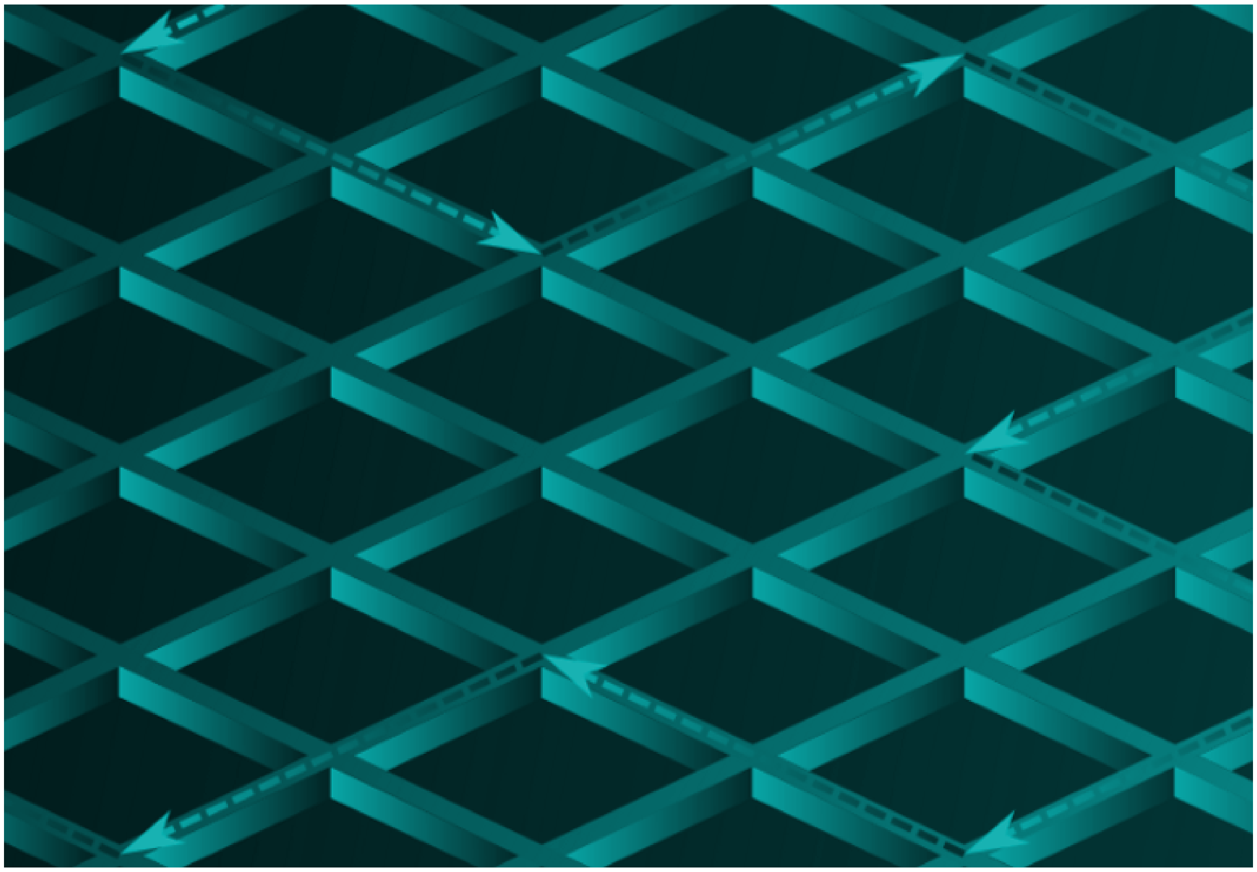


Lattice-based Cryptography

March 2022

Student: Bogdan Mezei

Student number: 293137

Course: IT-SCP1-S22

Supervisor: Richard Brooks

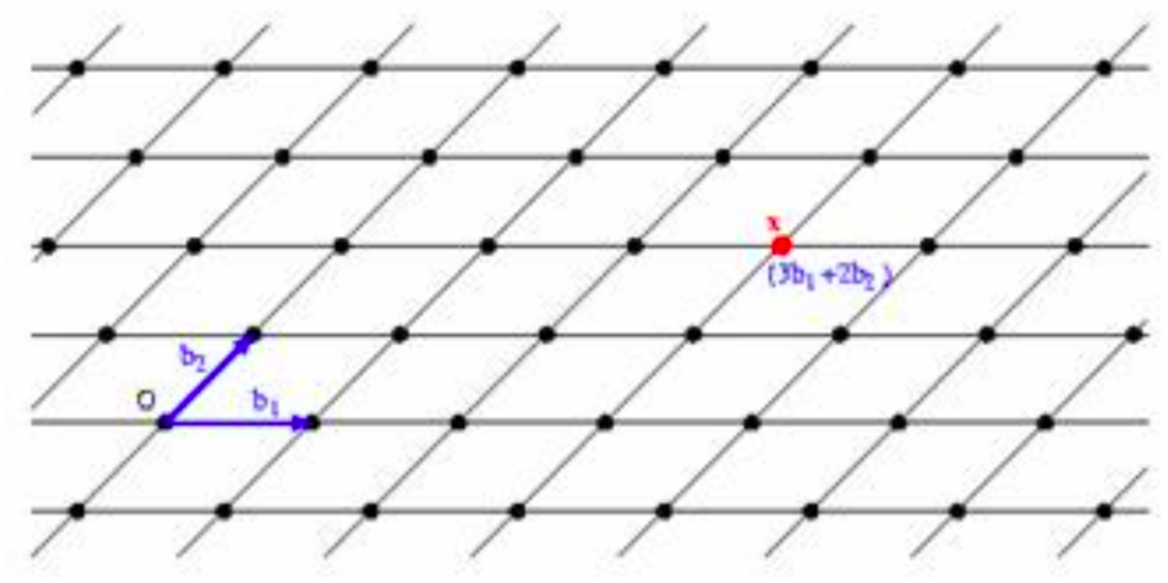
Table of contents

1. Abstract

This paper aims to explain the importance of lattice-based cryptography in a world where quantum computers have made most of our encryption techniques obsolete. The mathematical concepts behind this type of encryption/decryption will be presented as well as the reason why it is one of the most promising post-quantum cryptography contenders.

2. Introduction

The potential for quantum computing to revolutionise work in sectors such as meteorology, metallurgy, and medical research by accelerating the processing speed of difficult calculations is relatively well known. So is the threat that quantum computing may render some current methods of cryptography insecure, weakening the security that protects how we work, shop, bank, and live online. Lattice-based cryptography is one of the most promising candidates for post-quantum cryptography. It is also one of the most well-studied and understood families of hard problems in the field of mathematics.



A lattice is defined as a set of points in n-dimensional space with a periodic structure. Or equivalently, a lattice is the Z-linear span of a set of n linearly independent vectors.

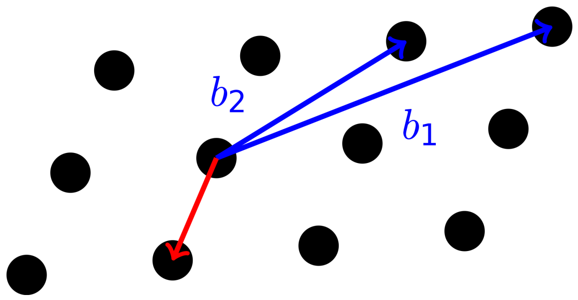
$$L = \{ a_1 v_1 + a_2 v_2 + \ldots + a_n v_n : a_1, a_2, \ldots, a_n \in Z \}$$

The vectors $v_1, v_2, \ldots, v_n$ are the basis of the lattice L. Every lattice is created from a set of basis vectors and when you multiply these basis vectors and add them together, they can form any point in the lattice. In the image above, the bases for the lattice are the vectors b_1 and b_2 and the point x is formed by computing $3b_1 + 2b_2$

3. Methods

