

Implementing Lattice Cryptography

Final Project Report

A project by Stanley Roberts (100899097)

IMPLEMENTING LATTICE CRYPTOGRAPHY	1
INTRODUCTION, MOTIVATION AND REQUIREMENTS	3
MOTIVATION.....	3
<i>Homomorphic Encryption</i>	4
PROJECT GOALS	4
REQUIREMENTS OF A QUANTUM SYSTEM.....	5
<i>Security</i>	6
<i>Efficiency/Cost</i>	6
<i>Implementation</i>	6
POST-QUANTUM CRYPTOGRAPHY MECHANISMS	7
<i>Symmetric Key modification</i>	7
<i>Multivariate Cryptography</i>	7
<i>Hash-based Cryptography</i>	7
<i>Code-based Cryptography</i>	8
<i>Supersingular Isogeny Cryptography</i>	8
<i>Lattice-based Cryptography</i>	8
CRYPTOGRAPHY FRAMEWORK	9
MATHEMATICAL THEORY AND LATTICE PROBLEM HARDNESS	10
NOTATION AND FOUNDATIONAL PROBLEMS	10
<i>Notation</i>	10
<i>Lattice Problems</i>	10
SHORT INTEGER SOLUTION (SIS).....	11
<i>Overview</i>	11
<i>Hardness</i>	11
<i>Relation to Cryptography</i>	12
LEARNING WITH ERRORS (LWE)	12
<i>Overview</i>	12
<i>Hardness</i>	12
<i>Relation to Cryptography</i>	13
<i>Amortisation</i>	13
RING-SIS	14
<i>Overview</i>	14
<i>Hardness</i>	14
RING-LWE.....	14
<i>Overview</i>	14
<i>Hardness</i>	15
<i>Relation to cryptography</i>	15
FULLY HOMOMORPHIC ENCRYPTION	15
<i>Overview</i>	15
<i>Correctness</i>	17
<i>Hardness</i>	17
<i>Relation to Cryptography</i>	17
PROBLEM CHOICE AND JUSTIFICATION FOR PROJECT	18
SOFTWARE ENGINEERING	19
PROJECT STRUCTURE	19
PROJECT FILE DETAILS	19
SOFTWARE ENGINEERING	21
<i>Testing</i>	21
<i>Methodology</i>	21
PROJECT DEVELOPMENT	23
DIARY	23
PROFESSIONAL ISSUES	26
FULLY HOMOMORPHIC ENCRYPTION EXTENSION	26
POSSIBLE CONTINUATIONS	27
FINAL REFLECTION	27
REFERENCES	28

Introduction, Motivation and Requirements

Summary:

This section outlines the motivation for quantum cryptography, as well as the requirements that must be met by a post-quantum cryptography system. Additionally we provide metrics for measuring and analysing postquantum systems and some broad options of post-quantum architectures, justifying our choice of a lattice-based system.

Motivation

Quantum computers are very quickly transforming from theory to practice with multiple successful, albeit small-scale, quantum computers having been constructed, with commercial systems available [1]. As this leap in technology is realised we must consider the very real threat it can pose on the security of many modern-day systems. Current cryptography mechanisms rely on the hardness of problems such as the Integer Factorisation Problem, however the arrival of quantum computers allows polynomial-time quantum algorithms to complete these problems [2].

Considering this, we need to develop quantum secure and efficient cryptography immediately that classical computers can use, to prevent attacks in which sensitive data is recorded now and decrypted when effective quantum computers are available. There are many different approaches to post-quantum cryptography and this section will cover most prominent approaches and analyse their viability, efficiency, and security.

Quantum computers can attack modern systems in many ways. Many of cryptography's foundational problems, for example: the Discrete Logarithm Problem (DLP), Integer Factorisation Problem (IFP) and the elliptic curve variant of DLP (ECDLP), are the base of many widely used cryptographic systems. Algorithms and techniques like Shor's algorithm [3] and quantum annealing [4] can compute answers to these in polynomial time. Some cryptographic primitives like hashes are also vulnerable to Grover's algorithm [5] and its variants.

If we consider the problems mentioned above (IFP, DLP, ECDLP), Shor's algorithm can compute their solution in polynomial time. Many cryptosystems are based around these problems' intractability, for example Diffie-Hellman relies on DLP and RSA relies on the IFP [6]. The realisation of Shor's algorithm on a sufficiently advanced quantum computer can allow the breaking of these (and any other system which rely on these problems) within an achievable time frame [7]. There are other quantum algorithms that may provide even more optimisations over Shor's, for example posing the problem as an optimisation problem and then using quantum annealing can also solve these once intractable problems [4].

Quantum algorithms can also attack hashes. Grover's algorithm allows function inversion with time complexity $O(\sqrt{n})$ whereas classical algorithms have a $O(n)$ limit (as the entire domain must be searched). This speedup allows faster brute forcing of secret symmetric keys. Grover's algorithm can also be applied to find collisions in hash functions or compute a preimage for a hash function [7]. This obviously poses a threat to the individual hashes and any system relying on a hash's security, for example HMAC.

Considering the threats posed above, it is obvious that significant changes to our security infrastructure are required (at least) before quantum computers arrive. Additionally, quantum

computers may well be on the horizon, billions are being poured into quantum computing related projects and some forecasts predict useable quantum computers within just 10 years [8].

This said, there are still several difficulties in building a quantum computer; obvious barriers include the cost and difficulty of obtaining suitable materials [9]. Quantum computers also require: scalability, qubits that are easily readable and 'initialisable', a gate set and, quantum gates faster than decoherence [10]. This last requirement is especially difficult; the quantum system needs to be entirely isolated from its external environment as well as controlling decoherence produced by parts of the quantum computer itself to ensure correct readings. Some academics even dispute that, due to this, a viable quantum computer could ever exist [11]. Additionally some quantum algorithms require a large running time before they begin to show an advantage over classical algorithms, most notably algorithms with only quadratic speedups. Cubic and quartic speedups, however, very quickly display their advantage [12].

So even if these problems remain unsolved (in a feasible and scalable way), why must we start researching post-quantum algorithms now? Firstly, we must consider that all the previously mentioned problems may be solvable; we must be prepared for the eventuality where quantum computers are realised. Next consider that sufficiently secure algorithms may take time to develop; these systems need to be implemented before viable quantum computers are available which can take considerable time, even in a crypto-agile system.

Finally, for sensitive information, there exists the threat of a 'harvest now, decrypt later' attack, in which valuable data is recorded now and decrypted when quantum computers arrive. Some data is critical even over a large amount of time, for example passwords or Personally Identifiable Information. A 10 billion USD attack on bitcoin is even possible using the 'harvest now, decrypt later' strategy [13]. These multiple considerations prove we need to develop and implement efficient post-quantum secure cryptography immediately.

Homomorphic Encryption

In this project we also develop a Levelled Homomorphic Encryption system, a concept first proposed by Gentry [14]. Fully homomorphic encryption (FHE) is a cryptographic system in which arbitrary computations can be performed on encrypted data without leaking information about that data. This concept may prove to be incredibly useful in the realm of cloud computing as it allows large computations on sensitive data to be offloaded to an untrusted party. Previously, this would not be possible without the third party decrypting the data to perform computation upon it. Another valuable use of FHE is remotely storing encrypted data in the cloud. Encrypted data may be stored, and the user will retain the ability to perform queries and manipulations on this data without the third party determining anything about the stored data or the operation performed on it. With the growing constraints on data protection, FHE may provide a solution for organisations to retain the ability to outsource computation and storage while retaining sufficient security on the sensitive data.

Project Goals

With the background behind quantum secure cryptography and other relevant mechanisms established we will now lay out the project purpose, goals, and deliverables.

The primary goal is to develop a system that can provide a quantum secure cryptography mechanism allowing encryption and decryption with public keys. This is built upon to produce a quantum secure Public Key Encryption system. We also provide the relevant theory required to prove the correctness and security for the schemes we use and provide some useful tools that utilise

the encryption system. Finally we provide a report on the relevance of this project to modern cryptography, and the rationale and development behind the project.

The following are the deliverables we provide to achieve these goals, which have been compiled to produce this document and the project code:

Title: Motivation for Quantum Cryptography Report

Description: A report detailing the necessity to develop post-quantum security, including the urgency of this (harvest now, decrypt later attacks). Provides reasoning as to why lattice-based solutions are a valid candidate for post-quantum cryptography and establish requirements and standards such a scheme must meet.

Motivation: Establishes the necessity and reasoning behind this project and lays the foundation of developing a lattice scheme.

Title: Lattice Structures and Problems Report

Description: A cursory description of relevant lattice theory and then an in-depth explanation of various lattice problems (SVP, BDD, LWE/SIS variants, etcetera) and how these relate to, and can be used in, post-quantum cryptography. We also analyse the hardness of each lattice problem we review, and how each lattice construction relates and reduces to these problems.

Motivation: Provides the necessary information to effectively understand and then choose a lattice problem, on which to base the post-quantum scheme. This report also provides a foundation for the minimum expected efficacy and security of each potential lattice algorithm.

Title: Lattice Scheme Implementation Program

Description: This initial program will implement Regev's [15] lattice scheme based on Learning with Errors, to implement a post-quantum secure PKE mechanism. This program will be written using SageMath. We additionally implement an optional amortisation [16] improvement for multiple bit encryption.

Motivation: This program provides the first post-quantum encryption scheme for our project, which implements public key cryptography.

Title: Ring-LWE and Fully Homomorphic Encryption System

Description: This implements the faster LWE variant, Ring-LWE and builds a PKE system using it. It also implements a fully homomorphic scheme for this system, supporting multiplication and addition on polynomial ciphertexts.

Motivation: This program provides an additional lattice scheme, Ring-LWE, and introduces more functionality for the cryptosystem allowing Fully Homomorphic encryption.

This project compiles various lattice algorithms and provides a practical cryptographic implementing them. We provide mechanism for interacting with and customising these algorithms, providing a concrete and practical implementation of previously theoretical algorithms. Compared to similar libraries such as Microsoft SEAL [17] we provide multiple different encryption schemes of various security and efficiency. SEAL only provides FHE algorithms, which intrinsically have a lower security than standard PKE algorithms and as such is not always suitable. We also aim to keep the core code close to the algorithms outlined in the literature, such that a casual reader can easily identify and understand how the algorithms are implemented.

