An Examination of Post Quantum Cryptography and Cryptographic Algorithms

Sadman Shafiq*
School of Computer Science and Engineering
University of New South Wales, Sydney, NSW
(Dated: April 13, 2021)

This paper aims to review recent findings of NIST's Round 3 Post Quantum Cryptography. In particular, looking at different methodologies of ensuring security in a post quantum world such as Code-Based McEliece and Lattice Based CRYSTALS-KYBER. As well as defining it against current methodologies of Symmetric and Asymmetric Algorithms.

I. INTRODUCTION

Cryptography is aimed at solving key issues with communication. The development of secure channels is essential because of potential eavesdroppers, who may be listening in on the data or modifying the data. Therefore, communication needs to be secured into three distinct categories: *Confidentiality*: The content of the message cannot be inferred. *Integrity*: The content cannot be modified without being noticed. *Authenticity*: The receiver has the proof that the message came from the intended sender. Current use cases of cryptography have two system types: Symmetric Key Encryption such as those available in AES-256, HMAC-SHA-256, SHA-3 [1]. As well as Asymmetric Keys such as RSA, DSA, ECDH. Symmetric Keys are used when the sender and receiver (commonly known as Alice and Bob) have met already and have a shared secret. Whereas Public (Asymmetric) Key Encryption exists if Alice and Bob have not yet met and need to communicate. Classically public key encryption has relied on number theory problems such as factorization or discrete logarithmic problems.

For the most part these encryption techniques can resist even the fastest supercomputers with exponential time required to solve these mathematical problems. However, in 1994 Peter Shor devised an algorithm that will work on a Quantum Computer to break our current encryption method of using factorization and discrete logarithms, requiring theoretically N^3 time to solve [2]. This is only a threat to current public encryption when Quantum Computers exist. For symmetric key encryption, if Alice and Bob already have a long secret key they can use the 'One time Pad' for confidentiality and HMAC for integrity and authenticity. AES-256 is a short secret key which can create a long key using 256 bits. In 1996 Lov Grover used Quantum Mechanics to solve unstructured search, where a large database needed to be matched with another database with unique links, this would take roughly $N/2$ operations to complete, while Grover's method of using Quantum Superposition reduced the operations to $\sqrt{N}$ This is secure from Quantum Computers for the most part as Grover's Algorithm will take 2^{128} operations to compute. Essentially if AES-128 is used then the time factor is much smaller and can cause problems for secure communication.

From the inception of Shor's algorithm till today there has been significant advances in the power of Quantum Computing, where the informational units used by computers is no longer binary, but a superposition of both states. This is called a qubit, due to it's quantum properties it can also leverage quantum mechanics such as entanglement and generating true randomness from a deterministic machine. From the first experimental 2-Qubit quantum Computer built in 1998 till the current Google's Bristlecone with 72-Qubits, a number of issues still exists with the quality of qubits as there are problems with interference and error correction, but in the same vein as Moore's Law there is a large possibility of Quantum Computing being available over the coming years. This poses a security threat not just to future communication but to past communication as well, as National and state Secrets can become decrypted before their normal age of expiry. This poses a big risk to public key encryption as Quantum Computing processes develop, and hence the US National Institute of Standards and Technology(NIST) issued a standardization process for Post Quantum Cryptography Cyphers which will attempt to secure communication against attackers with Quantum Computers. It is currently on its 3rd Round of validating and developing algorithms which will be resilient against the powers of Quantum Computers. In this paper, we will first explore the current cryptography schemes.

II. CLASSICAL CRYPTOGRAPHY

We can classify the cryptography use cases into two distinct Criteria: **Key Encapsulation Mechanisms** and **Digital Signature Algorithms**. Key Encapsulations are the ones that are currently under threat from Quantum Computing while Digital Signatures are safer for the time being. Digital Signatures is nominally using symmetric key encryption because the defining characteristic of Symmetric Key Encryption is its ability to be used in high frequency environments. Symmetric keys use only one key to encrypt and decrypt which allows for faster communication and less overhead when used for other

* Also at School of Computer Science and Engineering University of New South Wales, Sydney, NSW; z5304769@ad.unsw.edu.au

purposes outside communication. The AES is a block cipher and has been through a series of rigorous tests and attempts of breakage. Even against a Quantum computer which can attempt the Grover's method it can still be secure at the 256 bit length.