In this section, we shall describe and analyse why lattice-based cryptography is one of the most promising candidates for post-quantum cryptography. We will take a look at the different problems which have been defined as “hard” to solve even for quantum computers as well as at some different algorithms for encryption and decryption.

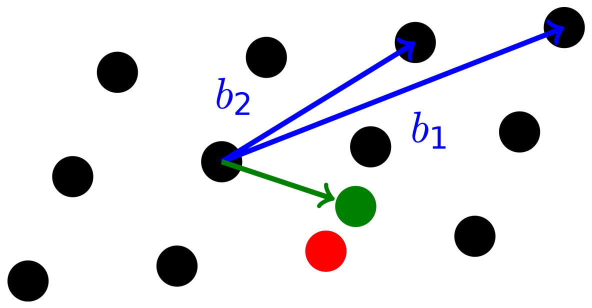
3.1 Lattice Problems



3.1.1 Shortest Vector Problem (SVP)

In the shortest vector problem one must find the shortest non-zero vector in the lattice. In the example above the basis vectors are b_1 and b_2 and the shortest vector found is marked with a red arrow. This problem is known to be NP-Hard

3.1.2 Closest Vector Problem (CVP)



In the closest vector problem, one must find the vector within the lattice which is closest to an arbitrary, external vector. In the example above the basis vectors are b_1 and b_2, the external vector is marked with the colour green and the closest vector is marked with red. The CVP is the most used problem within the proposed cryptographic algorithms for post-quantum cryptography.

3.2 Algorithms

3.2.1 Goldreich-Goldwasser-Halevi (GGH) Encryption Scheme

Mathematical Concepts

The GGH encryption scheme relies on the simplest and most intuitive solution to the CVP using Babai's Algorithm. As an example we will take a three-dimensional lattice with $v_1,\ v_2\ and\ v_3$ as the basis vectors. As such, we can create any point within the lattice by multiplying the basis vectors with a coefficient and adding them together like so:

$$p = a_1 * v_1 + a_2 * v_2 + a_3 * v_3$$

Because $v_1,\ v_2\ and\ v_3$ are all three-dimensional vectors; we can translate them into the following system of equations according to their coordinates.

$$x_p = a_1 * x_1 + a_2 * x_2 + a_3 * x_3$$

$$y_p = a_1 * y_1 + a_2 * y_2 + a_3 * y_3$$

$$z_p = a_1 * z_1 + a_2 * z_2 + a_3 * z_3$$

By solving this system of equations we can find the coefficient a. Afterwards we can calculate the closest point p to an arbitrary, external point w by solving a similar equation:

