Lab 08: Post-Quantum Cryptography (PQC) Security

Jasmin Kaur and Alec Braynen
University of South Florida
Tampa, FL 33620

I. Introduction

As quantum computing continues to grow more feasible, the breaking of the security of public-kye cryptographic algorithms grows along with it. In other words, quantum computing advancements will break some current mainstream cryptographic algorithms. Post-quantum cryptography (PQC), in response, is the field and science concerned with cryptographic security and algorithms in a post-quantum computing world. PQC assumes that quantum computers will be able to solve cryptographic algorithms, such as RSA and ECC, in polynomial time.

In this work, the authors delve into PQC, comparing a classic (ECC) and a PQC Kyber algorithm in terms of their performance and implementation. Additionally, we simulate Shor's quantum algorithm and present some results. The Shor simulation ran successfully and generated the factors for our given input. Additionally, the PQC Kyber algorithm has some positive traits, especially when compared to our ECC algorithm.

II. Reading Check

Question 1: Why do we need PQC? Why do we need to start looking for PQC now rather than later?

Answer: We need PQC to protect the data, domains, and communications currently secured by public key cryptographic algorithms. These assets will no longer be secure once quantum computing exists, since quantum computers will be able to break/solve the mathematics that protects these algorithms in polynomial time. We cannot wait until quantum computers are capable of this, because the consequences of current cryptographic algorithms breaking because of quantum computers would be tragic. [1].

Question 2: What's the difference between Key Encapsulation Mechanism (KEM) and Digital Signatures?

Answer: According to [1], key encapsulation methods are used for exchanging symmetric keys in symmetric key cryptography, while digital signatures are used to verifiably sign a document or message. In other words, the key encapsulation mechanism concerns cryptographically securing digital keys, while digital signatures concern verifiably signing a message or document that can be authenticated as signed by a particular signer.

Question 3: What is difference between Grover's and Shor's quantum algorithms?

Answer: Grover's quantum algorithm attacks symmetric key cryptographies such as AES. However, it does not fundamentally break the AES algorithm since we can change the parameters of it for continued useful protection against the quantum algorithm. Shor's quantum algorithm on the other hand, fundamentally breaks the public key, private key cryptographic algorithms such as RSA, that depend on the (as of now) hard computation problem of factorization. Shor's quantum algorithm moves the factorization problem into the polynomial time computation category (within quantum computing.) [1]

III. Methods

Our method is as follows: 1) we simulated a post quantum implementation of Shor's algorithm, 2) implemented an ECC algorith m in Xilinx, and 3) compared the total number of slices, LUTs and slice registers with the report [2]. These steps are explained in more detail below.

A. Software Setup

The software setup for this experiment consisted of a Windows 10 OS host machine and Xilinx's Vivado 2022 installed on this machine.

B. Experimental Procedure

Part A: Simulating Shor's Algorithm

Harvesting random bits from the FPGA:

- We utilized the Shor algorithm simulator: https://blendmaster.github.io/ShorJS/
- We set the number to factor: [n]
- We chose the number for the base: [a]
- We clicked "continue" to enter the "quantum part" of the simulation
- This initialized the Initial and Quantum Register graphs. We clicked "Measure"
- This initialized the Entangled and Fourier Transformed Register graphs. We clicked "Measure Register."
- This led to the "Classical Post Processing Step" where we got the output
- We repeated these steps with different values.

Part B: Classic vs PQC Algorithms on Hardware

Using Xilinx's Vivado we:

