Quantum Secure Encryption Hardware Implementation of Ring Learning With Errors

Anthony Fortner, Lahiru Perera, Vasanth Sadhasivan

June 14, 2019

Contents

1 Introduction

1.1 Project Background

As the world become more dependent of technology the need to secure our data become a growing concern. Though there are numerous security algorithm that we use today, none are expected to be computationally robust against the processing power of quantum computers. Learning with errors (LWE) is different in the since that it is based on lattice cryptography, a method though to be more resistant to an attack due to its higher complexity.

1.2 Potential Beneficiaries

With the growing need to secure data the possible applications for LWE are countless. The future of computing is widely thought to become dominated my quantum computers and LWE has the potential to become our primary means to secure data. This would allow alleviate one of the major concerns of quantum computers and assist in their adoption.

2 Research & Planning

2.1 LWE Basics

2.1.1 Private Key Generation

To generate the private key Alice simply needs to come up with a vector that is in the vector space of n integers modulo q or Z_q^n

$$s \in Z_q^n$$

The vector elements are chosen uniformly at random $q \geq 2$, is a prime number between n^2 and $2n^2$

2.1.2 Public Key Generation

Initially Alice choose m vectors from Z_q^n uniformly and independently, let this set of vectors be set A

$$A = a_1, a_2 \ldots a_m \in Z_q^n$$

Then she choose a random set of m errors

$$E = e_1, e_2 \ldots e_m \in Z_q^n$$

Error values are selected according to the distribution χ

$$m = (1 + y)(n + 1)log(q)$$

for an arbitrary y

2.2 Why Ring LWE

3 Prototype design

3.1 Other Components

3.1.1 Gaussian Number Generator

This component is used primarily in the generation of terms for the polynomial's coefficients. Our goal was to use develop an implementation that uses the Ziggurat algorithm. From our research we determined that this algorithm would give us the best results while not requiring too many additional resources to implement.

3.1.2 NTT

The nonlinear fast Fourier Transform (NFT) is a critical component used in various parts of the encryption/decryption process. The primary use is to quickly perform the needed operations on the polynomials that hold the message to be encrypted/decrypted.

3.1.3 Control Unit

The control unit is the component that will enable and disable various subsystem depending on the state that the system is in. Many of the system are designed to be controlled asynchronously such as the loading of gaussian random numbers into memory. There for the control unit's functionality is extremely important in making sure that the needed tasks are done in the correct order and duration.

3.2 Core Functions

Our hardware implementation consists of two main functions, encryption, and decryption.

3.2.1 Private Key Generation

3.2.2 Encryption

3.2.3 Decryption

3.3 Additional Features

For the ease of debugging and usability we have implemented a few additional features: UART interface,

3.4 Challenges & Revisions

3.5 System Diagram

4 Conclusions

4.1 Summary of Development

The majority of our time was initially spent researching about LWE and the various components that we would need to implement in order to successfully encrypt and decrypt a simple message. This initial step ended up taking much more time that we expected to complexity of the topics that was required to understand how and why the algorithm works the way it does. Once we had felt we had a good understanding of the topic we began coding an implementation in Python using the NumPy library which included many prewritten functions. This allowed us to confirm that our understanding of LWE was correct. With our python implementation in hand we begun to make an initial system design and determine what we components we would have to develop. After a bit of discussion, we decided that we would design all these components our self, this was a task that we again under estimated. Our reasoning for this was that we wanted to make our project's completely open and flexible to future changes. Currently we have our initial system architecture completed as well as the various components. We have begun assembling the components and have done some rudimentary testing.

4.2 Remaining Work & Future Development

The tasks that remain are mainly debugging and cleaning up of our implementation. With the various changes that we had to implement throughout the duration of our project we anticipate various bugs when all the components are put together. Once the system is assembled and base functionality is tested, we expect that the next step will be performing an analysis on the devices characteristics such as power consumption, speed of computations, etc. Finally, with that information in hand the system can be optimized and improved.

5 LaTeXref

$$1 + 2 = 3$$

$$1 = 3 - 2$$

$$\begin{aligned} 1 + 2 &= 3 \\ 1 &= 3 - 2 \end{aligned}$$

$$\begin{aligned} f(x) &= x^2 \\ g(x) &= \frac{1}{x} \\ F(x) &= \int_b^a \frac{1}{3}x^3 \end{aligned}$$

$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$

$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$$\left(\frac{1}{\sqrt{x}}\right)$$

Random citation [1] embedded in text. Other citations, here [**?**] and here [2]

References

[1] J. Doe, *The Book without Title.* Dummy Publisher, 2100.

[2] N. Göttert, T. Feller, M. Schneider, J. Buchmann, and S. Huss, “On the design of hardware building blocks for modern lattice-based encryption schemes,” in *Proceedings of the 14th International Conference on Cryptographic Hardware and Embedded Systems*, CHES’12, (Berlin, Heidelberg), pp. 512–529, Springer-Verlag, 2012.