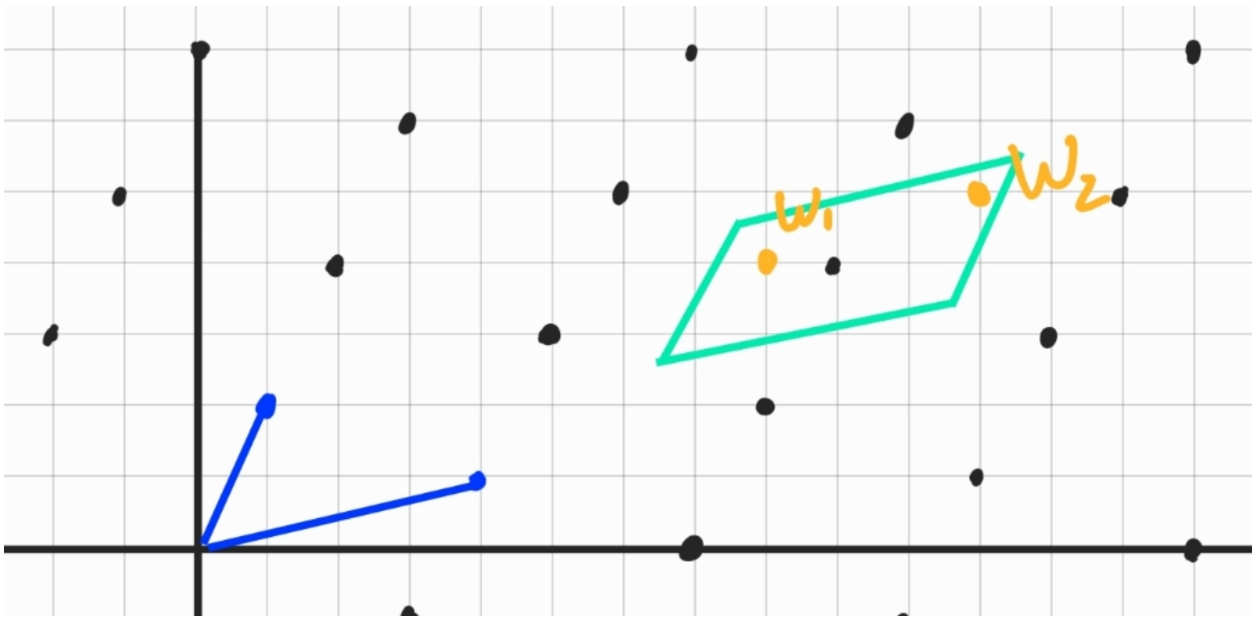
$$w = c_1 * v_1 + c_2 * v_2 + c_3 * v_3$$

Where c can take any value, so that the point is not exactly on top of an existing point within the lattice (i.e. it can be any real number). By solving this equation we can determine the closest point within the lattice p to a given point w.

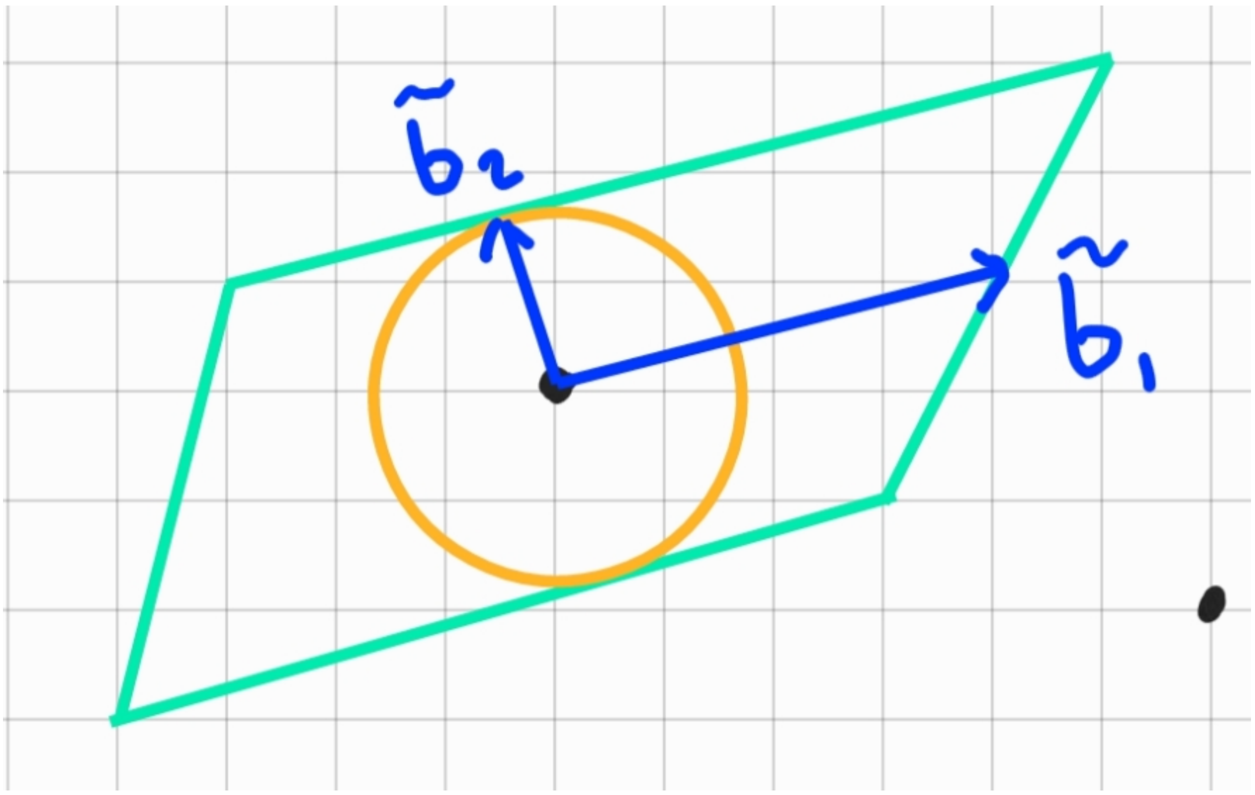
However, Babai's Algorithm works best when the bases of the lattice are relatively orthogonal. If we can find a lattice with two different bases (one relatively orthogonal and one relatively parallel) we can create an asymmetric cryptographic system where the public key encompases the parallel bases and the private key the orthogonal ones.

