically broken, often due to faulty key storage mechanisms or simply advanced mathematical and computational research (Litinski (2023)). Furthermore, the threat of quantum computing breaking currently used ciphers merits exploring alternatives in quantum-resistant algorithms and post-quantum cryptography.

This report is a survey of techniques for cryptographic communication. It specifically examines two interest area for quantum resistant algorithms, isogeny based cryptography and lattice based cryptography. The report explains how to find isogenies of elliptic curves, provides a practical application of isogeny-based cryptography in the Diffie-Helman key exchange, and compares it to the existing applications of the Diffie-Helman procedure. Then, the report examines how to find lattices and how they are used in lattice-based cryptography.

For the sake of clarity, a distinction must now be made: Quantum cryptography is not equivalent to post-quantum cryptography. While quantum cryptography describes the type of cryptography a quantum computer performs, post-quantum cryptography describes the cryptographic schemes a classical computer performs in response to a quantum computer threat and during the age of quantum machines.

Chapter 3: Elliptic Curve Cryptography

3.1 Elliptic Curves

In order to understand Elliptic Curve Cryptography (ECC), we must first better understand elliptic curves and the group operations on them (J. Hoffstein and Silverman (2010)).

An elliptic curve is a type of algebraic curve described by the equation $y^2 = x^3 + ax + b$ where $4a^3 + 27b^2 \neq 0$. It is important to observe that this curve is non-singular, meaning it has no repeated roots. In consideration of group theory, an elliptic curve can be made into an Abelian group if we let the point at infinity play the role of the identity of the group and define addition on the points of the curve as below. Theoretically any point on the curve can be assigned to be the identity of the group but then addition also will have to be modified accordingly. In most of the literature elliptic curves are notation-ally represented as

$$E = \{(x, y) \in k^2 | y^2 = x^3 + ax + b, 4a^3 + 27b^2 \neq 0\} \cup \{\mathbf{O}\},$$

where k is a field. From the equation of the curve, one can notice that a key characteristic of elliptic curves is that they are always symmetric about the x-axis. In the real plane, the elliptic curve is smooth as depicted in figure 3.1. In contrast, in the complex plane, the elliptic curve is a torus, as depicted in figure 3.2.

We will start describing operations across elliptic curves with addition. Let $P = (x_p, y_p)$, and $Q = (x_q, y_q)$ be points on a given elliptic curve E. Addition of points on the elliptic curve is defined by three basic cases. When adding across the curve, we obtain a third point $R = (x_r, y_r)$, on E such that $P \oplus Q = R$.

1. P and Q are distinct
2. $P = -Q$, meaning $x_p = x_q$ but $y_p = -y_q$

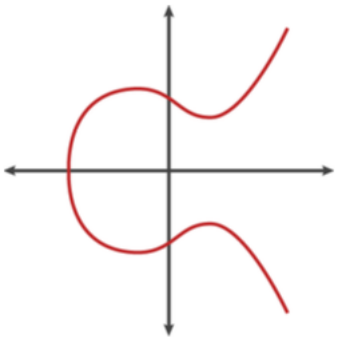

Figure 3.1: Elliptic Curve in the Real Plane

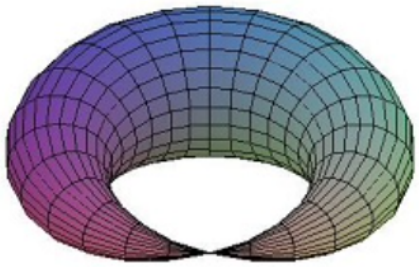

Figure 3.2: Elliptic Curve in the Complex Plane

3. $P = Q$

Case 1 ($x_p \neq x_q$): If a line is drawn between P and Q, then it is guaranteed that it will intersect the curve E at a third point R. As a result, the equation of this line is $y = \lambda x + \Upsilon$ where $\lambda = \frac{y_q - y_p}{x_q - x_p}$. For the elliptic curve addition, $P \oplus Q = R$ where $R = (x_r, y_r)$ such that.

$$x_r = \lambda^2 - x_p - x_q \text{ and } y_r = \lambda(x_p - x_r) - y_p$$

Case 2 ($x_p = x_q$ but $y_p = -y_q$): Since $P \oplus Q = P \oplus (-P)$, $P \oplus Q = 0$.

Case 3 ($x_p = x_q$ and $y_p = y_q$): In this case, the line is actually the tangent line at P so there are two further more cases:

If $y_p = 0$, $P = -P$ so we return to Case 2.

If $y_p \neq 0$, then using $\lambda = \frac{3y_p^2 + a}{2y_p}$,

$$x_r = \lambda^2 - 2x_p \text{ and } y_r = \lambda(x_p - x_r) - y_p$$

When adding across the curve, it is guaranteed that each line will intersect the curve E three times. Using that guarantee, addition on the curve is accomplished by drawing a line connecting the two points P and Q, obtaining the third point of intersection on the elliptic curve R', and taking its inverse with respect to the x-axis, R. This is graphically represented in figure 3.3. Examples of these computations can be found in David (2019).

Now that we better understand operations over elliptic curves, we must define the points on an elliptic curve as an Abelian group (Silverman and Tate (2015)).

