Post-Quantum Cryptography (PQC) Security Investigating PQC Hardware Security and Performance

Ganesh Veluru and Jwala Sri Hari Badam
University of South Florida
Tampa, FL 33620

I. Introduction

In the ever-changing cybersecurity landscape, quantum computing appears as a disruptive force ready to reshape the very fabric of cryptographic systems. The substantial influence on classic encryption techniques like as RSA and ECC is at the center of this revolution. These stalwarts, which have long been the cornerstone of secure communication, are under assault from quantum computers, which promise to unravel their security through the rapid resolution of mathematical difficulties. This laboratory experiment sets out to uncover the complexities of this quantum shift, with a special emphasis on the fearsome Shor's Quantum Algorithm.

This research is centered around two investigations, each of which sheds light on a different aspect of the field of quantum cryptography. The first section covers a thorough analysis of Shor's Quantum Algorithm, a ground-breaking invention that makes it possible to factor enormous numbers efficiently and directly challenges the security of traditional cryptography. In the second phase, the investigation is expanded to hardware implementations, and a comparative study between the post-quantum secure Kyber algorithm and the classical ECC method is conducted on the AMD Xilinx Vivado platform. In the context of actual embedded systems, this comparison study seeks to elucidate the usefulness and efficacy of post-quantum cryptography methods [1].

The vulnerability of RSA and ECC, which captures the urgent need to investigate post-quantum secure alternatives, is at the heart of our work. In this story, Shor's Quantum Algorithm plays a crucial role in demonstrating the potential of quantum computers to overcome the computational challenges that have long protected traditional cryptography systems. We explore the experimental terrain, moving between the classical and quantum domains, solving equations and deciphering the complex interplay between cryptographic security and quantum computing.

II. Reading Check

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

Answer: Post-Quantum Cryptography (PQC) is essential because quantum computers have the capacity to fundamentally disrupt the security of traditional cryptographic methods like RSA and ECC. When using algorithms like Shor's, quantum computers may answer complicated mathematical problems tenfold quicker than traditional computers. Shor's method poses a danger to the fundamental security features of popular public-key encryption schemes, making the protocols vulnerable to quick factorization and discrete logarithm computations. Thus, the imminence of investigating and implementing PQC stems from the impending danger that quantum developments bring to the established security guarantees offered by classical cryptography techniques [2].

The potential lead time needed for the creation and broad adoption of reliable cryptographic alternatives emphasizes the need to start the search for PQC solutions as soon as possible. Classical cryptography algorithms' window of vulnerability is getting smaller as quantum computing technology develops. The current state of quantum research and development indicates that the development of powerful quantum computers may not be a far-off possibility in the future. The cybersecurity community wants to remain ahead of the curve and make sure that strong cryptographic procedures are in place before quantum dangers manifest themselves, thus they are actively searching for PQC solutions now. Postponing PQC research could lead to a major gap whereby quickly developing quantum capabilities could abuse current cryptographic infrastructures. Therefore, it is strategic important to strengthen our digital security foundations against the impending quantum age by pursuing PQC in a timely manner. This is not merely a preventive measure.

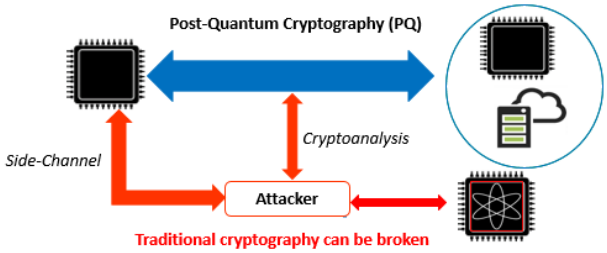


Fig. 1: Post Quantum Cryptography

Question 2: What's the difference between Key Encapsulation Mechanism (KEM) and Digital Signatures? (we use these terms in both classic and PQC)

Answer:

Digital signatures and the Key Encapsulation Mechanism (KEM) are crucial components of digital communications security in both conventional and Post-Quantum Cryptography (PQC) frameworks. In order to create shared secret keys over possibly insecure channels, Key Encapsulation Mechanism is essential for ensuring safe key exchanges between participants. This is achieved in the classical domain by protocols like Diffie-Hellman, and in the PQC domain by algorithms like Kyber, which use structured lattices to facilitate quantum-resistant key exchange. Conversely, digital signatures are essential for verifying the integrity and source of digital messages. Digital signatures in classical cryptography are often generated using algorithms like RSA and ECC, but secure alternatives are provided by PQC standards including CRYSTALS-Dilithium, Falcon, and SPHINCS+. Both KEM and Digital Signatures are fundamental components ensuring confidentiality, authenticity, and integrity in cryptographic protocols.