Babai's Algorithm Errors

Babai's algorithm does not give the correct result all the time as it depends on how close the arbitrary point is to any lattice point and on how orthogonal the bases for the lattice are. (Gylys-Colwell)



In the example above, for both $w1$ and $w2$, Babai's algorithm will show $v_1 + 2v_2$. Which is correct for $w1$ but wrong for $w2$. The area with the green boundaries shows where the result will always be $v_1 + 2v_2$. (Gylys-Colwell)



By applying the Gram-Schmidt process to our lattice bases $b_1$and b_2 the result will be the orthogonal bases $\overline{b}_1$and $\overline{b}_2$which can be used to create an area by choosing the smallest of the vectors as the radius of the circle . The area is represented by the yellow circle in the example above and represents where Babai's algorithm will always yield the correct result. (Gylys-Colwell)

The length of the smallest vector in $\overline{\{b}_1, \overline{b}_2, ..., \overline{b}_n\}$ will be taken as E and represents the error margin for the lattice. In a cryptosystem if we choose a point within the lattice and apply a noise vector smaller than E, we can be sure that the receiver (the person that has the private key) will be able to solve the closest vector problem using Babai's algorithm every single time.

Encryption Scheme Steps

1. Bob tries to send a message to Alice, but Eve is trying to intercept it.
2. Alice creates a lattice with the orthogonal bases (v_1, v_2, v_3) and keeps this as her secret private key.
3. Alice finds parallel vector bases for the same lattice (v'_1, v'_2, v'_3)
4. Alice uses the Gram-Schmidt process on her parallel bases to get $(\bar{v}'_1, \bar{v}'_2, \bar{v}'_3)$
5. Alice chooses the smallest vector from $(\bar{v}'_1, \bar{v}'_2, \bar{v}'_3)$ and sets that as E which is her error boundary.
6. Alice sends her parallel vector bases and error boundary E as her public key.
7. Bob writes his message (in secret) as numbers which will be the coefficients in the linear combination of Alice's public bases (visible to both Bob and Eve).
8. Bob uses his message as coefficients to find a point p in Alice's lattice.
9. Bob adds a small value known as noise to p in order to get p'. p must be smaller than E in order to ensure that Babai's algorithm always yields the correct result. This is done so that both Alice and Eve will have to solve the closest vector problem.
10. Bob sends p' to Alice and Eve intercepts it.
11. Alice can solve the CVP easily using Babai's Algorithm detailed above because she has the private key as the orthogonal bases for the lattice.
12. Since Babai's Algorithm doesn't work for parallel bases, Eve is unable to solve the CVP and therefore cannot find out the message.

Note: This encryption scheme is done with a very high number of dimensions so the difficulty and complexity are as high as they can be. The case described in the example with three dimensions is very simple and therefore not secure.

Security and Problems

However, the GGH encryption scheme is not secure at all because the attacker that has intercepted the ciphertext can use the Lenstra–Lenstra–Lovász (LLL) lattice basis reduction algorithm. Using LLL, the attacker can find the orthogonal basis for a lattice from the parallel basis, making the GGH encryption scheme not secure. The attacker can do this using a repetitive process that relies on a type of vector reduction called a Gram-Schmidt reduction.

3.2.2 NTRU Encryption System

Introduction

NTRU is an open-source public-key cryptography system that uses the same basic principles as GGH but with extra layers of security in order to prevent the attack using the LLL basis reduction algorithm. So far, there have not been any vulnerabilities detected in NTRU.

The main way NTRU protects against the Lenstra–Lenstra–Lovász lattice basis reduction algorithm is by using a mathematical operation known as a cyclic convolutional product.

$$f \otimes g = c_0 + c_1 x_1 + c_2 x_2 + \ldots + c_{n-1} x_{n-1}$$

$$c_k = f_0 g_k + f_1 g_{k-1} + \ldots + f_k g_0 + f_{k+1} g_{N-2} + \ldots + f_{N-1} g_{k+1}$$

The cyclic convolutional product basically multiplies every term of f with every other term in g. In a lattice-based cryptography context, these polynomials are derived from the lattices used within the encryption scheme as each point within a lattice can be described as a polynomial equation.