Theorem. *The set of points on an Elliptic Curve constitute an Abelian group.*

Proof. For an elliptic curve E, the set of points in E have the rule $\oplus$. The elements $a, b \in E$ are combined to obtain an element $a \oplus b = c \in E$. The operator, $\oplus$ must have four specific properties in order to be an Abelian group:

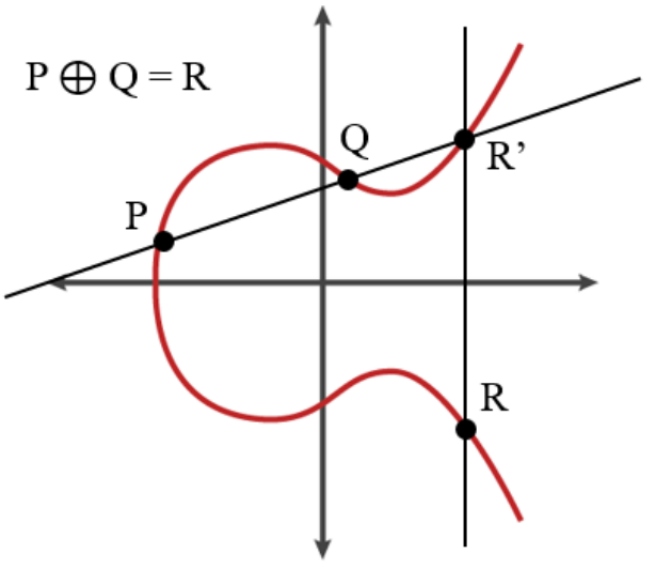

Figure 3.3: Addition Across an Elliptic Curve

1. Identity: $\forall P \in E$, we have that $P \oplus \mathbf{O} = \mathbf{O} \oplus P = P$.
   Because we define $\mathbf{O}$ to lie on all vertical lines, and further define it to be equal to P added to its inverse, $\mathbf{O}$ serves as the identity element for elliptic curves. Thus the identity property is upheld.

2. Inverse: $\forall P \in E$, there exists a $-P$ such that $P \oplus (-P) = \mathbf{0}$. Then, $P \oplus (-P) = \mathbf{O}$. By definition of $\mathbf{O}$, it is equal to P added to its inverse. Thus the inverse property is upheld.

3. Associative: $\forall P, Q, R \in E$, $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$.
   Proving the Associative Law is rather complicated for elliptic curves. By drawing the points on a curve, this law does not seem obvious and for that reason, here we provide a brief discussion of the cases and how they relate to association:
   If $P = \mathbf{O}$, then this case is trivial.

$$(\mathbf{O} \oplus Q) \oplus R = \mathbf{O} \oplus (Q \oplus R)$$

$$= Q \oplus R = Q \oplus R$$

If $P = -Q$, assuming Q and R are distinct, then this case also simplifies but it is slightly more involved. Without exhausting these cases, it is still appropriate to conclude that the associative property is upheld. A more rigorous proof for this can be found in Sutherland (2015)

4. Commutative: $\forall P, Q \in E,\ P \oplus Q = Q \oplus P$.
   Since the line going through P and Q is the same as the line going through Q and P, the order of the points does not matter. Thus the commutative property is upheld.

Therefore, elliptic curves satisfy each of the requirements and thus is an Abelian group. ■

3.2 Elliptic Curves Over Finite Fields

In order to use elliptic curves for cryptography, we must consider elliptic curves whose points have coordinates in a finite field $\mathbb{F}_p$ (Silverman and Tate (2015)).

Let E be an elliptic curve over $\mathbb{F}_p$ such that $E : Y^2 = X^3 + AX + B$ where $A, B \in \mathbb{F}_p$ and $4A^3 + 27B^2 \neq 0 \mod p$. Additionally, we will denote the points on E with coordinates in $\mathbb{F}_p$ by

$$E(\mathbb{F}_p) = \{(x, y) : x, y \in \mathbb{F}_p | y^2 = x^3 + Ax + B\} \cup \{\mathbf{O}\}$$

The critical part is that every point is computed modulo p. To demonstrate this point, we use the following example.

Example. *Find all the points on the elliptic curve $E : Y^2 = X^3 + 3X + 8$ over the field $\mathbb{F}_{13}$.*

The set of all possible X-coordinates of points on this curve is $X = \{0, 1, 2, 3, ..., 12\}$. In order to find all the points of $E(\mathbb{F}_{13})$, we must check each possible X value and check whether $X^3 + 3X + 8$ is a square, modulo 13. First we check $X = 0$:

$$(0)^3 + 3(0) + 8 = 8$$

Since 8 is not a square modulo 13, there are no points with the x-coordinate of 0. Next we check $X = 1$:

$$(1)^3 + 3(1) + 8 = 12$$

Now, 12 is a square modulo 13 and has two corresponding y-coordinate values.

$$5^2 \equiv 12 \bmod 13 \textit{ and } 8^2 \equiv 12 \bmod 13$$

Iterating though each option will yield the following result:

$$E(\mathbb{F}_{13}) = \{\mathbf{0}, (1,5), (1,8), (2,3), (2,10), (9,6), (9,7), (12,2), (12,11)\}$$

These nine points are all the points on this elliptic curve over $\mathbb{F}_{13}$.

Addition on the points of an elliptic curve defined over a finite field is further defined with precision in the following cases. Suppose there exists an elliptic curve E over a finite field $\mathbb{F}_p$ such that $P, Q \in E$.

Case 1: $P \oplus \mathbf{0} = \mathbf{0} \oplus P = P$

Case 2: If $Q = -P \bmod p$, then $P \oplus Q = \mathbf{0}$

Case 3: For all other cases, $P \oplus Q = R = (x_r, y_r)$ where

$$x_r = \lambda^2 - x_p - x_q \bmod p$$

$$y_r = \lambda(x_p - x_r) - y_p \bmod p$$

$$\lambda = \begin{cases} \frac{y_q - y_p}{x_q - x_p} (\bmod\ p) & P \not\equiv Q \bmod p \\ \frac{3x_p^2 + a}{2y_p} (\bmod\ p) & P \equiv Q \bmod p \end{cases}$$

Note: a property of the modulus operator is that $\frac{a}{b} \bmod p \equiv (a)(b^{-1}) \bmod p$ with b^{-1} such that $(b)(b^{-1}) \equiv 1 \bmod p$.

Theorem. The points on $E(\mathbb{F}_p)$ form an Abelian group (Silverman and Tate (2015)).

Proof. Let E be an elliptic curve over a finite field $\mathbb{F}_p$. Since the addition of points on $E(\mathbb{F}_p)$ are derived from the equation for $E(\mathbb{F}_p)$, the resulting point must also be in $E(\mathbb{F}_p)$. Thus the laws of addition for points on an elliptic curve also apply to elliptic curves over a finite field.