Requirements of a Quantum System

We have established the necessity of developing post-quantum security imminently, so we must now deduce what makes a system quantum-secure and how can we develop an algorithm that effectively provides this? Obviously our algorithm must be capable of efficiently running on a

classical computer, given we need to prevent ‘harvest now, decrypt later’ attacks but we also need to assure quantum security. This is troublesome, given how difficult it is to predict quantum capabilities (cost, speed, and memory) and due to infancy of this field, the many possible quantum algorithms that have yet to be developed.

A couple of simple preliminary requirements we can establish for developing a quantum system in a security context is that the algorithms should not have prominent components that rely on or incorporate quantum insecure problems (such as IFP or DLP) and the system should be able to provide at least one of: Public Key Encryption (PKE), Key Encapsulation Mechanism (KEM) or Digital Signature primitives.

To compare and analyse a potential post-quantum candidate we must establish more rigorous criteria. The following is based on the specifications laid out by NIST [18] and ETSI [19]. These requirements can broadly be categorised into the following considerations: security, efficiency and cost, and implementation.

Security

Firstly let’s analyse security requirements. A post-quantum viable system must be secure against classical computers. This security is much easier to quantify as classical computing speeds, algorithms and techniques are generally well understood. Measuring the strength against quantum computers is much harder, especially analysing the ‘bits’ of security in the same way we do with classical algorithms.

NIST instead proposes ranking algorithms based on the computational resources (quantum circuit size) required to break it compared to other well-known primitives (e.g. collision search on a certain bit hash, or key search for a certain key size block cipher). We also consider security reductions to known hard problems and additional desirable security features, such as forward secrecy¹ and side-channel attack² resistance. These evaluation criteria should provide an acceptable way to measure an algorithms security, even in the unknown quantum landscape.

Efficiency/Cost

Here we consider the general cost of using the algorithm. We first examine space costs, contributed to by large key sizes, ciphertexts and signatures (comparatively to other schemes offering the same security level). We must evaluate computational efficiency, both speed and resources. The speed the scheme can encrypt, or sign data, is considered in addition to the number of round trips required to provide these services. This efficiency should be measured across devices with varying computational power to ensure a scheme that is resource friendly, as well as fast. Memory requirements must also be recorded as part of this efficiency analysis.

Finally, we consider negatives such as a signature limit or the rate of decryption failures (if failures are possible). In total this allows us to establish the cost of using a given scheme and further quantify its viability.

Implementation

This is a vague category but refers to the administrative adoption of a scheme in a general system. We consider the simplicity of the scheme overall as well as its ease of implementation and operation by non-experts. We may again measure space and memory costs to evaluate its deployment capabilities on limited devices (e.g. embedded devices). We also recognise its overall flexibility, in

¹ The notion that past encryptions remain secure even if long-term keys are broken in the future

² An attack exploiting the implementation of a system, rather than the underlying algorithm. For example, an attack that uses information gained by analysing the time a server takes to compute various requests.

terms of: customisation, parallelisation, implementation versatility (in existing protocols, applications and devices), and additional functionality (beyond basic PKE, KEM and signing). We finally examine the difficulty of implementing a scheme into the prominent existing protocols and schemes, with minimal changes.

These three categories allow us to quantify how useful a post-quantum scheme is overall, and thus helps decide on a viable framework for building our scheme. Further it allows comparison to other existing schemes and lets users evaluate how compatible a given scheme is (e.g. a risk-averse company may prefer sacrificing efficient/cost requirements in favour of higher security).

Post-Quantum Cryptography Mechanisms

Now we have established the evaluation criteria for our quantum scheme we must choose the algorithmic framework best suited for a general-use post-quantum secure cryptography system. There are many possible considerations however we will use the most promising and well-researched algorithms: Symmetric Key modification, Multivariate, Hash-based, Code-based, Isogeny-based, and Lattice-based cryptography.

Symmetric Key modification

Most existing symmetric key algorithms are inherently resistant to quantum algorithms. Its only quantum insecurity is vulnerability to a brute force (optimised using Grover's algorithm) of the symmetric key, but this can be mitigated through a sufficiently increased key size [20].

Although implementation for such a system would be trivial due to the widespread use of symmetric key cryptography and the only modification being a key size change, it has some flaws that make it unsuitable for our purposes. The fact that the method of key agreement must be quantum secure, and that symmetric key cryptography can't provide PKE or KEM means that, in using this scheme, we'd have to rely on another quantum secure framework anyway.

Multivariate Cryptography

Multivariate cryptography attempts to use the foundational problem of solving multivariate equations to implement a quantum secure cryptography system. Unfortunately few viable quantum encryption schemes have been developed, so it is practically limited to digital signatures. However, among quantum algorithms, it constructs signatures very efficiently. Multivariate schemes can typically be reduced to solving multivariate polynomials, an NP-complete problem. The inefficiency of encryption (large keys and slow decryption) makes using this framework makes this a less desirable choice than the alternatives [21].

Hash-based Cryptography

Cryptographic hash functions have been used to implement digital signatures. Their main downside is the limited number of signatures the system can produce, leading to minimal use. However, some such schemes may be resistant to quantum attacks. The chosen cryptographic hash-function must be quantum secure, meaning a security reduction to a quantum solvable problem does not exist.

Hash function signatures use a Merkle tree to allow multiple signatures however this introduces tree traversal as a time complexity factor, which can only be mitigated through increased signature size. Even with Merkle trees, the total amount of signatures is still finite [22].

Additionally, implementing PKE using hashes is inefficient and fault prone. It requires relatively large computational and network resources and has a possibility of failure [23]. Due to the numerous flaws and the desirability of flexibility (we wish to implement multiple primitives, not just signing), hash-based cryptography is not our most suitable candidate.

Code-based Cryptography

Code-based systems are based on error correct codes and the hardness (NP-complete) of decoding a linear code. While not commonly used, it is quantum resistant as the introduction of randomness into the algorithm provides immunity to Shor's algorithm speedups [24]. The system uses codes with efficient decoding algorithms to produce public and private key pairs. The encryption algorithm introduces randomness, and the error correcting codes allow its removal during decryption.

This method of cryptography requires a good choice of code family else it may be vulnerable to a structural attack revealing the decoding algorithm. Additionally algorithms of this type have large memory requirements, and quantum security requires key sizes of over 8 million bits [25]. Given these factors, and the current minimal use of code-based cryptography means it is not the best fit for our framework.

Supersingular Isogeny Cryptography

Isogeny cryptography is a modification of (Elliptic Curve) Diffie-Hellman, replacing the secrecy mechanism from relying on the DLP to instead using elliptic isogenies. The algorithm has each party generate an isogeny using linear combinations of public elliptic points. These isogenies are used to compute new values for the original points, which are then transmitted. Finally the transmitted values are used with the linear combination of points to create a new isogeny (from a public supersingular curve), these two curves are isomorphic and have the same j -invariant³ which is used as the secret key.

The security of isogeny cryptography is equivalent to finding the isogeny mapping between two supersingular elliptic curves with equal number of points. The best method for this so far is solving the claw finding problem, whose optimal algorithm is improved by quantum computing but not significantly, leaving it a quantum secure algorithm [28]. Isogeny cryptography allows forward secrecy which makes it a good candidate. It also utilises low memory and small key sizes, however, suffers significantly in speed: hundreds of times slower than other quantum alternatives. Isogeny is also a poor choice for signatures, producing very large signatures: again hundreds of times slower than alternatives [29]. These inefficiencies lead us to look elsewhere.

Lattice-based Cryptography

Finally we have lattice cryptography, the most promising candidate for a quantum framework. Lattice cryptography has a strong mathematical foundation, relying on lattice problems like Short Integer Solution (SIS) and LWE (Learning with Errors) and ring-LWE which have all been proven to be reducible to worst-case lattice problems [30] (e.g. the Shortest Vector Problem, an NP-hard problem).

The implementation of lattice cryptography depends on the problem used, among other factors, but generally lattice cryptography is efficient and secure. It has also been used to develop PKE, KEM and digital signatures successfully [31, 32]. Furthermore, its hardness is based on well-studied mathematics giving us very high confidence in its quantum security. Additionally, key exchange using Ring-LWE can provide forward secrecy, a very desirable property. The only downside is it requires a large public key size, however considering its small signature size, fast speed and low resource requirements, lattice problems seem to provide a brilliant framework for post-quantum algorithms.

³ The j -invariant is a parameterisation of the elliptic curve over \mathbb{C} and defines isomorphism classes of the elliptic curves, i.e. curves with the same j -invariant are isomorphic. See [26, 27] for details.

Cryptography Framework

Given a quantum scheme, we have established its minimum requirements as well as evaluation criteria based on its security, efficiency, cost, and implementation. Given these metrics we have analysed the current leading post-quantum cryptographic frameworks, allowing us to judge different systems based on their merit and viability thus allowing us to choose a suitable framework. Given the candidates analysed, I settled on lattice cryptography given its very small drawbacks compared to other system as well as its strong mathematical basis and effectiveness with regards to security and efficiency.

Mathematical Theory and Lattice Problem Hardness

Summary:

This section is intended to provide a theoretical background for lattice problems, and their related hardness, such that these problems can be applied to a cryptographic system. It provides a foundation for choosing and justifying the problem that a lattice scheme relies on and will determine the lattice problem that this project's final cryptographic mechanism will use. For each relevant problem we first review its definition, and then evaluate its hardness assumptions and deductions and consequentially its application and practicality in developing a (quantum-secure) cryptographic mechanism.

This can be considered a literature review on the most prominent cryptography-related lattice problems alongside evaluation of each problems relevance to this project, concluding with a decision and justification for the chosen problem

Notation and Foundational Problems

Notation

As standard, \mathbb{F}^n represent an n -dimensional vector space on \mathbb{F} . Bold lowercase letters denote elements of the appropriate ring and bold uppercase letters denote matrices. An augmented matrix $(\mathbf{A}|\mathbf{B})$ is the horizontal concatenation of two matrices \mathbf{A} and \mathbf{B} . Additionally the norm $\|\cdot\|$ refers to the Euclidean norm unless otherwise specified.

A lattice \mathcal{L} is a subset of \mathbb{R}^n that is an additive subgroup (i.e. $\mathbf{0} \in \mathcal{L}$ and for every point $x, y \in \mathcal{L}$ then $-x, x + y \in \mathcal{L}$) and discrete (i.e. every lattice point $x \in \mathcal{L}$ has a neighborhood in \mathbb{R}^n such that x is the only lattice point).