Encryption Scheme Steps

This cryptographic system would work as follows:

1. Alice and Bob decide on some public values defining the scope of the lattices:
   a. N - the maximum number of elements within each polynomial
   b. p and q - Usually p is relatively small and q is relatively large. They can take any value as long as GCD(p,q) = 1
2. Alice generates two sets of polynomials L_f and L_g which are of degree $N - 1$. These sets will be used in the public and private key generation.
3. Bob also generates two sets of polynomials L_m and L_θ. The former contains the message he wants to send to Alice and the latter is used to generate the noise polynomial. The noise polynomial is used in the same way as in GGH.
4. Alice chooses 2 random polynomials from L_f and L_g called f and g. The polynomial f is her private key.
5. Alice uses the Euclidean algorithm to calculate two inverses: F_p is the inverse of $f \; mod \; p$ and F_q is the inverse of $f \; mod \; q$. (If these inverses do not exist, Alice must pick new polynomials f and g from L_f and L_g)
6. Alice generates the public key as:

$$h = F_q \otimes g \; (mod \; q)$$

7. Alice sends the public key to Bob, Eve intercepts it.
8. Bob selects his message m from L_m and his noise polynomial θ from L_θ.
9. Bob uses Alice's public key h to encrypt his message m as follows:

$$e = p\,(\theta \otimes h) + m \; (mod \; q)$$

10. Alice takes the equation received from Bob and substitutes h with her own equation in order to get:

$$e = p\,(\theta \otimes F_q \otimes g\,) + m \; (mod \; q)$$

11. Alice multiplies both sides of the equation with her private key f. Since F_q is the inverse of f they cancel each other out, resulting in this equation:

$$e \otimes f = p(\theta \otimes g) + m \otimes f \ (mod\ q)$$

12. Alice takes the entire equation $mod\ p$. Since $p < q$ she can remove $mod\ q$ completely. Also $p\ (mod\ p) = 0$ so she is left with the following equation:

$$e \otimes f = m \otimes f \ (mod\ p)$$

At this point Alice has solved the closest vector problem (CVP) as the equation no longer contains the noise polynomial θ and instead collapses onto the points within the lattice.

13. Alice multiplies the entire equation with F_p which cancels out $f\ (mod\ p)$ resulting in the message m:

$$F_p[e \otimes f\ (mod\ p)] = F_p[m \otimes f\ (mod\ p)] = m$$

Security and Problems

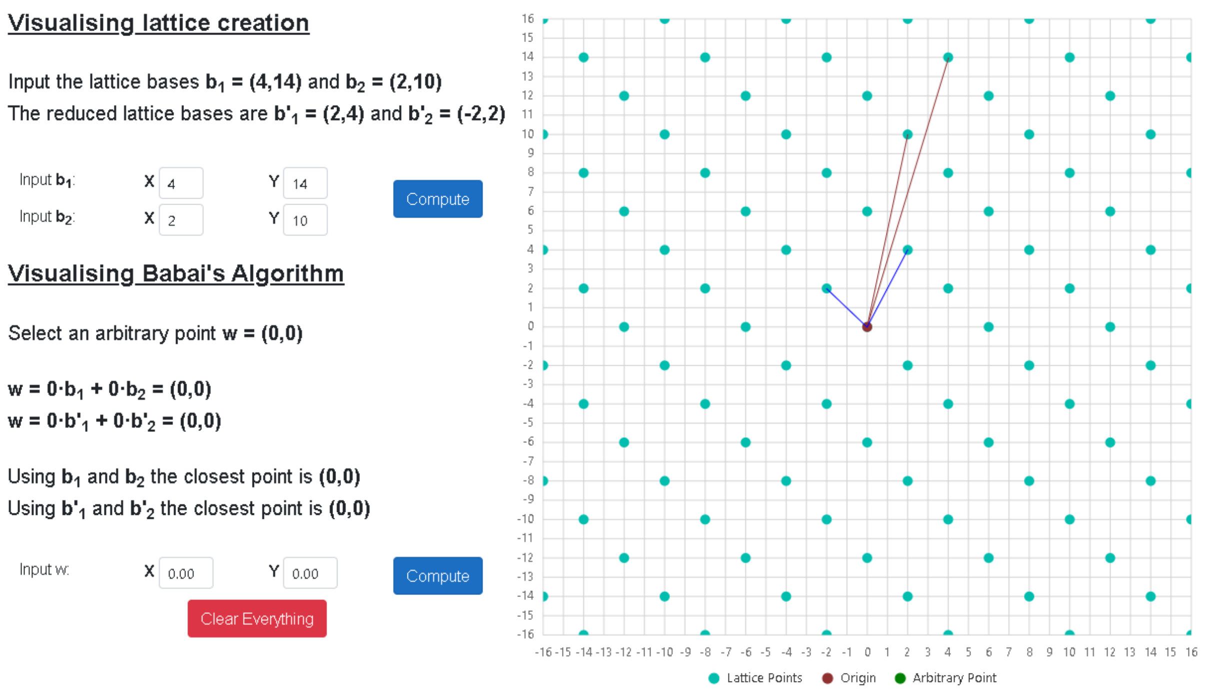
Although NTRU is known to be quantum resistant and it fixes the problems GGH has in regards to the LLL algorithm computing the orthogonal bases from the parallel ones, it still has a minor vulnerability. Because NTRU relies on the CVP similar to GGH, there is a possibility that the result from CVP is wrong and the message cannot be decrypted. However with a large enough q and small enough p values, this case is extremely unlikely so the consequences of this vulnerability are not critical.