The identity element being the point at infinity, $\mathbf{O}$, is upheld as a result of case 1 of point addition. The inverse property is evident from case 2 of point addition. Deeply testing the addition equations for the various special cases that fall under case 3 will result in an understanding that the associative law is upheld. Examining the addition equations will demonstrate that swapping the order of addition of two points will not impact the resulting point. Thus, the commutative property is satisfied. Therefore, the points on $E(\mathbb{F}_p)$ form an Abelian group. ■

Now that we have an understanding of operations and group classification for elliptic curves over finite fields, we are prepared to discuss the number of points on an elliptic curve.

If E is an elliptic curve over the finite field $\mathbb{F}_p$, we already know that $E(\mathbb{F}_p)$ is a finite group and we will let $\#E(\mathbb{F}_p)$ be the number of points of E over $\mathbb{F}_p$. The following theorem is called Hasse's Theorem and it is crucial in elliptic curve cryptography (Cohen and Frey (2006)).

Theorem (Hasse's Theorem). *Let E be an elliptic curve over the finite field $\mathbb{F}_p$, then*

$$|\#E(\mathbb{F}_p) - (p+1)| \leq 2\sqrt{p}$$

For large values of $q = p^n$, the number of points on the curve is in a narrow

range of width $4\sqrt{q}$, around $(q+1)$. This provides some insight into how to pick elements of $E(\mathbb{F}_q)$ with an almost uniform distribution.

Below is an algorithm of how to find a random point in $E(\mathbb{F}_q)$ (Cohen and Frey (2006)).

Algorithm 1 Find a random point in $E(\mathbb{F}_q)$

Input Elliptic curve E over $\mathbb{F}_q$
Output Random point $P \in E(\mathbb{F}_q)$
`Step 1:` Pick a random $x \in \mathbb{F}_q$
`Step 2:` Substitute x in the equation of E
`Step 3:` Solve the quadratic equation in y (i.e. by computing the Legendre symbol)
`Step 4:` If solutions are found, randomly decide which y to choose and let $P = (x, y)$. Return (P). $=0$

Now we have all the tools to discuss elliptic curve cryptography in the context of the Diffie Helman key exchange.

3.3 Elliptic Curve Cryptography

Elliptic curve cryptography (ECC) is used in public-key cryptography and is primarily used in key agreement and digital signature verification. As a simplified idea of how elliptic curves are used to construct a cryptosystems, we will use the discrete log problem (DLP) on the group of points of an elliptic curve and will denote this by the elliptic curve discrete log problem (ECDLP) (Litinski (2023)).

3.3.1 Elliptic Curve Discrete Log Problem

The discrete log problem (DLP) is defined as: given a group G, a generator g of the group and an element h of G, find the discrete logarithm to the base g of h in the group G. A similar difficulty can be accomplished by using the group of points $E(\mathbb{F}_p)$ on an elliptic curve E over a finite field $\mathbb{F}_p$ (Cohen and Frey (2006)). Given two points P, the generator of the group, and Q in $E(\mathbb{F}_p)$, the discrete log problem

for the group of points on an elliptic curve is to find n such that

$$Q = nP$$

Trying to solve for n is translated as solving the DLP for elliptic curves.

With elliptic curves over finite fields, there is no analog method of index calculations that would solve the DLP. As a result, cryptographers may use smaller prime numbers in their calculations, improving the computational efficiency without sacrificing security. As a result, the elliptic curve discrete log problem, based on the addition of rational points on an elliptic curve, is the backbone of elliptic curve cryptography.

On a classical computer with classical computer adversaries, the elliptic curve discrete log problem is sufficiently difficult, however, with the threat of quantum computers, the discrete log problem is no longer secure. There are known algorithms that will operate on a quantum computer and with sufficient efficiency break the elliptic curve discrete log problem. As a result, researchers are working on improving existing cryptographic schemes in order to make them resistant to quantum machine attacks.

3.3.2 Elliptic Curve Diffie-Helman Key Exchange

The Elliptic Curve Diffie-Helman (ECDH) key exchange follows this algorithm: Alice and Bob agree on a particular elliptic curve $E(\mathbb{F}_p)$ and a specific point on the curve $P \in E(\mathbb{F}_p)$ and publicize these values (Litinski (2023)). Alice and Bob each must select the secret integers n_A and n_B respectively and compute the appropriate values

$$Q_A = n_A P \text{ and } Q_B = n_B P$$

where $n_A P$ is the n_A-th multiplication of point P. Alice and Bob exchange the values Q_A and Q_B, which are points on the elliptic curve E, on the public channel.

After receiving the other's public value, each uses their secret multiple to compute the final secret value. Alice computes $n_A Q_B$ and Bob computes $n_B Q_A$. These values are equivalent because

$$n_A Q_B = (n_A n_B)P = n_B Q_A$$

and the result is the secret shared key. For more details about this algorithm in its coded form with associated elliptic curve initializations, please refer to Appendix B and is visually depicted in table 3.1. Now Alice and Bob can exchange keys and communicate secretly on the public channel!

The only way for an adversary Eve to find Alice and Bob's secret key is for her use the elliptic curve E and the three points on it P, Q_A, and Q_B to solve the Elliptic Curve Discrete Log Problem. If Eve solved this problem, then she would be able to uncover both secret multiples and subsequently calculate the secret shared key. This problem is analogous to the problem Eve faces with the Diffie-Helman key exchange for integers over a finite field. Yet, calculations along the elliptic curve are much quicker and more efficient while maintaining a large key space. However, this cryptographic scheme breaks under quantum computations and therefore is not quantum resistant.