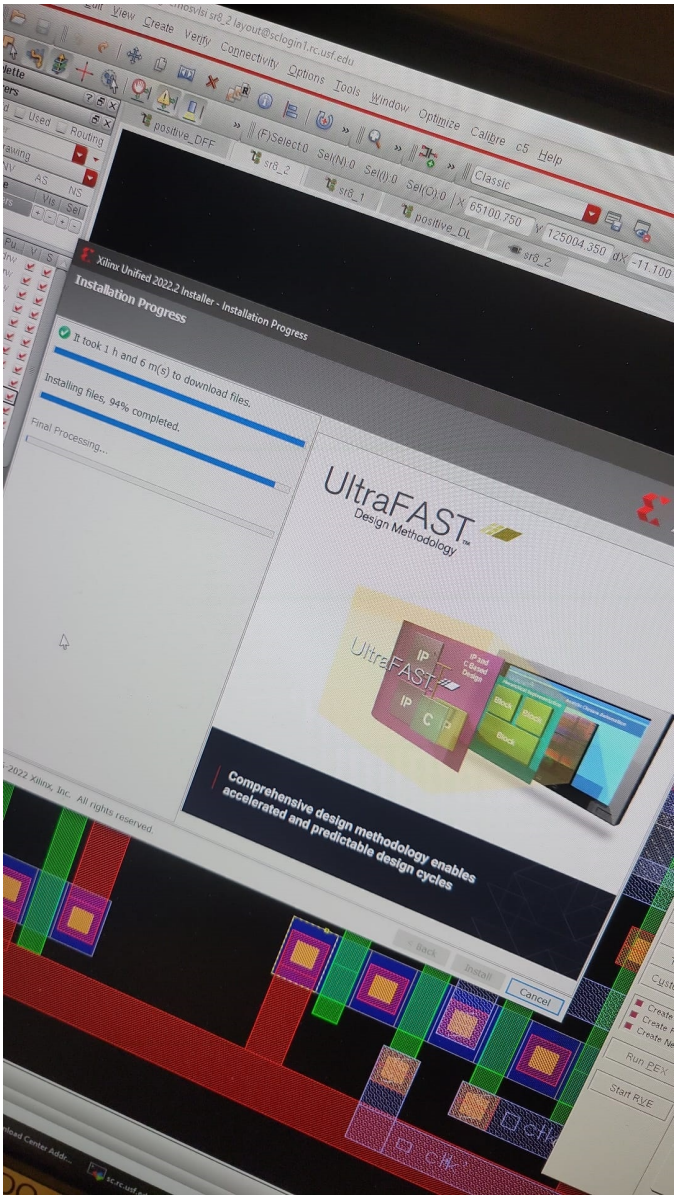

Fig. 1: Windows PC running Vivado

- Synthesized ECC-HDL RTL file
- Implemented the synthesized file on FPGA Part "XC7A200TFFG1156-1".
- Navigated to the "Map Report" section in Vivado
- Utilized the data in this section as a referent for comparison with the work [2].

IV. Results

The results of our work are shown in Figure 2, 3, 4 and Table 1. Figure 2, 3, and 4 show that the Shor simulation was able to calculate the factors for an **n=15** and the a value chosen was **a=7**. While this is a trivial example for classical computing, this method, when successfully applied to larger numbers, can break or solve (depending on the scientific field) public key cryptography or the factorization problem.

Table 1 compares our ECC hardware implementation with that of the post quantum hardware design called CRYSTALS-KYBER. As shown in the table, CRYSTALS-KYBER implementation is much less demanding of memory hardware than our implementation. Additionally, CRYSTALS-KYBER is a post-quantum secure algorithm. This makes it patently future proof regarding some quantum algorithms whereas our ECC implementation has no quantum protections.