4. Implementation

4.1 Introduction

LatticeViz is a single-user web application developed in Blazor alongside this paper. The Syncfusion library has been used for creating the complex graphs needed for this use case. The algorithms needed to compute the lattice points, reduce the bases when necessary, compute an arbitrary point and calculate the closest vector within the lattice to the said point have been implemented in C#.

The purpose of LatticeViz is to provide an easier way of understanding lattice-based cryptography. Because this type of post-quantum cryptography is so visual, it is best explained in an interactive way where the user gets to change different values and see the results instantly.



Visualising lattice creation

Input the lattice bases $\mathbf{b}_1$ = (4,14) and $\mathbf{b}_2$ = (2,10)
The reduced lattice bases are $\mathbf{b'}_1$ = (2,4) and $\mathbf{b'}_2$ = (-2,2)

Input $\mathbf{b}_1$: X 4 Y 14
Input $\mathbf{b}_2$: X 2 Y 10

Visualising Babai's Algorithm

Select an arbitrary point **w** = (0,0)

w = 0·$\mathbf{b}_1$ + 0·$\mathbf{b}_2$ = (0,0)
w = 0·$\mathbf{b'}_1$ + 0·$\mathbf{b'}_2$ = (0,0)

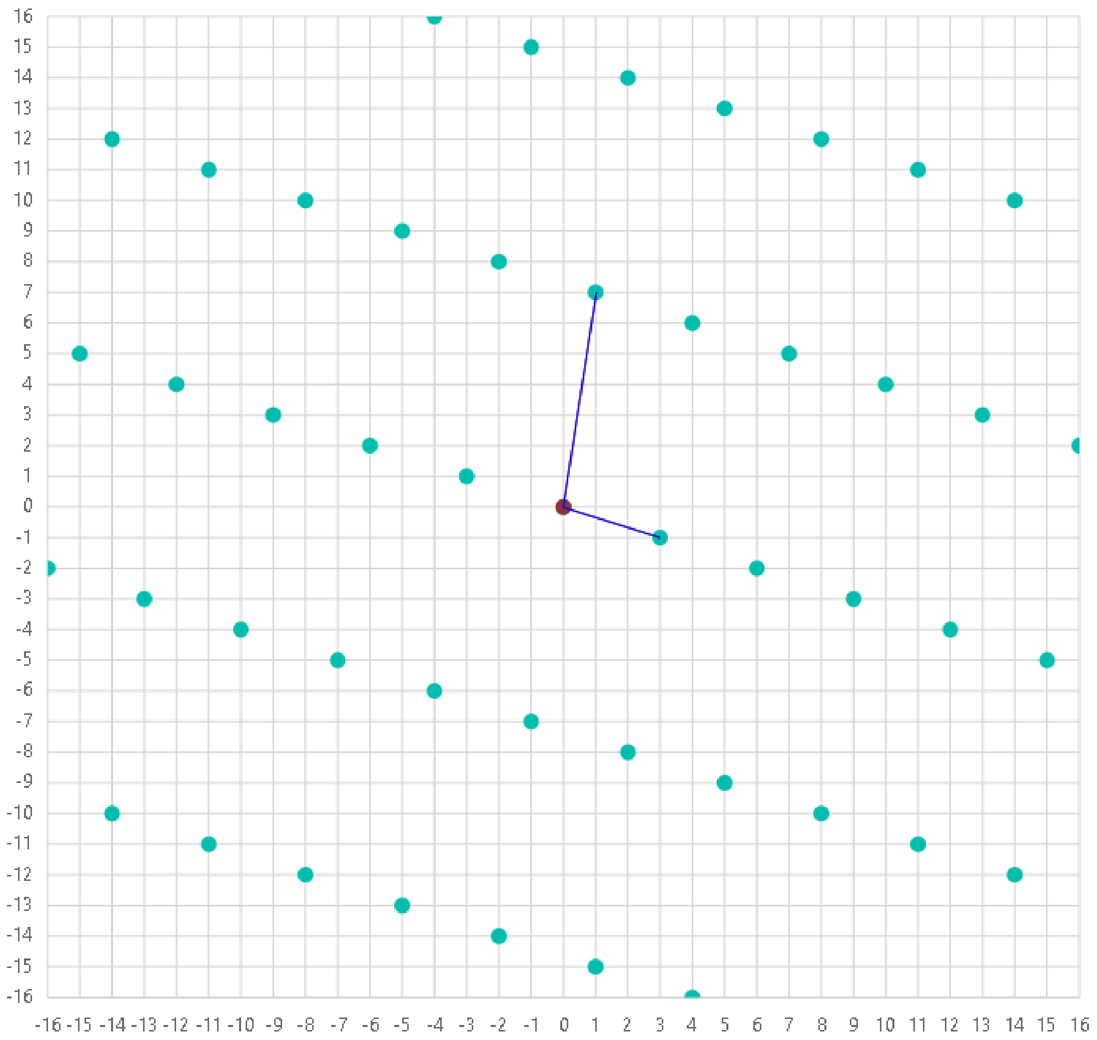
Using $\mathbf{b}_1$ and $\mathbf{b}_2$ the closest point is (0,0)
Using $\mathbf{b'}_1$ and $\mathbf{b'}_2$ the closest point is (0,0)