Public Parameters	
Finite field $\mathbb{F}_p$ Elliptic curve $E/\mathbb{F}_p$, such that $\#E(\mathbb{F}_p)$ is prime A generator P of $E(\mathbb{F}_p)$	
Alice	Bob
Pick a random secret value	
$0 < a < \#E(\mathbb{F}_p)$	$0 < b < \#E(\mathbb{F}_p)$
Compare public data	
$A = [a]P$	$B = [b]P$
Exchange data	
$A \rightarrow$	$\leftarrow B$
Compute shared secret key	
$S = [a]B$	$S = [b]A$

Table 3.1: Elliptic Curve Diffie Helman Key Exchange

Chapter 4: Lattices and Cryptography

4.1 Lattices

A lattice is an abstract structure that is often studied as part of order theory. It is very similar to a vector space, with the exception that lattices consist of only discrete vectors. While a vector space contains real-valued vectors, lattice vector elements have discrete values. Necessary terms in lattice theory are described below. In addition to these terms, further definitions for lattice related and matrix terminology can be found in Dong Pyo Chi and Kim (2015) and Li et al. (2022).

Formally, let v_1,...$v_n \in \mathbb{R}^m$ be a set of linearly independent vectors. A ***lattice*** L generated by $v_1, ...v_n$ is the set of integer linear combinations of the set of vectors:

$$L = a_1v_1 + ... + a_nv_n | a_1, ...a_n \in \mathbb{Z}$$

.

As we can observe from this definition, the difference between lattices and vector spaces is that the coefficients are integers. Thus we call these *integer lattices*. In this definition, we call the value m the *dimension* and value n the *rank* of the lattice. When $m = n$, the lattice is considered *full-rank*. In the context of cryptography, we only consider full-rank lattices. For a visual depiction, figure 4.1 is a lattice in $\mathbb{R}^2$.

A *basis* of a lattice L is a set of linearly independent vectors $B = \{b_1, ...b_n\}$ that spans the lattice. We can describe this as

$$L(B) = \{z_1b_1 + ... + z_nb_n | z_i \in \mathbb{Z}\}$$

.

When translating between different lattice bases, we use unimodular matrices. These matrices are also sometimes used in lattice basis reduction. For context, a matrix $A \in \mathbb{Z}n \times n$ is unimodular if it has a multiplicative inverse in $\mathbb{Z}n \times n$. Meaning, $A \in \mathbb{Z}n \times n$ is unimodular if and only if $A^{-1} \in \mathbb{Z}n \times n$.

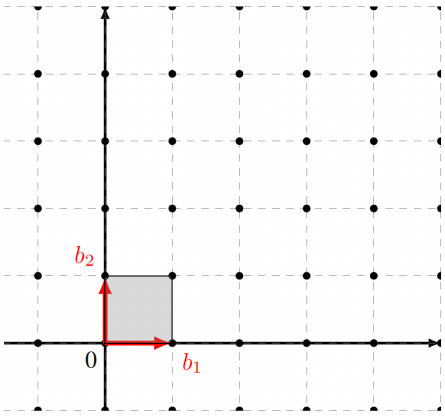

Figure 4.1: A lattice L with a basis $B = \{b_1, b_2\}$ and its fundamental domain F. Li et al. (2022)

As an example, let L be the lattice in figure 4.1. When we apply the relation $B' = AB$ where the change of basis matrix $A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$ is unimodular, the vectors $\{b'_1, b'_2\}$ form a different basis for L. This translation is visually depicted in figure 4.2.

Lastly, we need to establish the *fundamental domain* of a lattice. This concept is somewhat related to the sparsity of a lattice. A fundamental domain of an n-dimensional lattice L with a basis $\{v_1, ...v_n\}$ is

$$F(v_1, ...v_n) = \{t_1 v_1 + ... + t_n v_n | t_1 \in [0, 1)\}$$

. In figures 4.1 and 4.2, the fundamental domain is depicted by the domain shaded grey.

4.2 Lattice Based Cryptography

Similar to elliptic curve cryptography relying on the elliptic curve discrete log problem, lattice based cryptographic schemes rely on the hardness of lattice problems. The basic version of these problems is the shortest vector problem (SVP) (Li et al. (2022)). Given as input a lattice represented by an arbitrary basis $\mathbf{B}$, output the shortest nonzero vector in it. One variant of this problem is the approximation

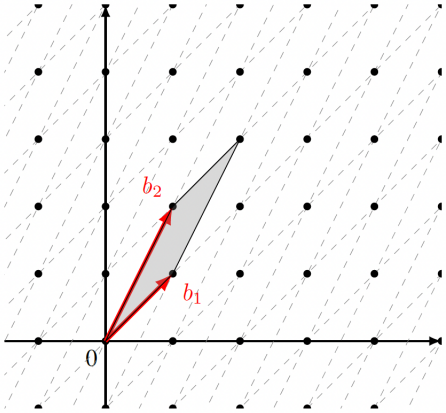

Figure 4.2: The same lattice L with a different basis $B' = \{b'_1, b'_2\}$ and its fundamental domain F', where $B' = AB$ for a unimodular change of basis matrix $A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$ Li et al. (2022)

variant of SVP. In this problem, the goal is to output a lattice vector whose length is at most some approximation factor times the length of the shortest nonzero vector.

Other hard lattice problems are the closest vector problem (CVP) and the Shortest Independent Vectors Problem (SIVP) (Micciancio and Regev (2008)). CVP is the problem of given a lattice basis $\mathbf{B}$ and a target vector $\mathbf{t}$, the goal is to find the lattice point $\mathbf{v} \in L(\mathbf{B})$. It is worth noting that that target vector $\mathbf{t}$ does not necessarily have to be in the lattice L. Alternatively, SIVP's goal is given a lattice basis $\mathbf{B} \in \mathbb{Z}^{n \times n}$ to find n linearly independent lattice vectors, minimizing the quantity of those vectors. Even for these two alternative hard lattice problems, the approximation invariant is typically applied.