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

Answer: The quantum algorithms developed by Grover and Shor tackle different problems in the field of quantum computing. Grover's approach offers a quadratic speedup over conventional algorithms and is mainly focused on the problem of searching an unsorted database. Grover's approach has significance for symmetric-key cryptography since it may be used to do a brute-force search for a secret key in approximately the square root of the classical time. Accordingly, in the post-quantum age, the security of symmetric-key algorithms can be virtually halved, and increasing key lengths is advised to preserve equal security [3].

Conversely, Shor's quantum algorithm is a revolutionary development with significant ramifications for public-key cryptography. Specifically, Shor's approach addresses discrete logarithm and integer factorization difficulties. Shor's technique represents a serious danger to both RSA and ECC, which rely on the difficulty of factoring big numbers and solving discrete logarithm issues, respectively. Due to its ability to calculate huge composite numbers efficiently and solve discrete logarithm problems in polynomial time, these traditional encryption systems are susceptible to quantum assaults. The diverse applications and impacts of Grover's and Shor's quantum algorithms on classical cryptography approaches are highlighted by this basic difference in focus.

III. Methods

A. Hardware and Software Setup

We will compare a conventional and a Post-Quantum Cryptography (PQC) method using AMD Xilinx Vivado v2021.1-2022.2, focusing on their implementation metrics and performance. We will do our study using the given Register-Transfer Level (RTL) specifications of the PQC algorithm Kyber and the Elliptic Curve Cryptography (ECC). The goal of this investigation is to show how useful PQC algorithms are for embedded designs, emphasizing the continuous advancements in both software and hardware implementations. We will also use an online link to test Shor's quantum algorithm, offering a thorough analysis of both classical and quantum cryptography techniques. This evaluation doesn't require any further instruments. Obtaining the Register-Transfer Level (RTL) or Hardware Description Language (HDL) files for Elliptic Curve Cryptography (ECC) is necessary in order to continue with Part B of the lab. The PHS-Lab-08 directory has the files we need to download because it has all of the implementation's necessary components.

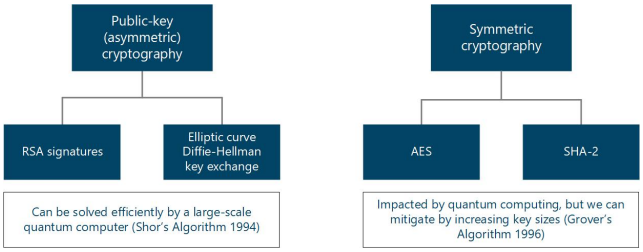
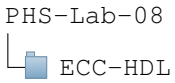


Fig. 2: Grover's and Shor's Algorithm

The directory is structured as follows:



B. Experimental Procedure

Part A: 8A: Shor's Quantum Algorithm

In terms of its implications for classical cryptography, Peter Shor's algorithm, a quantum method he developed in 1994, is a ground-breaking advancement in the realm of quantum computing. The technique tackles two basic number theory issues: discrete logarithms and integer factorization, which are the foundations of the security of popular cryptographic systems like RSA and several elliptic curve schemes. The two primary stages of Shor's algorithm's operation are classical and quantum.

A random number a that is less than the goal number n to be factored is chosen at random during the classical phase. The technique looks for a co-prime pair (n, a), which denotes that n and a have 1 as their greatest common divisor. This classical preprocessing prepares the system for the quantum phase that follows.

Shor's algorithm outperforms classical algorithms exponentially in the quantum phase. It effectively performs Fourier transformations and modular exponentiation on a superposition of states by utilizing quantum parallelism. This allows the procedure to determine a function's period associated with n factors. The last stage is to extract the desired elements from the quantum computing using traditional post-processing [4].