Input w: X 0.00 Y 0.00

4.2 Features

4.2.1 Visualising lattice creation

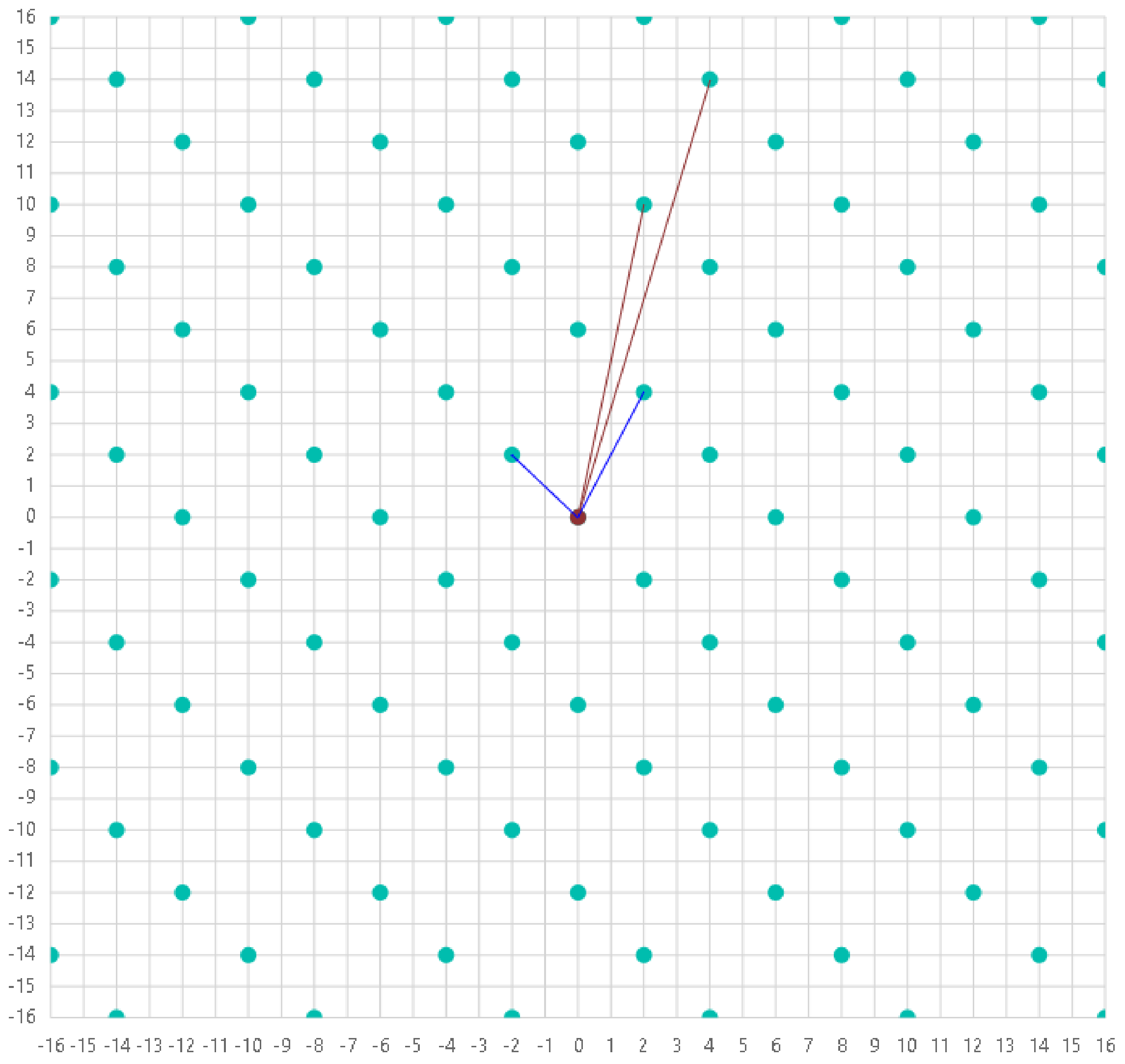
The application takes two vectors b_1 and b_1 as parameters. The user inputs these vectors as coordinates, both of them having the starting point set to the origin (0,0). In the example below, $b_1 = (3, -1)$ and $b_2 = (1, 7)$. The basis vectors are represented with blue lines and the lattice points computed from them are represented by the green dots.



4.2.2 Lattice Basis Reduction

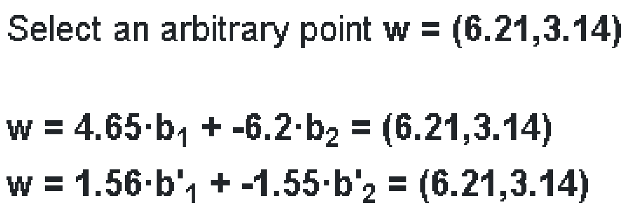
After the user inputs two vectors b_1 and b_1 as the basis for the lattice. The application will compute a reduced basis that will result in the same lattice. This is done by using the Gaussian Basis Reduction algorithm. This algorithm works only for two dimensional lattices and it works by repeatedly subtracting the smaller base from the larger one, in a similar fashion to the famous Euclidean algorithm. (Suzuki)

In the example below, $b_1 = (4, 14)$ and $b_2 = (2, 10)$ are the bases inputted by the user and being visualised with the red colored lines. The application computes the reduced bases as $b'_1 = (2, 4)$ and $b'_2 = (-2, 2)$ and visualises them with the blue colored lines.