The most well known and studied algorithm for lattice problems is the LLL algorithm, developed in 1982 by Lenstra, Lenstra, and Lovasz (Micciancio and Regev (2008)). This polynomial time algorithm for SVP achieves an approximation factor of $2^{O(n)}$ where n is the dimension of the lattice. While this may seem grossly inefficient, the LLL algorithm is still useful, with many applications in factoring, programming, and attacking cryptosystems.

There are extensions of the LLL algorithm leading to slightly better approximation factors, leveraging the concept of replacing the core of the LLL algorithm, with blocks of a larger size. However, this improvement in the approximation factor, leading to shorter vectors, also results in an increased running time.

Conjecture. *There is no polynomial time algorithm that approximates lattice problems to within polynomial factors.*

This means that approximating lattice problems to within polynomial factors is a hard problem and thus is suitable for designing cryptosystems. Furthermore, as circumstantial evidence for this conjecture, it is evident that progress in lattice algorithms has been stubbornly difficult, with no significant improvement in performance since the 1980s. In contrast, other areas of research into factoring problems have had many breakthroughs.

When applied to "real-world" lattices or at least lattices chosen randomly from some natural distribution, lattice reduction algorithms perform better than their worst-case theoretical performance (Alwen (2018)). While this application is not fully explained yet, known lattice reduction algorithms provide a sufficient approximation ratio. This experimental finding builds confidence in the application of lattices in secure cryptography.

Chapter 5: Quantum Computing

Quantum computers do not use classical bits, but rather use quantum bits known as qubits, which process information differently from classical computers (Faculty (2023)). Qubits are different from bits in three meaningful ways or principles:

1. Superposition
2. Entanglement
3. Uncertainty

In contrast to a classical bit, which can either be at a state of 0 or 1, a qubit can be at a *superposition* of both one and zero simultaneously, or at least until it is measured. Further, two particles can become *entangled*, meaning their states remain connected even when separated. The distances is irrelevant once the particles are quantumly linked in this way. Finally, since subatomic particles behave like waves, Heisenberg's uncertainty principle still applies. It states that we cannot measure both the position and speed of a particle. Instead, the more we know about one, the less we know about the other. This mean that we cannot measure with accuracy both the position and speed of a qubit. As a result of these properties, the computational efficiency of quantum computers not only risks classical computing, but also risks existing cryptographic schemes.

There are two applicable quantum algorithms that greatly impact and threaten current classical cryptography, Shor's algorithm and Grover's algorithm. Since the algorithms require more background knowledge on quantum computers from a physics and mechanics perspective, we will merely describe the impacts of these algorithms and more information can be found on them at Chung (2017), Shor (1994), and Grover (1996).

5.1 Shor's Algorithm

Shor's algorithms provide the discrete log and the factoring problems with a computation complexity of random polynomial time on a quantum computer. As a result, Shor's algorithm enables solving the discrete log problem much faster than classical algorithms. Since RSA and ECC are both based on the hardness of the discrete log problem, Shor's algorithm poses a threat to all modern cryptographic schemes that rely on these paradigms. With this risk, there is a clear need to expand modern cryptographic schemes beyond relying on the difficulty of the factoring problem.

5.2 Grover's Algorithm

Grover's algorithm is a search optimizer. It performs a search over an unordered set of elements in order to find a unique element that satisfies a given condition. Whereas the most optimal classical search algorithms execute in linear time, Grover's algorithm can run in square root complexity, a quadratic speedup. In order to achieve this speedup, the algorithm relies on the quantum superposition of states.

Since Grover's algorithm reduces the complexity of searches, fewer trials are needed to find a unique solution to a problem. This implies that brute force attacks on current cryptographic schemes would be possible in a reasonable amount of time with a quantum computer. With the approaching threat of quantum machines and their impacts on cryptographic schemes, classical cryptography is at risk and the need for post-quantum cryptography is even more evident.

Chapter 6: Post-Quantum Cryptography

As engineers enter the age of quantum computers, they need to generate cryptographic schemes to protect classical computers against the threat of quantum computers. There are currently five primary areas of study for post-quantum cryptography:

1. Lattice-Based Cryptography
2. Multivariate Cryptography
3. Hash-Based Cryptography
4. Code-Based Cryptography
5. Supersingular Elliptic Curve Isogeny-Based Cryptography

These five families of schemes are proposed to be resistant to quantum machine attacks, according to the NIST 2016 report, (L. Chen (2016)). Each family is suitable for different types of cryptographic applications. Lattice-based cryptography enables encryption applications yet is thought to be best for key exchange algorithms. It is relatively simple yet secure, but precise estimates of its security are still unknown. Multivariate polynomial cryptography is more commonly applied to digital signatures and relies on the difficulty of solving systems of multivariate polynomials over finite fields. Hash-based cryptography is used primarily in digital signatures, but offers some drawbacks since it requires the signer to keep track of exactly how many signatures they signed. Additionally, there is a limit on how many times a signer can digitally sign without increasing the signature size. Code-based cryptography is a method of detecting and correcting errors in messages and is best used in encryption schemes. NIST reports that Shor's algorithm efficiently solves the elliptic curve DLP

but there is no known algorithm to similarly solve the isogeny problem on supersingular curves (Lange (2016)). As a result, isogeny-based cryptography is another family of potential quantum resistant cryptographic schemes that is primarily used in key exchange protocols.

6.1 Post-Quantum Lattice Based Cryptography

As we previously discussed, lattice cryptography is based on the SVP, an problem for which there are currently no known quantum algorithms for solving that perform better than the best known classical algorithms.

Since Shor's algorithm, attempts to solve lattice problems by quantum algorithms have been made, but none so far have been successful. The main difficulty is that the periodicity finding technique which Shor's factoring algorithm uses does not apply to lattice problems. So, we must consider the following conjecture, which justifies the use of lattice-based cryptography for post-quantum cryptography Li et al. (2022):

Conjecture. *There is no polynomial time quantum algorithm that approximates lattice problems to within polynomial factors.*

While this may seem that lattice problems are sufficiently hard against quantum computers, however the inefficiency of these systems leads to impractical applications. The primary public key encryption scheme using lattices is GGH, proposed by Goldreich, Goldwasser, and Halevi (Micciancio and Regev (2008)). It is analogous to the McEielece cryptosystem but leverages lattices rather than decoding linear codes over finite fields.