A basis B for some lattice \mathcal{L} is a set of linearly independent vectors (basis vectors) that can define \mathcal{L} under linear combinations. Formally a basis B , composed of k basis vectors \mathbf{b} is a basis of lattice \mathcal{L} iff:

$$\mathcal{L} = \mathcal{L}(B) := \left\{ \sum_{i=1}^k z_i \mathbf{b}_i : z_i \in \mathbb{Z} \right\}$$

In this review, we define $\lfloor x \rfloor$ for some real x as the 'floor function', the largest integer smaller or equal to x . For some ring R we denote the quotient ring R/nR as R_n (e.g. \mathbb{Z}_5 is the ring of integers *mod* 5). We use $e \leftarrow \chi$ to denote sampling e according to some distribution χ . Finally, considering a lattice \mathcal{L} , then $\lambda_i(\mathcal{L})$ is defined as the smallest non-zero value v such that \mathcal{L} contains i linearly independent vectors whose norm is less than v (e.g. $\lambda_1(\mathcal{L})$ is equal to the norm of the smallest vector in \mathcal{L}) and $\text{dist}(p, \mathcal{L})$ is the distance from an arbitrary point p to the lattice \mathcal{L} .

Lattice Problems

It should be noted the following problems are all approximation problems, considering an approximation factor $\gamma \geq 1$.

SVP: Given basis B of an n -dimensional lattice $\mathcal{L} = \mathcal{L}(B)$, find a non-zero vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| \leq \gamma \cdot \lambda_1(\mathcal{L})$

GapSVP: Given basis B of an n -dimensional lattice $\mathcal{L} = \mathcal{L}(B)$ determine if $\lambda_1(\mathcal{L}) \leq 1$ or $\lambda_1(\mathcal{L}) > \gamma$, given that exactly one of these equalities is true.

SIVP: Given basis B of an n -dimensional full-rank lattice $\mathcal{L} = \mathcal{L}(B)$ produce n linearly independent vectors v such that $\|v_i\| \leq \gamma \cdot \lambda_n(\mathcal{L})$ for all i .

BDD: Given basis B of an n -dimensional lattice $\mathcal{L} = \mathcal{L}(B)$ and a target $t \in \mathbb{R}^n$ such that $\text{dist}(t, \mathcal{L}) < \lambda_1(\mathcal{L})/\gamma$ find vector $v \in \mathcal{L}$ closest to point t .

Short Integer Solution (SIS)

Overview

The short integer solution problem, introduced by Ajtai [33], is closely related to solving a system of linear equations, the main difference being the extra constraints requiring the solution be ‘short’ and an integer vector. An informal definition of SIS is as follows:

Given q and n are some positive integers, recall that \mathbb{Z}_q is the ring of integers modulo q and \mathbb{Z}_q^n is an n -dimensional vector space over \mathbb{Z}_q . We select m uniformly random vectors $a_i \in \mathbb{Z}_q^n$ which each make up the columns of a matrix $A \in \mathbb{Z}_q^{n \times m}$. SIS tasks us to find a non-zero vector $z \in \mathbb{Z}^m$ such that:

$$f_A(z) := Az = \mathbf{0} \in \mathbb{Z}^n$$

adhering to the constraints $\|z\| \leq \beta$ and $\beta < q$ where β is some small real.

We require that the norm of z be constrained to at most β as else we can solve the problem by applying Gaussian elimination to $(A|\mathbf{0})$ which reduces to the matrix $(I|z)$, a standard method for solving a system of linear equations. We also require $\beta < q$ else there are m trivial solutions as since $q = 0 \bmod q$ we can derive a solution $z = qu$ where u is some standard unit vector, which is a valid solution with $\|z\| \leq \beta$.

Note that increasing m does not increase the difficulty of the problem: a solution z for matrix A can be extended to $(A|B) \forall B \in \mathbb{Z}^{n \times k}$ by appending k zeroes to z . Conversely, increasing n requires an entire recalculation of z with more computations, thus increasing n also increases the hardness of the problem.

When deciding the values of β and m naturally we desire to make them small, but they must be large enough to guarantee a solution.

This occurs when $\beta \geq \sqrt{\bar{m}}$ and $m \geq \bar{m}$ where $\bar{m} := \lceil n \log q + 1 \rceil$.

Proof: WLOG assume $m = \bar{m}$. Then consider the set $X := \{0, 1\}^m$ whose cardinality is $|X| = 2^m$. By transitivity we can see $2^m > 2^{n \log q}$ where $2^{n \log q} = q^n$ is the number of vectors in \mathbb{Z}_q^n . Therefore, using the pigeonhole principle we can deduce that for any matrix A there must be two distinct x, x' in X such that $Ax = Ax' \in \mathbb{Z}_q^n$. These vectors difference $v = x - x'$ is of course a valid solution and belongs to the set $\{0, \pm 1\}^m$ thus its maximum norm is at most $\sqrt{\bar{m}}$, hence, in the general case, we require that $\beta \geq \sqrt{\bar{m}}$ to ensure this answer exists, and, of course, from our original assumption we require that $m \geq \bar{m}$.

Hardness

We can immediately see that SIS corresponds to an instance of SVP (average-case) upon a certain lattice family:

$$\mathcal{L}(A) := \{z \in \mathbb{Z}^m : Az = \mathbf{0} \in \mathbb{Z}_q^n\}$$

Where the aim is to find a sufficiently small non-zero vector z in a given lattice for some uniformly random A , quite clearly analogous to the SVP problem.

We can improve on this with a reduction, proving that solving an average instance of SIS is, with very large probability, at least as hard as solving a worst-case of GapSVP or SIVP (for sufficient parameters). The security reduction is proved by Ajtai [33] however numerous improvements on his parameter bounds (that still provide this security reduction) have been made. [34-36]

Relation to Cryptography

As we can see, assuming the SIS problem is hard, then finding a collision $\mathbf{Ax} = \mathbf{Ax}' \in \mathbb{Z}_q^n$ for some \mathbf{A} is also hard, as finding such a collision would solve the SIS problem. This ‘collision resistance’ property is obviously very useful for creating a cryptographic hash function. SIS problems can also provide functions which are easy to compute but hard to invert, called trapdoors. Such trapdoors can lay foundations for other cryptographic primitives such as quantum secure digital signatures and identity-based encryption systems [34].

Learning with Errors (LWE)

Overview

This system is very similar to SIS, however with a few differences allowing its use in a public encryption system [32]. As in SIS, \mathbb{Z}_q^n is the n -dimensional vector space of the ring of integer modulo q , where n and q are positive integers. We also define a probability distribution χ over \mathbb{Z} that generates suitably short vectors with high probability (typically a discrete gaussian whose width controls the shortness)

We choose a secret $\mathbf{s} \in \mathbb{Z}_q^n$, and define an LWE distribution $A_{\mathbf{s},\chi}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q$ which outputs a sample (\mathbf{a}, b) , where $\mathbf{a} \in \mathbb{Z}_q^n$ is chosen uniformly at random and $b = \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q$ where $e \leftarrow \chi$. We define two problems based on this:

Search-LWE: Given m samples (\mathbf{a}_i, b_i) from $A_{\mathbf{s},\chi}$ for some fixed \mathbf{s} , find \mathbf{s}

Decision-LWE: Given m samples (\mathbf{a}_i, b_i) where each sample is either generated from $A_{\mathbf{s},\chi}$ with a fixed \mathbf{s} , or from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$, identify whether each sample was generated from $A_{\mathbf{s},\chi}$ with a non-negligible advantage.

Similarly to SIS, we can combine the sample into a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ whose columns are the vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$. This allows us to rephrase the set of samples as:

$$\mathbf{b}^t = \mathbf{s}^t \mathbf{A} + \mathbf{e}^t \pmod{q}$$

where $\mathbf{e} \leftarrow \chi^m$, thus we can see that the entry i in the vector $\mathbf{b} \in \mathbb{Z}_q^m$ corresponds to sample part $b_i \in \mathbb{Z}_q$

Hardness

This matrix definition allows us to rephrase the problem as a lattice problem on a lattice family:

$$\mathcal{L}(\mathbf{A}) := \{\mathbf{A}^t \mathbf{s} : \mathbf{s} \in \mathbb{Z}_q^n\} + q\mathbb{Z}^m$$

Where the goal is to find \mathbf{s} given \mathbf{b} , which is an instance of BDD (average-case) as \mathbf{s} is relatively close to \mathbf{b} are as defined by the error \mathbf{e} .

Similarly to SIS, we can again improve on this hardness with a reduction. Regev [32] proved that for sufficient parameters, solving decision-LWE in the average case is at least as hard as solving a worst-case instance of GapSVP or SIVP.

Relation to Cryptography

The main benefit of LWE, is it's the 'encryption-enabling version' of SIS. LWE provides a suitable problem foundation for building encryption, including public-key cryptography. Using \mathbf{s} as the secret key and samples of the LWE function as a public key \mathbf{A} . This allows a quantum secure architecture with (relatively) small key sizes ($\tilde{O}(n^2)$ public keys and $\tilde{O}(n)$ secret keys and ciphertext).

This public key cryptography mechanism was first detailed by Regev [15] with private key equalling the secret \mathbf{s} and public key equalling the m samples generated from $A_{\mathbf{s},\chi}$ and encryption/decryption are defined as:

Encryption: for a uniformly random subset S of the public key, the encryption of a binary bit p is $(\sum_{i \in S} \mathbf{a}_i, p \cdot \lfloor \frac{q}{2} \rfloor + \sum_{i \in S} b_i)$

Decryption: for a pair (\mathbf{a}, b) the decryption is 0 if $b - \langle \mathbf{a}, \mathbf{s} \rangle$ is closer to 0 than $\lfloor \frac{q}{2} \rfloor \bmod q$ and is 1 otherwise.