4.2.3 Visualising Babai's Algorithm

The application is capable of using Babai's algorithm in order to solve the closest vector problem (CVP). The example below uses the same bases as before, $b_1 = (4, 14)$ and $b_2 = (2, 10)$, the reduced bases being $b'_1 = (2, 4)$ and $b'_2 = (-2, 2)$. The user inputs an arbitrary point $w = (6.21, 3.14)$ and the algorithm automatically computes two polynomials, one for the "bad" base and one for the reduced base, which describe the same point w. These polynomials are then used for Babai's Algorithm in order to solve the closest vector problem.

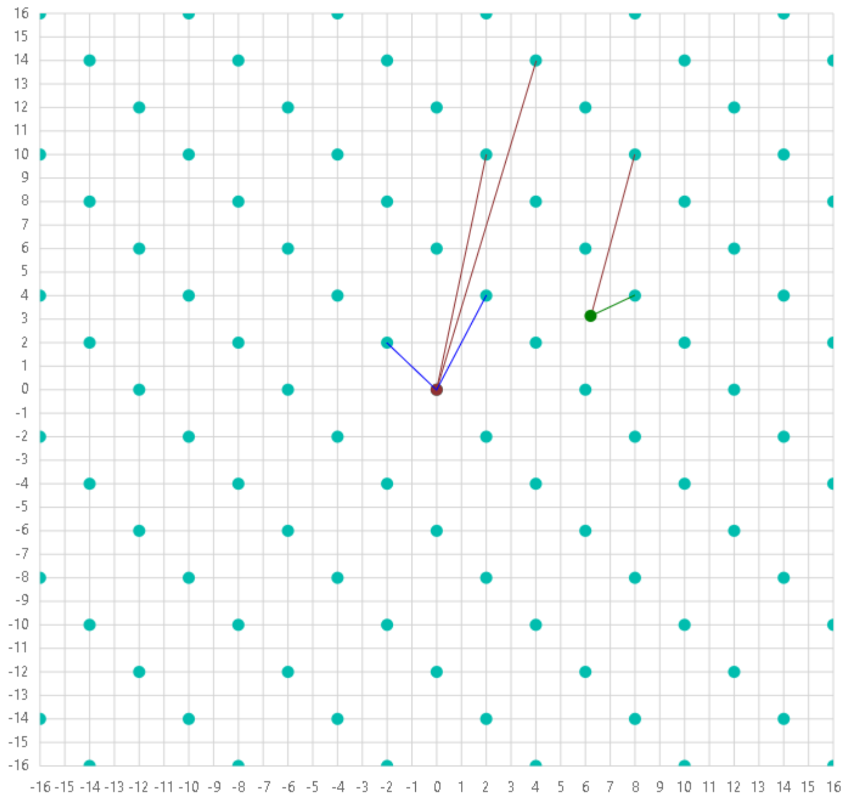


Select an arbitrary point w = (6.21,3.14)

$w = 4.65 \cdot b_1 + -6.2 \cdot b_2 = (6.21,3.14)$

$w = 1.56 \cdot b'_1 + -1.55 \cdot b'_2 = (6.21,3.14)$

Solving the CVP using the reduced base yields the correct result while using the "bad" base results in the wrong answer. This visualisation shows why when using GGH it is not reliable for someone to try solving the CVP using the "bad"/parallel bases. The correct result is visualised using a green colored line between the arbitrary point and the closest lattice point while the wrong result is shown with a red line. In some cases, Babai's algorithm will output the same, correct answer for both polynomials.



5. References

- Stoev, Vihren. “The Essence of NTRU: Key Generation, Encryption, Decryption.” Medium, 21 August 2019. Retrieved from https://medium.com/tixlcurrency/the-essence-of-ntru-key-generation-encryption-decryption-7c0540ef8441
- Gylys-Colwell, P. (n.d.). Lattice-Based Cryptography. Retrieved from https://sites.math.washington.edu/~petkusgc/blog_posts/lattice.html
- Bernstein, Daniel J. and Tanja Lange. "Post-Quantum Cryptography." 2017
- Yost, Daniel, Lattice-Based Cybersecurity In the Quantum Era: An Exploration of the Future’s Security, May 15, 2020
- Nguyen, Phong Q. (1999). "Cryptanalysis of the Goldreich–Goldwasser–Halevi Cryptosystem from Crypto '97"
- Alwen, Joël. “What is Lattice-based cryptography & why should you care.” Medium, Wickr, 15 June 2018, web. Retrieved from https://medium.com/cryptoblog/what-is-lattice-based-cryptography-why-should-you-care -dbf9957ab717
- Suzuki, Jeff, “Finding a nearly orthogonal basis”, May 2015, Retrieved from https://www.youtube.com/watch?v=uYSvb7JpySU