The GGH system works in accordance with the following algorithm:

Since the error vector is sufficiently small, this algorithms is deemed correct. While no asymptotically significant attack for this algorithm is known, it is also too computationally complex to be practical. Perhaps applying the LLP technique

Algorithm 2 Lattice based Public Key GGH Cryptosystem

`Step 1:` The private key is a "good" lattice basis $\mathbf{B}$. A good basis selection is a basis consisting of short, almost orthogonal vectors.

`Step 2:` The public key $\mathbf{H}$ is a "bad" basis for the same lattice L, which means that $L(\mathbf{H}) = L(\mathbf{B})$. One suggestion for selecting this basis is the Hermite Normal Form (HNF) of $\mathbf{B}$ because it can be efficiently computed and provides a lower basis.

`Step 3:` The encryption process consists of adding a short noise vector $\mathbf{r}$ to a specifically selected lattice point $\mathbf{v}$. It is often proposed to select the vector $\mathbf{v}$ such that all the coordinates of $(\mathbf{r} + \mathbf{v})$ are reduced with the modulo operator of the corresponding element along the diagonal of the HNF public basis $\mathbf{H}$. The resulting vector $(\mathbf{r} + \mathbf{v})$ is denoted as $\mathbf{r}$ mod $\mathbf{H}$. This result provably makes cryptanalysis hardest because $\mathbf{r}$ mod $\mathbf{H}$ can be efficiently computed from any vector of the form $(\mathbf{r}+\mathbf{v})$ with $\mathbf{v} \in L(\mathbf{B})$. while also being secure against attack.

`Step 4:` The decryption problem is finding the lattice point $\mathbf{v}$ closest to the target ciphertext $\mathbf{c} = (\mathbf{r} \bmod \mathbf{H}) = \mathbf{v} + \mathbf{r}$, with some associated error vector $\mathbf{r} = \mathbf{c} - \mathbf{v}$ due to the approximation. Return =0

to this algorithm would improve its computational requirements. This exercise is left for further research. Next, we will explore the more practical Isogeny Based Cryptography.

6.2 Isogenies of Elliptic Curves

Classical elliptic curve cryptography is based on the addition of rational points on an elliptic curve over a finite field. Furthermore, classical ECC relies on the difficulty of the elliptic curve discrete log problem. In contrast, isogeny-based cryptography does not involve point addition on the curve at all. Rather it is based on supersingular isogeny graphs on which multiple elliptic curves are depicted. The relationships between isogenies of elliptic curves is based on the j-invariant of the curve (Costello (2016)).

Isogenies are maps between elliptic curves that satisfy certain properties. Let

E and E' be elliptic curves, then there exists a map $f : E \to E'$ between the curves such that if one of the following conditions is true, then all three are true:

1. f is a surjective group morphism
2. f is a group morphism with a finite kernel
3. f is a non-constant algebraic map that relates the point at infinity of E onto the point at infinity of E'

An isogeny between these two elliptic curves E and E' is a group homomorphism: there exists a map $f : E \to E'$ between the two groups such that the group operation is preserved. Most importantly, the identity element and the inverse map are both preserved. Two elliptic curves are called *isogenous* if there exists an isogeny between them. In practical cryptography, rather than exchanging points on the curve, the communicators will be exchanging particular homomorphisms.

Since Shor's algorithm breaks the elliptic curve discrete log problem, it is necessary to improve the existing elliptic curve cryptographic schemes in order to make them quantum resistant. By relying on isogenies between elliptic curves rather than addition across a single curve, it is possible to construct more complicated schemes intended to protect classical computers against quantum machines. More practically, selecting appropriate parameters for isogeny-based cryptography is rather straightforward, especially with existing algorithms to find appropriate elliptic curves.

6.3 Supersingular Isogeny Diffie-Helman Key Exchange

These tools we use in finding isogenies between curves and isogeny graphs can apply to a more secure and quantum resistant Diffie Helman procedure. Random walks in isogeny graphs are the basis for the quantum resistant isogeny based cryptography (David (2019)).

The general concept for isogenous Diffie-Helman key exchange is that Alice and Bob both agree to start at the same curve E_0 on the agreed upon isogeny graph G. Both Alice and Bob take random walks from E_0 of a secret number of steps to some curves E_A and E_B respectively. They exchange the curves E_A and E_B but keep the number of steps it took them to get there private. Then, Alice begins at curve E_B and repeats the "same" secret steps she took while Bob begins at curve E_A and repeats the "same" secret steps he took. They will both arrive at the same secret shared curve E_S and can then communicate secretly across that curve. While this method seems simple, there are a few theorized methods of implementing this algorithm and further improving its efficiency.

Supersingular isogeny graphs are very useful in generating key-exchange protocols for two main reasons. First, isogeny graphs can be obtained from one single isogeny degree. As a result, the protocols are more efficient. Second, an Abelian group does not greatly impact the graphs, so it is difficult for quantum computers to speedup the brute force problem of finding the right path. The Supersingular Isogeny Diffie-Helman (SIDH) key exchange follows this algorithm:

Alice and Bob must agree on a few more values than in the simpler protocols. First they must agree on two small primes, l_A and l_B so that Alice's graph consists of l_A-isogenies and Bob's graph will consist of l_B-isogenies. Alice selects a secret walk of length e_A, meaning she is selecting a secret cyclic subgroup $<A> \subset E[l_A^e{}_A]$. Bob similarly selects a secret walk of length e_B which is also equivalent to selecting the secret cyclic subgroup $<B> \subset E[l_B^e{}_B]$. As a result, there is a subgroup $<A>+<B> = <A, B>$, which is an isogeny to $E/<A, B>$. Also, this group $<A, B>$ is a cyclic subgroup of order $l_A^e{}_A l_B^e{}_B$. For the exchange, Alice and Bob randomly choose $<A>$ and $<B>$ in a large enough group so that they can both obtain $E/<A, B>$ without revealing their secret information.