Given the alleged complexity of these mathematical puzzles, Shor's technique is significant because it can factor big numbers tenfold quicker than the most well-known classical

algorithms. This poses a serious danger to the security of cryptographic systems. Post-quantum cryptography solutions are becoming more and more necessary as quantum computing technology develops in order to maintain the robustness of secure communication and data protection. . The algorithm can be simply restated and shown in Fig **??**.

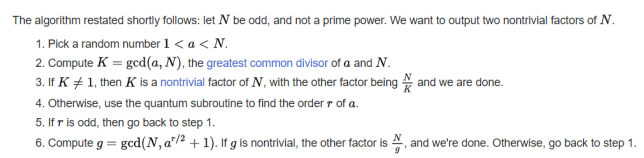


Fig. 3: Shor's Algorithm

Step 1: using the shor's simulator we are choosing the N=15

Step 2: Now on clicking the Factor button it will ask to choose the 'a' value and we can randomize them.

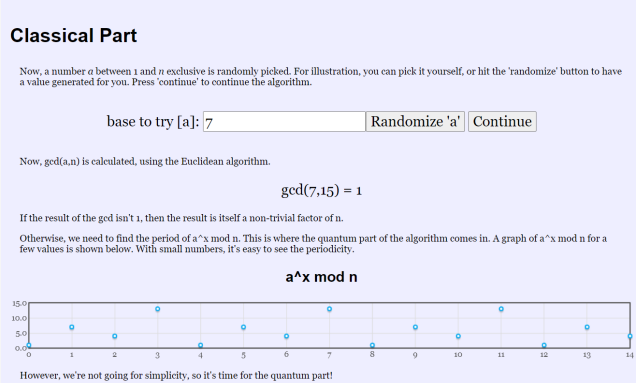


Fig. 4: Classical Part

Step 3: GCD(a,N) should be equal to 1, otherwise we should repeat the process with new 'a' value.

Step 4: After calculating we will get Transformed Quantum Register and Fourier Transformed register.

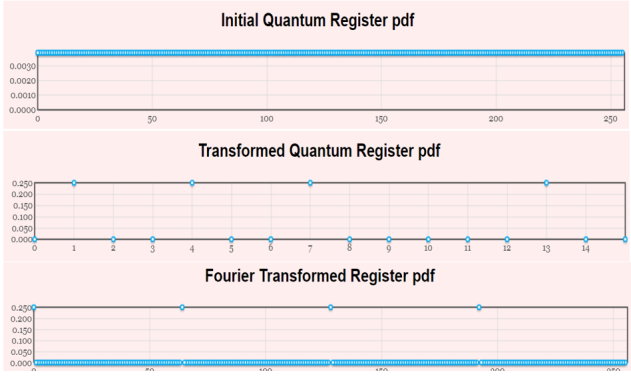


Fig. 5: Different Registers in Quantum Part

Step 5: if the value of 'r' is odd then we need to repeat the process with new values. At the end we will get

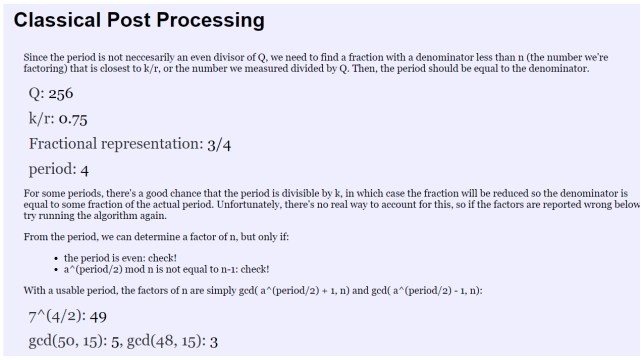


Fig. 6: Classical Post Processing

period value which is used to find the actual factors- factors of n are simply gcd(a^(period/2) + 1, n) and gcd(a^(period/2) - 1, n)

Step 6: Repeat this for other values of n.

Part B: 8B: Classic vs. PQC Algorithms on Hardware

A key element of contemporary cryptographic systems, elliptic curve cryptography (ECC) offers a strong basis for secure communications. The basic equation for ECC is $y^2 = x^3 + ax + b$. It functions primarily on elliptic curves formed over finite fields.