A. Advanced Encryption Standard

In 1997, NIST set out for a new Encryption Standard, similar to the current Post Quantum Cryptography. The previous encryption method developed by IBM, DES, was not as fast as what was being required and there was a need to develop a new standard. During the competition cryptographers attacked the cypher and the cyphers were judged across a number of key criteria such as CPU Cycles, low memory footprint, deployable and completion time based. Rijndael performed well across a lot of devices and ended up being chosen. It is a 128 bit symmetric block cypher. The key can be 128 bits to upto 256 bit security. AES arranges every 16 bytes into 4 by 4 byte matrices. It performs a series of steps including substitution and permutation across a series of rounds where a key is added at the end of each round. Each round consists of:

1. Substitution of Bytes using a fixed table in a specific manner
2. Shift rows: left shift is applied to each of the four rows, all other entries are entered on the right. Resulting in a new matrix with same 16 bytes
3. Mix Columns: where a function is applied to each of the four byte columns and the 4 bytes in a column is processed and transmuted into 4 different bytes.
4. The resulting 128 bits is XOR with a round key of 128 bits. The Round Keys are generated using a previously agreed upon key and divided across how many layers of protection is needed (128,192,256)

The AES is used in both hardware and software security Protocols such as Wifi but is also susceptible to Side Channel Attacks. These attacks are based on weaknesses on the physical implementation such as leakages in the radio signals. This is an attack on not only the algorithm itself but also on the data that is coming out through the processes such as power consumption of the encryption process [3]

B. HMAC and ECCDH

A Hashed Message Authentication Code is a form of signature algorithm that will authenticate the user and make sure the message has not been tampered in any way. It introduces a secret key which makes sure the message that is received is from the intended sender. HMACs are intended to be used in high-performance systems such as routers and servers and are commonly found in the SSL Certificate. The algorithm for any particular HMAC is as follows:

1. Make the length of symmetric key (K) equal to several bits (b) in each block.
2. EXOR K with inner Pad equivalent to b bits producing a variable secret (S1)
3. Append S1 with the message (M)
4. Apply a hash such as SHA-256 on (S1 — M)
5. Simultaneously apply a buffer of n-bits which produces a n-bit hash: H(S1 — M)
6. EXOR K with Outer Pad equivalent to b bits producing secret (S2)
7. Append S2 with the previous hash
8. Apply hash algorithm again to get: H(S2 — H (S1 — M))

If the key itself is compromised then attackers can create unauthorized messages.

Until now we have discussed digital signatures and sending encrypted data which require a key to have already been shared by both sender and receiver. In order to produce this key we need to have a look at Key Encapsulation:

The Diffie Hellman Key Exchange Algorithm uses a Symmetric Key Exchange but in the current use case it has been combined with Elliptic Curve Cryptography to provide a more robust and safer exchange method.

Elliptic Curves require shorter key lengths which use fewer memory and CPU usage. The elliptic curve requires knowledge of a few points: The equation for Elliptic Curve is given as: $E(x, y) : y^3 = x^3 + ax + b\{O\}$ Addition of Points: in an elliptic curve once 2 points are added the line that connects the two points and intersects the curve produces the output as the inverse of the Y coordinate, as seen in Figure 1:

Secondly if two points have the same x coordinate then it is the point at infinity. Also known as the identity element. O

A one-way function is generated out of the scalar multiplication of points on the Elliptic Curve as: $E(Z/pZ)$ If we are given an elliptic curve with a point P and Q as a multiple of P, the difficulty lies in finding the value of k such that Q = kP.