Now that they selected their private information, Alice and Bob publicize their small primes l_A and l_B and agree on the public prime p such that $p = l_A^e{}_A l_B^e{}_B \pm 1$.

Public Parameters	
Primes l_A, l_B, and prime $p = l_{A^A}^e l_{B^B}^e$ Supersingular elliptic curve $E over \mathbb{F}_{p^2}$ A basis $<P_A, Q_A>$ of $E[l_{A^A}^e]$ A basis $<P_B, Q_B>$ of $E[l_{B^B}^e]$	
Alice	Bob
Pick a random secret subgroup $<A> = <[m_A]P_A + [n_A]Q_A>$	$<B> = <[m_B]P_B + [n_B]Q_B>$
Compute secret isogenies $\alpha : E \to E/<A>$	$\beta : E \to E/<B>$
Exchange data $E_A, \alpha(P_B), \alpha(Q_B) \to$	$\leftarrow E_B, \beta(P_A), \beta(Q_A)$
Compute shared secret key $E/<A, B> = E_B/<\beta(A)>$	$E/<A, B> = E_A/<\alpha(B)>$

Table 6.1: Supersingular Isogeny Diffie Helman Key Exchange Algorithm

They also agree on the supersingular curve E over $\mathbb{F}_{p^2}$. Finally, they publicise the fixed bases of the groups they walked to.

Now, they accomplish some computations independently. They select their random subgroups and secret isogenies. Then they publish the results of their mappings: $E_A = E/<A>$ and $E_B = E/<B>$, respectively. Then, they calculate the shared key $E/<A, B>$. This algorithm is summarized in the table 6.1.

If Eve is trying to intercept or decrypt the public messages, she would have to use a meet-in-the- middle attack. She would have to iterate through all the possible random walks of e_A steps from the starting curve E until she finds a collision. Once she finds that, she would be able to use Alice's secret information to compute the shared secret key. As a result of this risk, a larger key space would deter Eve's ability to find that collision.

Chapter 7: Conclusion

As this report explores, isogeny based cryptography and lattice based cryptography is a viable method of protecting classical computers against quantum computer attacks. More research into the efficiency of the Supersingular Isogeny Diffie-Helman key exchange is necessary. However, since supersingular isogeny graphs are rather efficient, this method is promising in generating more secure and efficient cryptosystem. In comparison to the other instances of the Diffie-Helman key exchange, the supersingular isogeny method certainly requires smaller primes as its public keys.

Although these two specific cryptoschemes are currently considered quantum resistant, there are three other primary types of cryptographic schemes are are considered to be quantum resistant. While isogeny based cryptography is most efficient in key exchange algorithms, lattice based cryptography is also primarily used in key exchange algorithms, but it relies on the shortest vector problem rather than the discrete log problem. Since lattices are complicated primitives, the associated mathematics can be complicated and inefficient in comparison to isogeny graphs.

When considering isogeny based cryptography, we first study simple elliptic curves and isogenies between them. However, there are other higher order elliptic curves that can be further explored. Such curves are hyperelliptic curves and superelliptic curves. Hyperelliptic curves are of the form $y^2 = f(x)$ where $f(x)$ is a polynomial of degree $n > 4$ that has n distinct roots. The term hyperelliptic is only used to describe specific higher order curves and is related to the genus. The genus g of a curve is and integer computed relative to the degree as $n = 2g + 1$ or $n = 2g + 2$. Superelliptic curves are similar to hyperelliptic curves, except the exponent of y must be greater than 2. Thus a superelliptic curve is of the form $y^m = f(x)$ where $m > 2$ and $f(x)$ is a polynomial of degree $n > 4$ that has n distinct roots. Currently, higher genus curves are string candidates for post-quantum cryptography.

Cryptography with higher genus curves is more complicated than with isogenies of elliptic curves because higher genus curves are not groups. In order to apply the group structure to these higher genus curves requires additional properties and computations. Still, using genus 2 hyperelliptic curves produces higher degree isogeny graphs compared to elliptic curve isogeny graphs for the same prime. As a result, genus 2 hyperelliptic curves may produce more secure cryptosystems that are quantum resistant. Since higher genus do not have equivalent theorems to elliptic curves, additional research must be conducted on their efficiency prior to incorporating them into post-quantum cryptography schemes.

The government and military are especially interested in communicating securely over public channels. Although the US Army today primarily uses AES as its cryptosystem, a currently unbreakable scheme, the threat of quantum computers breaking these classical ciphers is real. As the cyber domain grows in tactical importance, and general quantum computers become accessible, users will need to ensure their classical computers are protected against quantum attacks. Some existing schemes are not scalable to be secure post-quantum. The Rivest–Shamir–Adleman (RSA) scheme and elliptic curve cryptography (ECC) are two such cryptosystems because Shor's Algorithm breaks them in polynomial time on quantum computers. For this reason, exploring and implementing post-quantum secure cryptosystems like isogeny based and lattice based cryptography is so important.