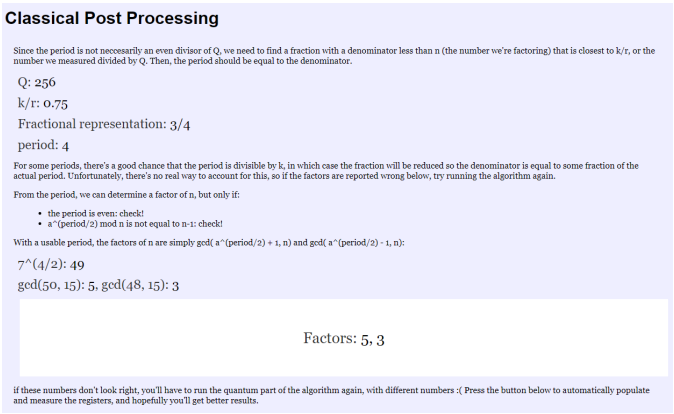

Fig. 2: Shor Algorithm Simulation Results

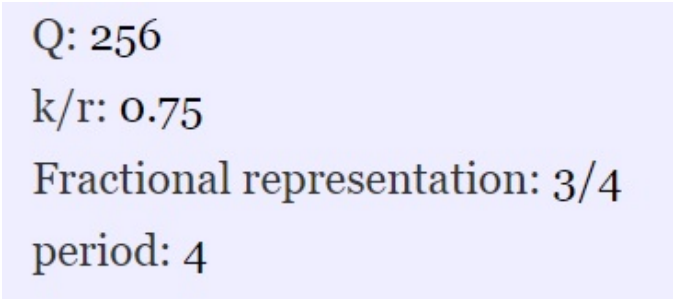

Fig. 3: Classical Post Processing Variables

Fig. 4: Shor Algorithm Results

TABLE I: Comparison of LUTs, Slices and Slice Registers of ECC and PQC algorithm

Implementation	Slices	LUTs	Slice Registers
ECC	7621	25370	7048
CRYSTALS-KYBER	2121	7412	4644

V. Discussion

Shor's algorithm simulator shows a potential implementation of quantum computing algorithm. The classical part of the algorithm transforms the problem from a factoring problem, to one of find the period of $a^x mod * n$. It does this by utilizing the Euclidean algorithm. If this algorithm returns a greatest common divisor that is not 1, the computation is trivial; however, if it returns a greatest common divisor of 1, then the period must be calculating using quantum computing.

The quantum part of the algorithm computes the period of the function created above. Technically, it utilizes quantum registers big enough to hold Q numbers such that $n^2 <= Q <= 2n^2$. In other words, it utilizes a "Q"-qubit wide quantum register. Superposition allows for every single calculation to find the period $a^x modn$ to occur simultaneously. The result of these calculations are stored in another quantum register. Quantum entanglement and the measurement of the second then first register allows for only values that are constrained to the measure of the second register to be returned from the first register. A quantum fourier transform is then applied to the first (entangled) register. This returns the value k*Q/r which is then post-processed via classical computing.

Notably, the Shor algorithm simulation can fail due to its probabilistic modeling of its hypothesized quantum computer. In these scenarios, a re-computation must occur to obtain the correct answer. Additionally, for trivial factorizations, the quantum simulation does not occur due to the answer being computable in classical computing.

Our results show that there is much work to be done in the field of PQC. The Shor algorithm demonstrates the theoretical feasibility of quantum algorithms while our comparison of our hardware implementation to that of KYBER shows the work being done in the field to protect devices from the coming threats of quantum computing to security.

The experiments went as expected and besides having to upgrade Vivado on our testing machine, there weren't any challenges. In order to build on this work, next time we should research more works with which to compare our results and implementation parameters.

VI. Conclusion

In this lab, we ventured slightly into the field of Post Quantum Cryptography. We simulated Shor's post quantum algorithm and demonstrated the theoretical possibility of quantum computers solving factorization for large numbers; this would lead to the breaking of cryptographic systems depending on this mathematical security. Additionally, we compared a traditional cryptographic system (ECC) with that of a post quantum system (KYBER) and compared the implementations. The PQC algorithm seems more efficient than our ECC implementation and additional has been designed with quantum security in mind. We ultimately learned through both research and our work here, that PQC is a vibrantly novel field for further research and investigations. In regards to future work, we would like to explore other post quantum algorithms and implementations and see how they can be improved.

References

[1] R. Karam, S. Katkoori, and M. Mozaffari-Kermani, "Experiment 8: Post-Quantum Cryptography (PQC) Security," in *Practical Hardware Security Course Manual*. University of South Florida, Aug 2022.

[2] Y. Xing and S. Li, "A compact hardware implementation of cca-secure key exchange mechanism crystals-kyber on fpga," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 328–356, 2021.