ECC is elegant because it can carry out cryptographic operations by multiplying points on the elliptic curve scalarly. A point P is multiplied by a scalar k in this scalar multiplication to get a resultant point $Q = k \cdot P$. The safety of ECC depends on how difficult the discrete logarithm problem is in elliptic curve scenarios, where it is computationally impossible to find the scalar k from the points Q and P.

The complexity of the discrete logarithm problem based on elliptic curves provides the foundation for the security strength of ECC. This issue, expressed in terms of cryptography, is to discover the scalar k given two points Q and P so that $Q = k \cdot P$. Since the inherent difficulty of this problem's solution adds to ECC's defense against attacks, public-key cryptography favors it. Notably, compared to more conventional cryptosystems like RSA, ECC offers a strong efficiency advantage by offering strong security with reduced key lengths. This efficiency makes ECC a key component in the cryptographic landscape for guaranteeing both security and computational efficiency, especially in situations when resources are limited.

we need to compare the metrics of Kyber512 with the provided RTL and steps is as follows to collect the metrics:

Step 1: We must take the following actions in order to implement the given RTL (Register Transfer Level) design in the AMD Xilinx Vivado v2021.1-2022.2 tool: Synthesize and implement the top-level Verilog object first, then all related entities.

Step 2: To ensure compatibility, select the FPGA part "XC7A200TFFG1156-1" from the Artix-7 FPGA family.

TABLE I: Results of Shor's Algorithm Simulator

N	a	Q	k/r	k*Q/r	period	factors
15	7	256	0.75	192	4	5,3
30	13	1024	0.25	256	4	6,10
21	8	512	0.5	256	2	3,7
18	7	512	0.666	341	3	-
18	5	512	0.167	86	6	-
18	3	-	-	-	-	3

TABLE II: Comparison between ECC and Kyber512

Parameter	ECC	Kyber
LUT	25370	7412
FF/Slice Register	7048	4644
slice	7621	2126

Step 3: After the synthesis and implementation are finished, we need to open the AMD Xilinx Vivado tool and go to the Map Report area. Important metrics must be noted in this report in order to evaluate the effectiveness and resource management of the executed design

Step 4: In particular we are concentrating on, count the total number of "LUTs" (both "LUTs as Memory" and "LUTs as Logic"), "Slices," and "Slice Registers."

Step 5: These metrics describe how the FPGA is used, and where the registers, lookup tables, and logical elements are distributed in the design.

Step 6: we will get tree-like RTL representation of the implemented design, since we also need to make sure that the tool under the new project is correctly reading all source files. An extensive comprehension of the hardware resources used by the ECC (Elliptic Curve Cryptography) architecture on the designated FPGA platform is made possible by this thorough evaluation.

IV. Results

The Shor's Quantum algorithm is repeated for other values of N and the results are listed in Table I.

The comparison between ECC RTL values and Kyber512 values is represented in Table II.

The implementation output of ECC in vivado is shown in the below Fig 7.

V. Discussion

Shor's algorithm's "Classical Part" consists of classical calculations that set up the prerequisites for the quantum phase. This step involves selecting a random integer, a, and computing the GCD of 'a' and the number that has to be factored, n. In order to ensure that a and n are coprime, the classical part seeks to find a value of a such that GCD(a, n) = 1. In the 6th case from Table I the GCD(a,n)!=1 and the process stopped at classic part. After finding a suitable a, the method moves on to the quantum phase, where the factors of n are revealed by the quantum computer's efficient performance of modular exponentiation and Quantum Fourier Transform. The precise choice of a in the classical portion is essential to the quantum factorization's success.

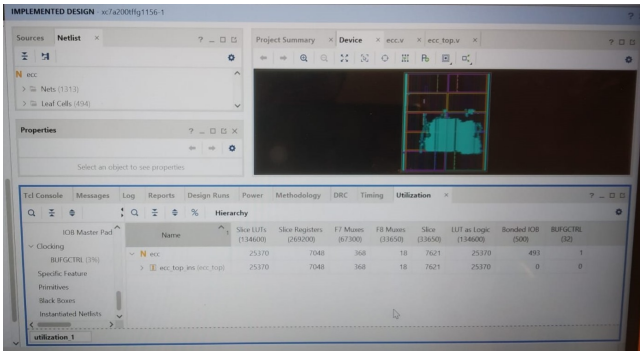


Fig. 7: Implementation Output

Shor's algorithm's "Quantum Part" is essential to using quantum computing principles to factor big composite numbers quickly and effectively, which defeats conventional public-key cryptography procedures. This part of the program uses entanglement and superposition, two quantum features, to enable parallelized computations on a quantum computer. In particular, it uses modular exponentiation and the Quantum Fourier Transform (QFT) on a quantum computer to quickly identify the periodicity present in the modular exponentiation function. Shor's approach successfully converts the factorization issue into a system of equations that can be solved quickly by classical computers by removing this period [5].