Appendix A: Simple Diffie Helman Key Exchange

This appendix demonstrates the simple Diffie-Helman key exchange algorithm for integers over a finite field executed in python code.

```
import secrets

def diffieHelmanBasic(g, p):
    """This is a very simple implementation of Diffie Helman

    There are set public integer parameters for this exchange:
        g is a multiplicative generator
        p is a large enough prime number"""

    #alice and bob each chooses a secret integer between 0 and p-1
    asecret = secrets.randbelow(p-1)
    bsecret = secrets.randbelow(p-1)

    #both compute their public A and B
    A = g**asecret
    B = g**bsecret

    #A and B are exchanged

    #Each computes the shared key
    ashare = B**asecret
    bshare = A**bsecret

    #ensure they are the same
    if (ashare != bshare):
        print("error in calculations - key not properly exchanged")
    else:
        print("Key successfully exchanged!")
```

Figure A.1: Python code for Simple Diffie-Helman Key Exchange algorithm

Appendix B: Another Appendix

This appendix includes two images: the helper functions and the Elliptic Curve Diffie-Helman algorithm in python. The helper functions are used in initializing an elliptic curve in python code to include elliptiic curve verification, addition of points on the curve, and multiplication-by-m functions. All programming is accomplished in Python3.

```
def diffieHelmanEC(FP, a, b, xg, yg):
    """This is an elliptic curve implementation of Diffie Helman

    There are set public integer parameters for this exchange:
        FP is a finite field FP with
        a, b determines an elliptic curve over the finite field, FP
            where the number of elements in E(FP) is prime
        xg, yg is a generator (base point) of E(FP) """
    #verify this is in fact an elliptic curve
    ellipticCurve(a, b)

    #first lets calculate the number of elements in E(FP)
    numEF = 5 #***

    #alice and bob each chooses a secret integer between 0 and #E(FP)
    asecret = secrets.randbelow(numEF)
    bsecret = secrets.randbelow(numEF)

    #both compute their public points A and B
    (xa, ya) = addToSelf(xg, yg, a, asecret)
    (xb, yb) = addToSelf(xg, yg, a, bsecret)

    #A and B are exchanged

    #Each computes the shared point key
    ashare = addToSelf(xb, yb, a, asecret)
    bshare = addToSelf(xa, ya, a, bsecret)

    #ensure they are the same
    if (ashare != bshare):
        print("error in calculations - key not properly exchanged")
    else:
        print("Key successfully exchanged!")
```

Figure B.1: Python code for implementing Elliptic Curve Diffie-Helman Key Exchange algorithm

```python
def ellipticCurve(a, b):
    # assume it is already in the Weierstrass form
    #that is, y^2 = x^3 + ax + b
    #here we check to ensure that it is a usable EC
    if (discriminantCheck(a, b)):
        print("using EC y^2 = x^3 + %d x + %d" % (a, b))
    else:
        print("Improper EC because discriminant = 0")

def discriminantCheck(a, b):
    #this is just a helper function in elliptic curve definition
    dis = 4 * a*a*a + 27 * b * b
    if (dis != 0):
        return True
    else:
        return False

def testPoint(a, b, x, y):
    #this is used to ensure the point in question is on the curve
    return y*y == x*x*x + a * x + b

def addPoints(xp, yp, xq, yq, a):
    #here we actually add points on the curve!
    (xr, yr) = (0, 0)
    if (xp != xq):
        gam = (yq-yp)/(xq-xp)
        xr = gam**2 - xp - xq
        yr = gam*(xp-xr) - yp
    elif (yp == yq and yp != 0):
        gam = (3*(yp**2)+a)/(2*yp)
        xr = gam**2 - 2*xp
        yr = gam*(xp-xr) - yp
    return (xr, yr)

def addToSelf(xp, yp, a, num):
    #in point multiplication, we add the point to itself that many times
    (xa, ya) = addPoints(xp, yp, xp, yp, a)
    while (num-1 > 0):
        (xa, ya) = addPoints(xa, ya, xa, ya, a)
        num - 1
    return (xa, ya)
```

Figure B.2: Helper functions in python3 for implementing Elliptic Curve Diffie-Helman Key Exchange algorithm

Bibliography

J. Alwen. What is lattice-based cryptography and why should you care. *Wickr*, 2018.

H. Chung. Quantum attack on cryptography: Shor's and grover's algorithm. *National Taiwan University*, 2017.

Henri Cohen and Gerhard Frey. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Taylor Francis Group, Boca Raton, Florida, 2006.

Craig Costello. A gentle introduction to isogeny-based cryptography. *CRRao AIMSCS*, 2016.

Tony M. Damico. A brief history of cryptography. *Inquiries Journal*, 1(11):1, 2009.

Rachelle David. Isogeny based cryptography in a post-quantum cyberspace. *Unpublished*, 2019.

Jeong San Kim Dong Pyo Chi, Jeong Woon Choi and Taewan Kim. Lattice based cryptography for beginners, 2015.

CALTECH Faculty. What is quantum computing? *CalTech Science Exchange*, 2023.

Lov K. Grover. A fast quantum mechanical algorithm for database search, 1996.

J. Pipher J. Hoffstein and J. Silverman. An introduction to mathematical cryptography. *Springer*, 2010.

Y. Liu D. Moody R. Peralta D. Smith-Tone L. Chen, S. Jordan. Report on post-quantum cryptography. *National Institute of Standards and Technology Internal Report*, 2016.

T. Lange. Code-based cryptography. *Post-Quantum Cryptography Winter School*, 2016.

Yang Li, Kee Siong Ng, and Michael Purcell. A tutorial introduction to lattice-based cryptography and homomorphic encryption, 2022. URL `https://doi.org/10.48550/arXiv.2208.08125`.

Daniel Litinski. How to compute a 256-bit elliptic curve private key with only 50 million toffoli gates, 2023.

Daniele Micciancio and Oded Regev. Lattice-based cryptography. *New York University Courant Institute*, 2008. URL `https://cims.nyu.edu/~regev/papers/pqc.pdf`.

James Nechvatal James Foti Lawrence Bassham E. Roback Morris Dworkin, Elaine Barker and James Dray. Advanced encryption standard (aes), 2001-11-26 2001.

Department of Defense. Department of defense software modernization. *Memorandum for Senior Pentagon Leadership*, 2022.

P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *IEEE: Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994. doi: 10.1109/SFCS.1994.365700.

J. Silverman and J. Tate. Rational points on elliptic curves. *Undergraduate Texts in Mathematics*, 2015.

Andrew V. Sutherland. Lecture notes 2: Elliptic curves, 2015.