Correctness: for encryption of bit $p = 0$ producing the pair (\mathbf{a}, b) then $b = p \cdot \lfloor \frac{q}{2} \rfloor + \sum_{i \in S} b_i = p \cdot \lfloor \frac{q}{2} \rfloor + \sum_{i \in S} \langle \mathbf{a}_i, \mathbf{s}_i \rangle + e_i = p \cdot \lfloor \frac{q}{2} \rfloor + \langle \mathbf{a}, \mathbf{s} \rangle + \sum_{i \in S} e_i$. Decrypting this b as $b - \langle \mathbf{a}, \mathbf{s} \rangle$, produces the plaintext $p \cdot \lfloor \frac{q}{2} \rfloor + \sum_{i \in S} e_i$, and assuming $\sum_{i \in S} e_i$ is small (which it is with high probability as defined by the error distribution), then the decryption $p \cdot \lfloor \frac{q}{2} \rfloor$ is closer to 0 only when $p = 0$, and is closer to $\lfloor \frac{q}{2} \rfloor$ when $p = 1$.

Security: We assert that a uniformly random public key \mathbf{A} is indistinguishable from a correctly generated public key. This follows from the fact that to distinguish the uniformly generated samples the attacker must use an algorithm D which takes samples from \mathbf{A} and ciphertext pairs (\mathbf{a}, b) and outputs whether each input was generated from $A_{\mathbf{s},\chi}$ or is uniformly distributed. D is clearly a Decision-LWE solving algorithm, as samples from \mathbf{A} are uniformly distributed and ciphertext pairs (\mathbf{a}, b) are samples generated from $A_{\mathbf{s},\chi}$. Thus, distinguishing a uniformly random public key from a correct public key, is hard if Decision-LWE is hard.

Additionally by the leftover hash lemma [37] a pair $(\mathbf{A}, \mathbf{u} = \mathbf{A} \cdot \mathbf{x})$ for a uniform $\mathbf{A} \leftarrow \mathbb{Z}_q^{(n+1) \times m}$ and $\mathbf{x} \in \{0,1\}^m$ is indistinguishable from a uniformly random sample. The pair (\mathbf{A}, \mathbf{u}) is analogous to ciphertext pairs (as adding a vector $(\mathbf{0}, p \cdot \lfloor \frac{q}{2} \rfloor)$ to \mathbf{u} preserves its uniformity) and thus when the ciphertext is also uniformly random it is indistinguishable from the above problem, in which solely the public key is uniformly random.

Amortisation

An optimisation on Regev's LWE cryptosystem is amortisation [16] which allows multiple bits to be encrypted and decrypted simultaneously. This is achieved through 'reusing' the subsets of the public key such that we can encrypt and decrypt using a constant number of matrix manipulations. This is done by introducing the new parameter l which is equal to the desired message length. The secret key \mathbf{S} is generated the same way however it is now a $n \times l$ matrix. The public key is composed of \mathbf{A} (as before) and $\mathbf{B} = \mathbf{S}^T \mathbf{A} + \mathbf{X}$ where \mathbf{X} is an $l \times m$ matrix whose entries are independently generated from χ . Encryption and decryption are essentially the same as Regev's system, however references to \mathbf{s} and \mathbf{b} now refer to \mathbf{S} and \mathbf{B} respectively, and the bit p , is instead a vector \mathbf{p} of length l whose entries are each bit in the bit string.

This relatively simple improvement provides a massive speedup on encrypting multiple bits, as encrypting each bit sequentially is incredibly costly as each encryption instance requires multiple samplings of the public key and a modification of the plaintext bit. The amortisation cuts this down to just one sampling, and one matrix/vector multiplication.

Ring-SIS

Overview

Ring SIS can be thought of an improvement on SIS, using rings to create an analogous problem with a smaller m . As we showed earlier, this smaller m does not affect security but will provide efficiency improvements. SIS essentially ‘works in’ the product ring \mathbb{Z}_q^n , an integer ring. Micciancio [38] showed SIS’ weakness to an exhaustive search (relative to the size of m) and demonstrated Ring-SIS, a system which works in generalized rings which better resist the exhaustive search despite a smaller m .

Ring-SIS considers positive integers q, n, m and β as in SIS (i.e. q and n are some positive integers, m is the number of samples and β is the norm bound for our solution). We define a polynomial, degree n ring $R = \mathbb{Z}[\alpha]/f(\alpha)$ where $f(\alpha)$ is typically $\alpha^n - 1$ (such that every element of R is a polynomial of degree $< n$ and whose coefficients are in \mathbb{Z}). We also define a norm $\|\cdot\|$ on R as $\|\mathbf{z}\| = \sqrt{\sum_i \|\mathbf{z}_i\|^2}$ for a vector \mathbf{z} in R and finally define the quotient ring R_q such that $R/qR = \mathbb{Z}_q[\alpha]/f(\alpha)$.

The Ring-SIS problem is, given m uniformly random elements $a_i \in R_q$, which define a vector $\mathbf{a} \in R_q^m$ (roughly comparable to combining the m samples in SIS into a matrix form) find a nonzero vector $\mathbf{z} \in R^m$ such that:

$$F_{\mathbf{a}}(\mathbf{z}) := \langle \mathbf{a}, \mathbf{z} \rangle = \mathbf{0} \in R_q$$

adhering to the constraint $\|\mathbf{z}\| \leq \beta$. This $F_{\mathbf{a}}$ defines functions within a family, defined by the ring R . We can see this is very similar to the SIS problem.

Because there are an exponentially larger number of short ring elements that can be used to generate each $a_i \in R_q$ compared to the small number of $\mathbf{a}_i \in \mathbb{Z}_q^n$ with integer coefficients for traditional SIS, we are permitted a much smaller m , just $\log q$ as opposed to SIS’ $n \log q$.

Hardness

Evaluating the hardness of Ring-SIS is more complex as we must consider ideal lattices, a lattice corresponding to an ideal in R , when evaluating the hardness of lattice problems. Ideal lattices possess algebraic properties that can speedup certain problems (e.g. GapSVP is easy under an ideal lattice) however SVP and SIVP appear to be hard on ideal lattices. The choice of R is critical as, under certain choices of R , the function $F_{\mathbf{a}}$ is not collision-resistant [39] (as opposed to the SIS function $f_{\mathbf{A}}$ which is). However $F_{\mathbf{a}}$ is as hard to invert in the average case as a worst-case lattice problem on an ideal lattice [38]. Additionally solving Ring-SIS is as hard as the worst-case of SVP on ideal lattices [40].

Ring-LWE

Overview

Similarly to how Ring-SIS is comparable to an improvement on SIS, Ring-LWE can also be considered this way. Ring-LWE can provide faster running times and shorter ciphertexts by utilising rings. Similarly to Ring-SIS, the ring we use it typically a cyclotomic polynomial ring [30].

We define a ring R of degree n over \mathbb{Z} , a positive integer modulus q and an error distribution χ over R . For a secret $s \in R_q$ we define the ring-LWE distribution $A_{s,\chi}$ over $R_q \times R_q$ which produces an

output (a, b) where $a \in R_q$ is chosen uniformly at random and $b = s \cdot a + e \bmod q$ where $e \leftarrow \chi$. Once again, we use this to define two problems.

Search-Ring-LWE: Given m samples $(a_i, b_i) \in R_q \times R_q$ generated from the distribution $A_{s, \chi}$ with a fixed s , recover s .

Decision-Ring-LWE: Given m samples $(a_i, b_i) \in R_q \times R_q$ where each sample is either uniformly randomly generated or sampled from the distribution $A_{s, \chi}$ (for a fixed uniformly random secret), determine which sample are generated from $A_{s, \chi}$ with a non-negligible advantage.

Once again this is very clearly parallel to the original LWE problem. The use of rings allows many optimisations, for example given a sample (a_i, b_i) from $A_{s, \chi}$ both a_i and b_i are elements R_q whereas in traditional LWE, b_i represented just a scalar value in the integer mod ring. Additionally, the algebraic properties of this problem allow us to improve efficiency for computation times, for example we can now use Fourier Transform techniques to perform multiplication between ring elements, speeding up encryption times [41].

Hardness

Like Ring-SIS we must make careful decisions about our choice of R to ensure hardness, additionally it is advisable to choose χ at random from a family of distributions on R . Assuming appropriate constraints, Ring-LWE is, on average, at least as hard as a worst-case instance of SVP. The proof for this is obtained using a quantum security reduction proving Search-Ring-LWE is at least as hard as SVP, and then a classical reduction from Decision-Ring-LWE to Search-Ring-LWE [30].

Relation to cryptography

The intrinsic algebraic nature of these ideal lattices, pose some security doubts. For example, certain problems (e.g. GapSVP with sufficiently small γ) prove to be trivial on typical ideal lattices [40] (e.g. $x^n + 1$ ideal lattices). It should be noted however, that the values of γ required to trivially solve GapSVP on an ideal lattice are not cryptographically practical anyway, as such a small γ would already provide too little security for modern applications. Consequently, since the ideal lattices introduce algebraic properties, we may not be able to assign the same degree of confidence to their security as we do with their traditional counterparts. Though no algorithms have yet been produced that offer any significant speedups on typical ideal lattices we should remain cautious. Plainly, although ring-SIS/LWE security reductions to lattice problems are valid, we are not completely certain that these lattice problems are sufficiently hard on ideal lattices.

Both Ring-LWE and Ring-SIS, can be considered specific algebraic instances of LWE and SIS. These 'instances' allow us useful features due to these algebraic properties, such as smaller space requirements or efficiency optimisations. However they can provide the exact same cryptographic constructions as their non-ring counterparts. Thus the relation to cryptography is essentially providing an improvement upon traditional LWE/SIS based systems.

Fully Homomorphic Encryption

Overview

Fully Homomorphic Encryption (FHE) schemes have been proposed for both standard LWE and Ring-LWE. We consider the BFV [42] scheme which is based on Ring-LWE. We do this as, BFV provides a system that is more practical with faster computations and smaller keys. Additionally the conversation from Brakerski's LWE scheme [43] to this R-LWE adaptation is relatively trivial. The following is the scheme using relinearisation version 1:

As in R-LWE we define a ring $R = \mathbb{Z}[\alpha]/f(\alpha)$ where $f(\alpha)$ is some cyclomatic polynomial. We define R_n to be the set of polynomials in R whose coefficients belong to \mathbb{Z}_n . We then choose a positive integer modulus q and some error distribution χ over R . This provides our ring R_q under which all operations in our scheme take place, unless otherwise specified.

We sample a secret key $sk \leftarrow R_2$. We generate the public key $pk = (-a \cdot sk + e, a)$ where $a \leftarrow R_q$ and $e \leftarrow \chi$.