In the Second case for n=30 the shor's algorithm simulator doesn't gave accurate result and in 4th case the period=3 which is odd number so we are unable to find the factors and in the 5th case a$\hat{}$(period/2) mod n is equal to n-1 where again we failed to find the factors of 'n'.

The Look-Up Table (LUT), slice registers, and slice values for the Kyber and Elliptic Curve Cryptography (ECC) algorithms offer information about the effectiveness and resource needs of these cryptographic techniques. The comparatively larger values of slices (7621), slice registers (7048), and LUT (25370) in the context of ECC point to a higher resource consumption. These metrics provide an indication of the computational overhead and complexity related to ECC implementations. Because of the high number of LUTs and slices, ECC may demand more hardware resources, which could result in reduced efficiency in terms of space and power usage.

On the other hand, the Kyber algorithm uses less resources, as seen by the LUT values of 7412, slice register values of 4644, and slice counts of 2126. These significantly lower figures imply that Kyber is more resource-efficient and requires less hardware to operate. In a variety of applications, especially those with limited resources, this efficiency might be vital.

Furthermore, as we pointed out, Kyber's quantum resistance is a big benefit over ECC. Kyber's resilience to quantum assaults makes it a more safe option for post-quantum cryptography applications, whereas ECC is susceptible to them because it depends on the complexity of specific mathematical problems.

In conclusion, Kyber seems to be a more secure and effective option than ECC based on the values and taking the quantum resistance factor into account. Kyber's reduced resource needs make it appropriate for a wider variety of applications, particularly in situations where resource limitations are an issue.

VI. Conclusion

To sum up, this lab experiment shed light on the field of post-quantum cryptography (PQC) and the impending difficulties brought about by the development of quantum computing. The investigation into Shor's quantum algorithm highlighted how susceptible public-key cryptography schemes like RSA and ECC are to quantum computers that can factor data in polynomial time. Seeing the method in action—even in a virtual setting—emphasized how urgently quantum-resistant encryption techniques need to be developed and implemented.

Additionally, the viability of implementing PQC algorithms in embedded architectures was clarified by comparing the performance of conventional and PQC algorithms on AMD Xilinx Vivado hardware. The examination of the PQC algorithm Kyber and the RTL of ECC showed how hardware and software implementations are changing, with PQC algorithms proving practical in real-world applications. In the long run, this experiment encourages more research into the creation and harmonization of post-quantum cryptography systems. It is imperative that we proactively incorporate quantum-resistant algorithms into our security frameworks as quantum computers develop. Subsequent research endeavours may encompass a more comprehensive examination of the performance measures of diverse post-quantum cryptography (PQC) algorithms, their potential for adaptation to diverse computer environments, and the current standardisation initiatives in the field. This lab is a first step toward a more comprehensive grasp of how the field of cryptography is changing in response to quantum developments.

References

[1] R. Karam, S. Katkoori, and M. Mozaffari-Kermani, "Experiment 8: Post-Quantum Cryptography (PQC) Security/Introduction," in *Practical Hardware Security Course Manual*. University of South Florida, Nov 2023.

[2] ——, "Experiment 8: Post-Quantum Cryptography (PQC) Security/Background," in *Practical Hardware Security Course Manual*. University of South Florida, Nov 2023.

[3] ——, "Experiment 8: Post-Quantum Cryptography (PQC) Security/Background-1," in *Practical Hardware Security Course Manual*. University of South Florida, Nov 2023.

[4] ——, "Experiment 8: Post-Quantum Cryptography (PQC) Security/Lab Assignment," in *Practical Hardware Security Course Manual*. University of South Florida, Nov 2023.

[5] ——, "Experiment 8: Post-Quantum Cryptography (PQC) Security/Lab Assignment-1," in *Practical Hardware Security Course Manual*. University of South Florida, Nov 2023.