Encryption: for a plaintext message $m \in R_t$, where t is some positive integer determining the plaintext coefficient modulus, we define its encryption as:

$$ct = (p_0 \cdot u + e_1 + \Delta \cdot m, p_1 \cdot u + e_2)$$

where $\Delta = \lfloor q/t \rfloor$, $p_0 = pk[0]$, $p_1 = pk[1]$, $u \leftarrow R_2$ and $e_1, e_2 \leftarrow \chi$.

Decryption: the decryption of a ciphertext pair ct is defined as:

$$pt = \left\lfloor \frac{t \cdot c_0 + c_1 \cdot sk}{q} \right\rfloor \in R_t$$

where $c_0 = ct[0]$ and $c_1 = ct[1]$

Addition: for two ciphertexts ct_1 and ct_2 we define addition as:

$$ct = (ct_1[0] + ct_2[0], ct_1[1] + ct_2[1])$$

Multiplication: we first must compute a relinearisation key rlk , for version 1 multiplication this is defined as:

$$rlk = \left((-a_i \cdot sk + e_i) + T^i \cdot sk^2, a_i \right) : i \in [0, \dots, \ell]$$

For some positive integer T chosen such that the noise of relinearisation is smaller than that of multiplication, $\ell = \lfloor \log_T q \rfloor$ and $a_i \leftarrow R_q$, $e_i \leftarrow \chi$. We then compute the ciphertext values:

$$\begin{aligned} c_0 &= \left\lfloor \frac{t \cdot (ct_1[0] \cdot ct_2[0])}{q} \right\rfloor \\ c_1 &= \left\lfloor \frac{t \cdot (ct_1[0] \cdot ct_2[1] + ct_1[1] \cdot ct_2[0])}{q} \right\rfloor \\ c_2 &= \left\lfloor \frac{t \cdot (ct_1[1] \cdot ct_2[1])}{q} \right\rfloor \end{aligned}$$

Note that to obtain these values we are simply multiplying the polynomials ct_1 and ct_2 and then scaling the result by $q/t \cong \Delta$ before rounding. Then we must reduce these three values, into just two to create a valid ciphertext. We do this using a process called relinearisation, making use of the rlk calculated previously. We first modify c_2 such that it is in base T and then the final ciphertext is (c'_0, c'_1) with:

$$\begin{aligned} c'_0 &= c_0 + \sum_{i=0}^{\ell} rlk[i][0] \cdot c_2^{(i)} \\ c'_1 &= c_1 + \sum_{i=0}^{\ell} rlk[i][1] \cdot c_2^{(i)} \end{aligned}$$

Where $c_2^{(i)}$ denotes the coefficient i in the base T polynomial c_2 .

Correctness

Relinearisation converts three ciphertexts into two while preserving the encryption correction property:

$$c_0 + c_1 \cdot sk + c_2 \cdot sk^2 = c_0 + c_1 \cdot sk + r$$

This is done by attempting to approximate $c_2 \cdot sk^2$ as $c_2 \cdot rlk[0] + rlk[1] \cdot sk$. Recall that $rlk[0] + rlk[1] \cdot sk = sk^2 + e_0$ as defined in the relinearisation key. This introduces a problem however that our error e_0 gets multiplied by c_2 (a random polynomial in R), massively increasing the noise r . To rectify this our relinearisation algorithm slices c_2 into polynomials of a small norm and then multiplies with slices of rlk accordingly. We can see that the initial multiplication preserves the encryption correction property and the relinearisation retains this with a small error. For more details on the correctness of this algorithm and the noise introduced in each step see the full BFV paper [42].

Noise: You may notice that we sample sk and u from R_2 rather than R . Choosing the secret key from this smaller distribution has just a minor effect on security [44] but reduces the noise of encryption. The noise is the error introduced in each step, and if this grows large we will be unable to perform correct decryption. We define the noise of encryption as $v = u \cdot e + e_1 + e_2 \cdot s$, thus it is evident that reducing the coefficient sizes of u and s reduces the noise that encryption introduces, allowing a more efficient system with deeper multiplicative depth.

Hardness

As this scheme is essentially an extension of an Ring-LWE cryptosystem [30], with addition and multiplication operations, the scheme's security is equivalent to that of R-LWE. Note that the addition and multiplication operations do not provide any extra information about the encrypted data. This is because the new ciphertext is identical to the ciphertext in which the operations are performed on the plaintext first and then encrypted. It should be noted that this scheme intrinsically does not provide message integrity, since we require that our ciphertexts can be modified into new and valid ciphertexts.

Relation to Cryptography

This system provides a very secure, reasonable efficient system in which arbitrary computations may be performed on encrypted data. The scheme presented above is a levelled homomorphic system, which means the number of multiplications is bounded, but may be made arbitrarily large. This scheme can be extended even further to a true fully homomorphic system using bootstrapping [14]. This construction may prove to be a very useful tool and the project implements the above scheme to demonstrate its use. The project does not incorporate bootstrapping due to the large computational overhead, but it does provide an instantiation of the system with parameters that allow a bootstrapping based FHE implementation.

Problem choice and justification for project

Given the desirability to make a system capable of implementing various cryptographic functions, I have opted to disregard SIS and Ring-SIS as while they are suitably hard and efficient, they lack the capability to implement a public key system. We now must choose between LWE and its Ring variant. Given the main drawback and primary concern of most quantum secure cryptography mechanisms is efficiency, I have chosen to use Ring-LWE for the final deliverable. Ring-LWE provides cost improvements on traditional LWE and though less deeply investigated, the lattice problems on ideal lattices in rings appear to be as hard as worst-case lattice problems on arbitrary lattices. In the project we implement both a standard LWE encryption scheme and a Ring-LWE (with homomorphic encryption) scheme. We additionally create a benchmarking program to demonstrate the efficiency and space benefits Ring-LWE provides.

Software Engineering

Summary:

This section provides an overview of the software engineering processes used in the project including methodologies, UML diagrams and testing. We also highlight and explain various important files and sections of the code. Additionally we provide instructions on installing and using the project files.

Project Structure

The project code contains two directories: 'Notebooks' and 'SAGE'. The code contained within each is identical, they are just structured in a different way to allow easier interaction. The notebook directory contains various Jupyter notebook files (.ipynb). Cells are annotated with markdown describing their function and various choices made in the code. These notebooks provide an easy-to-understand walkthrough of the project. Where relevant, the notebooks also contain unit tests for itself.

The 'SAGE' directory contains SageMath files (.sage) and standard Python files (.py). These provide a more traditional project structure. Inside the Sage directory is a directory named 'LWE'. This directory contains the LWE module that is the bulk of the project, containing the LWE encryption system and the R-LWE fully homomorphic encryption system. Each file in the module is accompanied with a test.py file which contains unit tests for the corresponding file. Returning to the 'SAGE' directory, we can see various high-level sage files. These contains high level programs that use the LWE module, both providing useful functions and demonstrating usage of the module.

Project File details

Name: LWE_PKE

Content: This file contains a PKE system based on LWE as detailed in Regev's paper [32] and an amortisation improvement [16]. This code is written in Python using Sage. Documentation for the code is available as traditional Python DocStrings as well as the Markdown text sections of the Notebook (if applicable).

Purpose: This code provides a full implementation of an LWE crypto system that provides PKE. It includes a factory function createLWE that returns an LWE instance, either traditional or amortised.

Name: FHE_RLWE

Content: This file contains a Fully Homomorphic Encryption system, whose security is based on Ring-LWE. As such, it performs encryption and decryption on polynomial plaintexts and supports addition and multiplication operations on encrypted ciphertexts. Documentation for the code is available as traditional Python DocStrings as well as the Markdown text sections of the Notebook (if applicable).

Purpose: Provides a full Levelled Fully Homomorphic Encryption system demonstrating the possible applications of lattice-based cryptography. It includes a createFHE function which returns an LWE instance, either with some reasonable parameters [45] or with parameters suitable for bootstrapping.

Name: CLI

Content: A singleton command-line interface for the LWE module implementing a simple PKE message exchange between two unique entities Alice and Bob. The CLI maintains a handful of variables and allows encryption, decryption and (in the case of FHE) manipulation. The CLI supports all basic operations one would use an encryption system for. An example execution of this code

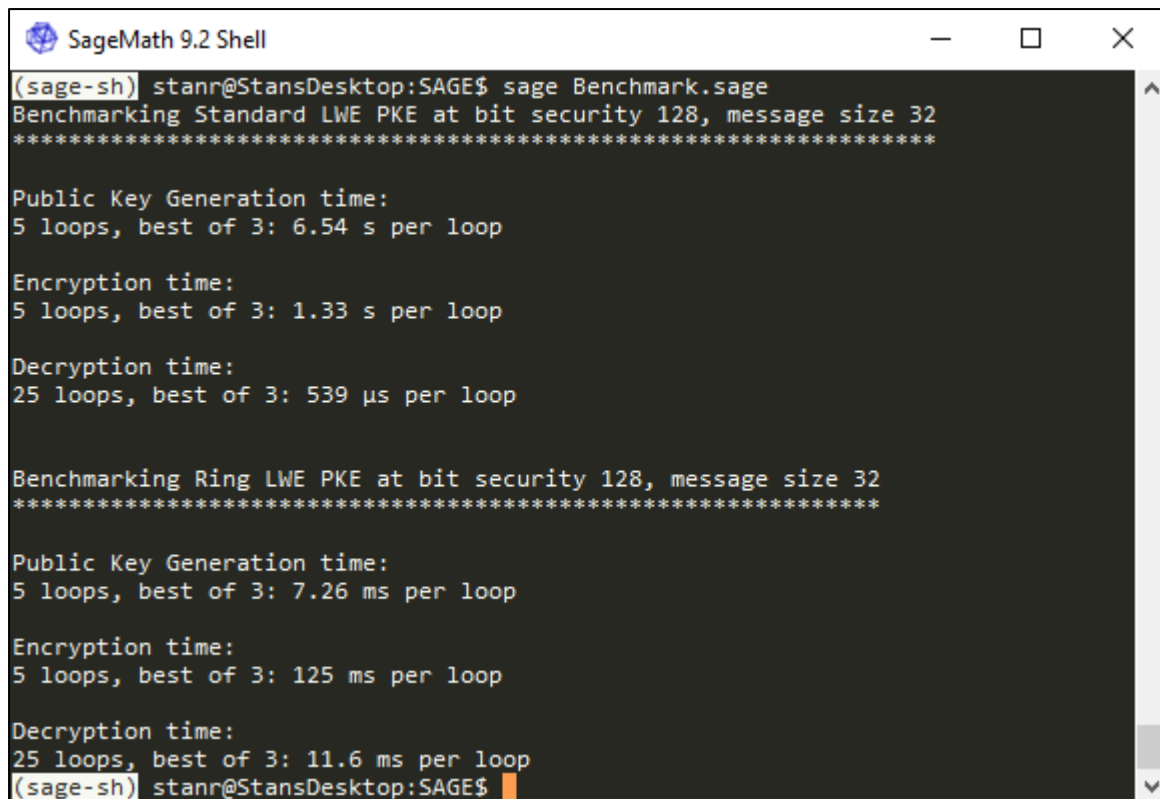
(using the FHE R-LWE module) is available on YouTube: <https://youtu.be/fNn73rmjHDo>

Purpose: This program demonstrates the usage of the LWE module to provide a distinct LWE instances with which to encrypt/decrypt LWE messages.

Name: Benchmark

Content: This program uses SageMath's timeit implementation to time various computations with both the standard and ring LWE systems.

Purpose: This serves to establish estimates of average running times and demonstrates the efficiency benefits that Ring-LWE displays over standard LWE. Here is a sample output after running the program:



```
(sage-sh) stanr@StansDesktop:SAGE$ sage Benchmark.sage
Benchmarking Standard LWE PKE at bit security 128, message size 32
*****

Public Key Generation time:
5 loops, best of 3: 6.54 s per loop

Encryption time:
5 loops, best of 3: 1.33 s per loop

Decryption time:
25 loops, best of 3: 539 µs per loop

Benchmarking Ring LWE PKE at bit security 128, message size 32
*****

Public Key Generation time:
5 loops, best of 3: 7.26 ms per loop

Encryption time:
5 loops, best of 3: 125 ms per loop

Decryption time:
25 loops, best of 3: 11.6 ms per loop
(sage-sh) stanr@StansDesktop:SAGE$
```

Name: demo_lwe.ipynb

Content: A very simple notebook showing a simple implementation of the LWE module. An example execution of this notebook is available on YouTube: <https://youtu.be/0vAuaHWUA5w>

Purpose: Demonstration of the LWE module in action, encrypting single and multibit messages between two entities: Alice and Bob.

Name: demo_fhe.ipynb

Content: A very simple notebook showing a simple implementation of the FHE module. An example execution of this notebook is available on YouTube: <https://youtu.be/frZouu9As8Q>

Purpose: Demonstration of the FHE module in action, encrypting polynomial messages between two entities: Alice and Bob and then performing homomorphic operations on the ciphertexts.

Using project files: All notebook files (.ipynb) are runnable as IPython notebooks (e.g. using Jupyter). They can also be run via the Sage shell using `'sage -n jupyter [filename].ipynb'`. Sage files (.sage) are runnable in the Sage shell using `'sage [filename]'` and Python files (.py) are runnable in the Sage shell using `'python [filename]'`. Finally, if you wish to write your own program using the project, the LWE module is importable using standard Python imports, e.g. `'import [LWE.module.filename]'` (note that

importing the notebook files is also possible using the `'import_ipynb'` python module). SageMath must be installed on the machine running any of these files.

Software Engineering

Testing

When developing the system, I used a unit-test based Test-Driven Development approach to build components.

Unit tests for Sage files are included inside the module as a separate Python (.py) file. Unit tests for notebooks are implemented in a notebook cell within the file and test various classes and functions through that module. Running the cell will run the unit tests however unit tests are not run when importing the module unless explicitly called. This approach allows TDD while using the Notebook format in a logical way and retaining modularity.

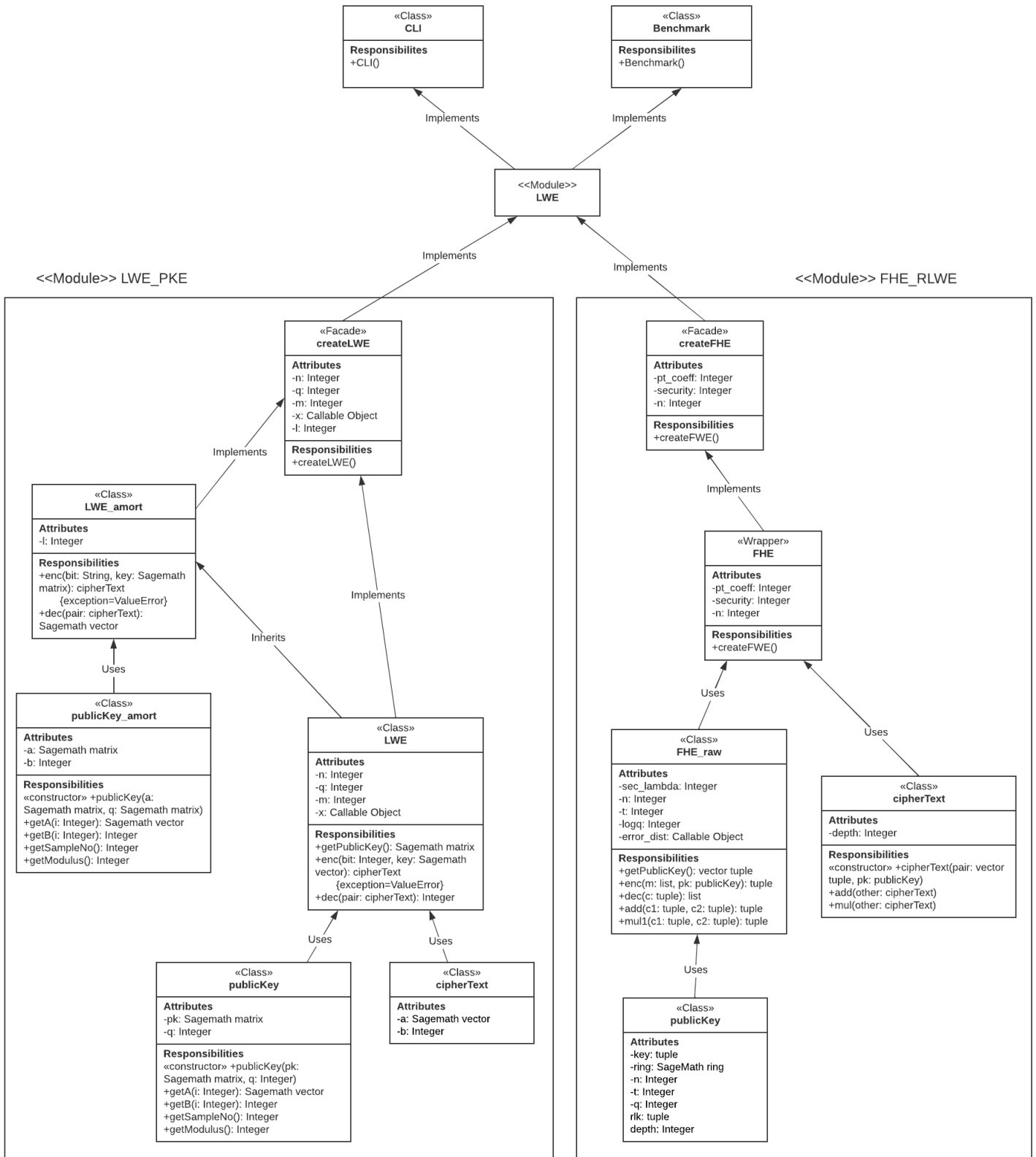
Most tests either test the normal case (for which a random instance is generated) or test an error case (for which an unacceptable instance is generated). These unit tests are run after any modification to the associated component, to ensure the changes made do not affect the existing functionality or logic of the code.

Methodology

Given the program is developed using Python I decided to follow an engineering framework that intrinsically synergizes with Python: modular programming (which is similar to the OOP pattern of interface-based programming) . The LWE system is coded as a distinct module, capable of being used externally without any dependencies or coupling, however internally it is designed in an Object-Oriented way with classes representing logical objects. We also define a singleton CLI which provides a user interface for the LWE module and provides a blueprint for implementing the LWE module for message encryption and decryption. While most of the code was written using Jupyter notebook, I used the version control system SVN to track changes to the notebook files. Additionally, to facilitate the modular programming/OOP approach, I created a UML diagram to detail the basic program structure, which is as follows:

LWE UML

Stanley Roberts | February 17, 2022



As evident in the above diagram, the program follows a modular programming and OOP hybrid, while logical objects interact, each module is entirely self-contained with minimal dependencies.

Project Development

Summary:

This section provides an overview of how the project progressed and developed. We review a timeline of the project and review various project decisions such as scope increases. We also provide the various resources used in each stage as well as problems we encountered.

Diary

Week	Achieved	Plan	Problems, Notes and Resources
1 st Oct to 10 th Oct	I took the initial week to familiarise myself with the foundational theory behind quantum algorithms, and their relation to post quantum cryptography, and research into the ‘threat’ of quantum computing and the reliability, scale, and speed of this threat. With this knowledge I completed the first part of Motivation Report (Abstract and Motivation sections).	My plan for the next week was to review the materials required to complete the Motivation Report and following that, to complete the report	I noticed I was heading somewhat out of scope in reviewing the workings of specific quantum algorithms, as a thorough understanding of the mechanics behind these algorithms is not necessary for this project. Consequently I focused my research away from the mechanical aspect and towards the specific motivation for post-quantum crypto. This is directly working toward the ‘Motivation for Quantum Cryptography Report’ outlined in my plan. Resources: [2, 46, 47]
10 th Oct to 17 th Oct	This week I delved into more technical research, such as the necessary requirements for post-quantum cryptography (e.g. NIST standards) and the various prominent mechanisms of implementing PQC. I used this to complete my Motivation Report (Requirements and Mechanisms sections).	For the next week I aimed to setup a coding environment begin the lattice implementation, as well as starting the Problem Report (from initial plan)	This marks the completion of the ‘Motivation for Quantum Cryptography Report’ detailed in my plan. Resources: [32, 48]
17 th Oct to 24 th Oct	I began the Problem Report, starting by giving a technical overview of all the relevant problems. I also setup a Sage environment with some testing programs and familiarised myself with the Sage interpreter and	Given the relatively solid theoretical work done so far providing a solid base I wanted to properly begin the LWE lattice implementation.	I decided to prioritise the Problem Report as I realised a stronger theoretical base would speed up the coding implementation of the LWE system. Resources: [32, 48, 49]

	functions (using the test apps and docs).		
24 th Oct to 31 st Oct	I developed several tests to motivate the TDD process and completed a rough draft of the LWE module including class and function placeholders and rigorous definition of the parameters, such as error distribution and secret generation. Additionally I improved the Problem Report by adding a hardness and relevance section to each detailed problem.	The plan for the next week is a continuation of this week, aiming to progress towards (or succeed in) completing the LWE system.	This week marks completion of the 'Lattice Structures and Problems Report' and the 'Lattice Hardness Report' and significant progress on the 'Lattice Scheme Implementation Program', all of which are defined in my initial plan. Resources: [48-52]
31 st Oct to 7 th Nov	I finished developing a fully functional LWE PKE system, through addition of an LWE sampler and fully working bitwise encryption and decryption functions. I also introduced some new unit tests (see notes).	Given the functional state of code, moving forward I wanted to review my reports and provide additional functionality to the LWE program.	I slightly increased the scope of the program code, including unit tests for functions that encrypt a string and decrypt a matrix. The motivation behind this is that these should be relatively simple extensions of the existing enc/dec functions. This week solely consisted of working on (and completing) the 'Lattice Scheme Implementation Program' as specified in my initial plan. Resources: [50]
7 th Nov to 14 th Nov	This week involved mainly reviewing and improving past work. Better modularisation capabilities were introduced into the code and both the Motivation Report and Problem Report were reviewed and updated. Additionally I started writing the Interim Report (Abstract, Project Files and part of the engineering methodology)	Given the upcoming Interim Review, I wanted to complete the Interim Report. I also wished to improve upon the modularisation of the LWE system, perhaps using an example CLI which implements the system.	This week allowed me to review and improve all my deliverables outlined in report. Resources: [48, 50]
14 th Nov to	I added a functional CLI module which implements the LWE	After review of my planned interim submission by supervisor	This CLI development is technically out of my initial plan's scope, but it is a logical extension that helps with

21 st Nov	system, demonstrating the modularisation and use of the developed system. I also completed the interim review, completing the engineering testing and methodology, the project diary and fleshing out the previous sections)	I aim to rectify any criticisms they may have. I also wish to complete the String/Matrix implementation of the LWE system.	presentation and given all other deliverables are completed, it's a worthwhile (and small) scope increase.
21 st Nov to 28 th Nov	I added amortisation to improve the efficiency of encrypted multiple bits and finalised the interim report by consolidating the various deliverables outlined in my plan	With the interim review deliverables finalised next week is to be spent focusing on the presentation	Resources: [30]
9 th Jan to 23 rd Jan	I reviewed various FHE materials and used these to implement a homomorphic scheme. Additionally I restructured the repository.	I wanted to incorporate version 2 relinearisation and bootstrapping into the FHE scheme	The restructuring mainly consisted of modifications based on interim feedback (such as adding a readme). After discussion with my supervisor, I decided to increase my scope to include a FHE scheme as it seemed achievable, given the progress made so far. Resources: [42, 45, 53-55]
23 rd Jan to 6 th Feb	Added the mathematical foundation for FHE to the report. Additionally updated the report to match project feedback and include the new FHE system in the software engineering section.	My plan for next week was to tidy the code and flesh out the CLI program into a less linear CLI that functioned as a pseudo-shell.	I decided against spending time adding v2 relinearisation and bootstrapping as these provide little benefit over the functions already implemented, and instead focused on completing the report to match the current project state.
6 th Feb to 13 th Feb	I vastly improved the CLI, allowing it to use both the standard and ring schemes as well as managing a set of variables that can be modified and displayed at will. I additionally added more comprehensive unit tests to the FHE system.	My plan for next week was to finalise the report, and review revisiting v2 relinearisation.	

Professional Issues

Any project in computing is guaranteed to encounter some ethical issues, we highlight these here to demonstrate our adherence of good practice and any implications this project may have.

Firstly, all code produced during this project was written entirely independently. The code uses standard python modules and SageMath [56] which is licensed under GNU GPL. Python itself is licensed under PSFL [57]. This project itself is private and thus is not licensed.

In this report we raise issues with the current model of encryption and its potential vulnerability to quantum computers, however this is done to establish the importance of quantum-secure encryption and no concrete methods are given to break existing encryption. Additionally this report is not publicly available so we can be assured that we are not providing tools or information that may be misused.

Throughout this project we adhere to the ACM code of ethics [58]: following common ethical principles, producing high quality and (where appropriate) secure professional work. As a single author, the leadership principles do not apply however regular meetings with my project supervisor are used to ensure compliance to the ACM code and common ethics.

The largest point of concern regarding professional issues for this project is the assurance of a secure system. This issue has both ethical considerations, as we do not wish to unwittingly provide false security assurances, and professional concerns, as we must create a high quality and transparent system. We mitigate this potential issue through a variety of means. Code is thoroughly documented to provide transparency behind the implementation of the system. Unit tests are constructed for each component of the code to ensure robustness and correctness. We also query a wide range of literature, and closely follow the results in these, to provide algorithms with well-established security and hardness values.

With such a theoretical and literature heavy project we must follow proper plagiarism conventions, both in the report and the code produced. Any component of the report that makes use of other work, is properly referenced. Additionally all components in the code rely only on the work written in this report. As such, we can remain confident that all content used during this report, is correctly cited. Some resources, such as YouTube videos or other tutorials/surveys, were also used as a reference while developing the project report and code. In the interest of complete transparency and plagiarism obedience, we list these in the Project Diary where appropriate.

Fully Homomorphic Encryption Extension

As the Fully Homomorphic System was not initially part of my project plan and is a rather significant extension, we wish to discuss this further. Recall that the initial project goal was to provide implementations of various lattice algorithms that implement Public Key Encryption. The construction of FHE surpasses these initial goals, introducing a unique form of encryption that is enabled thanks to lattice cryptography.

This is foremost a project on lattice cryptography and upon discovering this possibility, I felt it was far too compelling to not demonstrate. FHE had been an open problem for over 30 years [59] until Gentry [14] proposed a solution based on lattice cryptography. FHE allows arbitrary computations on encrypted data by an untrusted third party. Such a concept allows 'bypassing' of data regulations, allowing institutions to share private data in its encrypted form for third parties to operate upon. One such example is cloud computing, allowing parties with less computational power to offload manipulations on sensitive data to another party without risking a breach of any privacy law.

Though we go into detail in the theory section, we will briefly reiterate the form of our FHE scheme to highlight the extra achievements we have completed. We incorporate a FHE scheme based on Ring-LWE. Plaintexts and ciphertexts are expressed in the form of polynomials of some fixed degree. As with the other schemes produced, this FHE scheme supports the standard PKE operations including, encryption, decryption, and public/private key generation. We also support addition and multiplication operations on ciphertexts. These functions produce ciphertexts, which when decrypted, provide plaintexts equivalent to performing addition or multiplication, respectively, on the original plaintexts.

The scheme is bounded by an arbitrary number of multiplications, making it a levelled scheme, however this bound may be modified at will. We also provide the framework for a non-levelled system whose parameters are defined as recommended in the BFV paper [42], allowing an unbounded fully homomorphic system, given a bootstrap function.

While this FHE scheme is beyond the initial scope of the project, it fits quite effectively; demonstrating the additional capabilities lattice cryptography can provide. The undertaking was of course discussed with my project supervisor to ensure that it was feasible to produce this extension to a high degree of quality, which I believe has been done.

Possible Continuations

There are numerous ways I would have wished to continue this project had more time been available. I shall cover some of the possible directions further work may have taken.

One extension would be to provide a system based on Module-LWE (MLWE) [60]. MLWE is closely related to the Ring-LWE scheme [61] and involves using module elements on a given polynomial ring, rather than single polynomial elements of the ring. You might consider the module \rightarrow ring-element (which in this case is a polynomial) relationship as analogous to the vector \rightarrow scalar relationship.

Due to the similar nature of RLWE and MLWE this would be a relatively small extension but would serve to flesh out the variety of lattice-based algorithms we provide, with our system now providing Public Key Encryption systems based on each of the major LWE variants (standard, ring, and module).

A further continuation would be to build upon the FHE scheme we have implemented and introduce parallel schemes as alternatives, such as CKKS [62] and BGV [63]. This would further the FHE extension we have already supplied and would allow us to run benchmarks against these different systems to compare their efficiency.

One final direction would be to explore the other uses of lattice-cryptography. Namely, Attribute Based Encryption (ABE) [64], Identity Based Encryption (IBE) and its hierarchical generalisation (H)IBE [65]. Implementing these will allow us to further explore the different higher-level functionalities lattice-based cryptography can provide.

Final Reflection

The initial scope of this project was to implement a lattice-based encryption scheme. We have completed this in our LWE_PKE module; but we also provided numerous other extensions of this initial plan. We have implemented optimisations on the LWE encryption system (such as amortisation). Additionally we have implemented an encryption scheme based on Ring-LWE. This system also provides a Levelled Homomorphic Encryption scheme to perform arbitrary computations on encrypted data. We have also implemented some high-level interactions with the

LWE module, namely a command line interface to perform simple computations within the system and a benchmarking program to evaluate and demonstrate the efficiency of Ring-LWE versus standard LWE.

We completed everything laid out in the project plan, as well as numerous extensions. We also managed to avoid any serious incidences of the risks detailed in this plan, thanks to the mitigation steps.

Overall we have managed to provide:

- An in-depth report on the motivation behind lattice cryptography, and the relevance of the rise of quantum computers. This report also contains an explanation of Fully Homomorphic Encryption and why such a system is desirable.
- An overview of the mathematical theory behind lattice algorithms necessary to implement post-quantum encryption. We also provided the mathematical explanation for homomorphic encryption. Additionally we provided reasonable security assumptions and reductions for the mentioned algorithms and problems.
- A software development report which highlights and explains many important design decisions in the development and coding of the project. We also detail how to use the code and the overall repository structure.
- A final analysis of the project including a diary tracking the progress of the project
- A SageMath implementation of an LWE-based encryption scheme and a Ring-LWE based levelled homomorphic encryption scheme, both with relevant unit test. We also provide a command line interface and a benchmarking program for these schemes.

References

- [1] Anonymous "*IBM Unveils World's First Integrated Quantum Computing System for Commercial Use*", Jan 8, Retrieved: October 17, Available: <https://search.proquest.com/docview/2164398922>.
- [2] P.W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," SIAM Rev., vol. 41, pp. 303-332, January 1., 1999.
- [3] P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," pp. 124-134, November 1994.
- [4] S. Jiang, et al, "Quantum Annealing for Prime Factorization," Scientific Reports, vol. 8, pp. 17667, 2018.
- [5] L.K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," in Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pp. 212-219, 1996.
- [6] D. Aggarwal and U. Maurer, "Breaking RSA Generically Is Equivalent to Factoring," in Advances in Cryptology - EUROCRYPT 20, pp. 36-53, 2009.
- [7] V. Mavroeidis, et al, "The Impact of Quantum Computing on Present Cryptography," International Journal of Advanced Computer Science and Applications, vol. 9, 2018.

- [8] Lux Research. *"Lux Research / Quantum Computing Executive Summary"*, Retrieved: Oct 17, Available: <https://www.luxresearchinc.com/quantum-computing-executive-summary>.
- [9] Martin Giles. *"We'd have more quantum computers if it weren't so hard to find the damn cables"*, Retrieved: Oct 17, Available: <https://www.technologyreview.com/2019/01/17/137811/quantum-computers-component-shortage/>.
- [10] D.P. DiVincenzo, "The Physical Implementation of Quantum Computation," *Fortschritte Der Physik*, vol. 48, pp. 771-783, 2000.
- [11] Katia Moskvitch. *"The Argument Against Quantum Computers"*, -02-07T12:59-05:00 Retrieved: Oct 17, Available: <https://www.quantamagazine.org/gil-kalais-argument-against-quantum-computers-20180207/>.
- [12] R. Babbush, et al, "Focus beyond Quadratic Speedups for Error-Corrected Quantum Advantage," *PRX Quantum*, vol. 2, pp. 010103, Mar. 2021.
- [13] Andreas Baumhof. *"How to Steal 10 Billion USD in Bitcoin with a Quantum Computer"*, Retrieved: Oct 17, Available: <https://www.quintessencelabs.com/blog/steal-10-billion-usd-bitcoin-quantum-computer/>.
- [14] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, pp. 169–178, 2009.
- [15] O. Regev, "The Learning with Errors Problem (Invited Survey)," pp. 191-204, 2010.
- [16] C. Peikert, V. Vaikuntanathan and B. Waters, "A Framework for Efficient and Composable Oblivious Transfer," in *Advances in Cryptology – CRYPTO 20*, pp. 554-571, 2008.
- [17] Anonymous "Microsoft SEAL (release 3.7)," sep. 2021.
- [18] Computer Security Division, Information Technology Laboratory. *"Post-Quantum Cryptography Standardization - Call for Proposals"*, -01-03 Retrieved: Oct 17, Available: <https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization>.
- [19] Anonymous *"ETSI GR QSC 001 V1.1.1 (2016-07) - Quantum-Safe Cryptography (QSC); Quantum-safe algorithmic framework"*, Retrieved: October 17, Available: <https://standards.iteh.ai/catalog/standards/etsi/f92675fa-fc19-4e00-9e21-531eeb3e2409/etsi-gr-qsc-001-v1-1-1-2016-07>.
- [20] R.A. Perlner and D.A. Cooper, "Quantum Resistant Public Key Cryptography: A Survey," in *Proceedings of the 8th Symposium on Identity and Trust on the Internet*, pp. 85–93, 2009.
- [21] W. Beullens, J. D'Anvers, A. Husling, T. Lange, L. Panny, C. Guilhem, N. Smart. *"Post-Quantum Cryptography: Current state and quantum mitigation"*, May Retrieved: October

17, Available: <https://www.enisa.europa.eu/publications/post-quantum-cryptography-current-state-and-quantum-mitigation>.

[22] D. Cooper, et al, "Recommendation for Stateful Hash-Based Signature Schemes," 2020.

[23] G. Davida, Y. Desmedt and R. Peralta, "A Key Distribution System Based On Any One-Way Function," in Advances in Cryptology — EUROCRYPT '89, Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 75-79.

[24] H. Dinh, C. Moore and A. Russell, "McEliece and Niederreiter Cryptosystems That Resist Quantum Fourier Sampling Attacks," in Advances in Cryptology – CRYPTO 20, pp. 761-779, 2011.

[25] D. Augot, et al, "Initial recommendations of long-term secure post-quantum systems," 2015.

[26] D. Jao and L. De Feo, "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies," in Post-Quantum, pp. 19-34, 2011.

[27] J.H. Silverman, "The Geometry of Elliptic Curves," in The Arithmetic of Elliptic Curves, J.H. Silverman, New York, NY: Springer New York, 2009, pp. 42-47.

[28] D. Jao and L. De Feo, "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies," in Post-Quantum, pp. 19-34, 2011.

[29] J. Bos, et al, "Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1006–1018, 2016.

[30] V. Lyubashevsky, C. Peikert and O. Regev, "On Ideal Lattices and Learning with Errors over Rings," J.Acm, vol. 60, nov. 2013.

[31] X. Lu and J. Zhang, "Lattice-based PKEs/KEMs," Natl Sci Rev, vol. 8, 2021.

[32] O. Regev, "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography," J.Acm, vol. 56, sep. 2009.

[33] M.' Ajtai, "Generating Hard Instances of the Short Basis Problem," in Automata, Languages and Programming, pp. 1-9, 1999.

[34] C. Gentry, C. Peikert and V. Vaikuntanathan, "Trapdoors for Hard Lattices and New Cryptographic Constructions," in Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, pp. 197–206, 2008.

[35] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 372-381, 2004.

- [36] D. Micciancio and C. Peikert, "Hardness of SIS and LWE with Small Parameters," in *Advances in Cryptology – CRYPTO 20*, pp. 21-39, 2013.
- [37] T. Holenstein, "Pseudorandom Generators from One-Way Functions: A Simple Construction for Any Hardness," in *Theory of*, pp. 443-461, 2006.
- [38] D. Micciancio, "Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions," *Computational Complexity*, vol. 16, pp. 365-411, December 1, 2007.
- [39] C. Peikert and A. Rosen, "Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices," in *Theory of*, pp. 145-166, 2006.
- [40] C. Peikert and A. Rosen, "Lattices That Admit Logarithmic Worst-Case to Average-Case Connection Factors," in *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pp. 478–487, 2007.
- [41] V. Lyubashevsky, C. Peikert and O. Regev, "A Toolkit for Ring-LWE Cryptography," in *Advances in Cryptology – EUROCRYPT 20*, pp. 35-54, 2013.
- [42] J. Fan and F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption," 2012.
- [43] Z. Brakerski, "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP," in *Advances in Cryptology – CRYPTO 20*, pp. 868-886, 2012.
- [44] S. Goldwasser, et al, "Robustness of the Learning with Errors Assumption," *Other University Web Domain*, -01. 2010.
- [45] M. Albrecht, et al, "Homomorphic Encryption Security Standard," *HomomorphicEncryption.org*; *HomomorphicEncryption.org*, 2018.
- [46] L.K. Grover, "Quantum Mechanics Helps in Searching for a Needle in a Haystack," *Phys. Rev. Lett.*, vol. 79, pp. 325-328, July 14, 1997.
- [47] Microsoft. "*Quantum Computing for Computer Scientists*", May 14, Retrieved: November 18, .
- [48] C. Peikert, "A Decade of Lattice Cryptography," *Tcs*, vol. 10, pp. 283-424, /3/23. 2016.
- [49] Martin Albrecht and Leo Ducas. "*Sage for Lattice-Based Cryptography*", Retrieved: December 10, Available: <https://www.maths.ox.ac.uk/system/files/attachments/sage-introduction.pdf>.
- [50] Anonymous "*Sage Documentation v9.4*", Retrieved: December 10, Available: <https://doc.sagemath.org/html/en/index.html>.
- [51] N.A. Alkadri, et al, "A Framework to Select Parameters for Lattice-Based Cryptography," *IACR Cryptol. ePrint Arch.*, 2017.

- [52] R. Lindner and C. Peikert, "Better Key Sizes (and Attacks) for LWE-Based Encryption," in Topics in Cryptology – CT-RSA 20, pp. 319-339, 2011.
- [53] A. Kim, et al, "General Bootstrapping Approach for RLWE-based Homomorphic Encryption," 2021.
- [54] S. Sinha Roy, et al, "FPGA-Based High-Performance Parallel Architecture for Homomorphic Computing on Encrypted Data," in 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 387-398, 2019.
- [55] Anonymous "*Introduction to the BFV encryption scheme*", Retrieved: Feb 18, Available: <https://inferati.com/blog/fhe-schemes-bfv>.
- [56] Anonymous "*SageMath Mathematical Software System - Sage*", Retrieved: Mar 7, Available: <https://www.sagemath.org/>.
- [57] Anonymous "*History and License - Python*", Retrieved: March 07, Available: <https://docs.python.org/3/license.html>.
- [58] Anonymous "*ACM Code of Ethics and Professional Conduct*", Retrieved: Mar 7, Available: <https://www.acm.org/code-of-ethics>.
- [59] R.L. Rivest and M.L. Dertouzos, "ON DATA BANKS AND PRIVACY HOMOMORPHISMS," 1978.
- [60] A. Langlois and D. Stehle, "Worst-Case to Average-Case Reductions for Module Lattices," 2012.
- [61] M. Albrecht and A. Deo, "Large Modulus Ring-LWE \geq Module-LWE," pp. 267-296, 2017.
- [62] J.H. Cheon, et al, "Homomorphic Encryption for Arithmetic of Approximate Numbers," in Advances in Cryptology -- ASIACRYPT 2017, pp. 409-437, 2017.
- [63] Z. Brakerski, C. Gentry and V. Vaikuntanathan, "(Leveled) Fully Homomorphic Encryption without Bootstrapping," in Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, pp. 309–325, 2012.
- [64] E. Affum, et al, "Efficient Lattice CP-ABE AC Scheme Supporting Reduced-OBDD Structure for CCN/NDN," Symmetry, vol. 12, 2020.
- [65] S. Agrawal, D. Boneh and X. Boyen, "Efficient Lattice (H)IBE in the Standard Model," in Advances in Cryptology -- EUROCRYPT 2010, pp. 553-572, 2010.