The key exchange using Elliptic Curve Diffie Hellman requires a few domain parameters such as the field (mod P), a and b as curve parameters, G as the Generator Point, n as order(G) and h as the cofactor. These are shared publicly as well as a combination of the private key held by Alice and held by Bob multiplied with the Generator to get: AG for Alice and BG for Bob. When these are exchanged again, the multiplication is hard to reverse, which is secured by the same method of Diffie

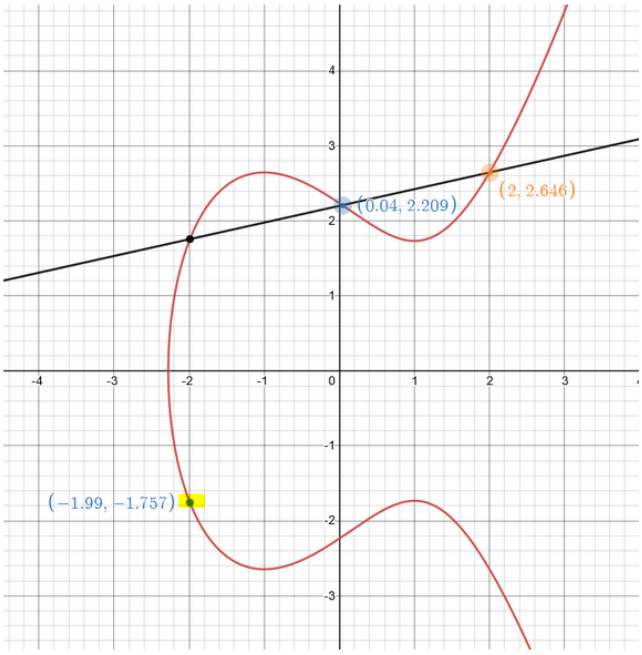

FIG. 1. Elliptic Curve Algorithm defined using arbitrary points

`https://www.desmos.com/calculator/ialhd71we3`

Hellman and can only be cracked by brute force. The Shared secret is now a combination of both and which is used to find a point in the curve and used for further secure communication

III. QUANTUM CRYPTOGRAPHY

The process of quantum cryptography is exciting as well as worrying. The main form of cryptography that is currently being explored is the Quantum Key Distribution Process as well as other lesser known ones such as Bit Commitment and an example of a Quantum attack is by using Shor's Algorithm. When we discuss Quantum computing we also need to define and the use of Qubits. These Qubits use Quantum state ψ which exists within a finite dimensional complex vector space also known as Hilbert Space. A Qubit exists within a two dimensional Hilbert Space with a basis of 0 and 1. The main difference being a Qubit and a normal bit is that it can be defined coherently in a superposition of both bases. The states of a Qubit can be used to project information and hold them. One of the main properties of Qubits is the No-Cloning Theorem. Which states that we can only copy orthogonal or identical states. [4]

A. Quantum Enforced Security

Quantum Key Distribution is reliant on being able to send and receive signals which encapsulate the quantum superposition of a photon or a material that can store the quantum state. This channel is used to send Qubits in the form mentioned in the BB84 Protocol. They use four quantum states in two conjugate bases. (+ linear basis, and x diagonal basis) The process is as follows:

1. Alice prepares $2n$ qubits, each in one of the four states and sends them to Bob on the Quantum channel

2. For each qubit Bob receives, he chooses at Random one of the two bases (+ or x) and measures the qubit with respect to the basis. In the case of no noise and on the same basis chosen as Alice their measurement will be the same as Alice's bits. If the basis are different, Bob's results are completely random

3. Alice tells Bob the basis used for each qubit. The bits where Bob has the same basis as Alice, Bob will keep those bits, and they will have approximately n bits left. This creates the sifted key.

4. Alice and Bob choose a subset of the sifted key to estimate the error rate. If the error rate is too high the protocol is aborted, and they do so by Bob sharing the bit values of the subset and not sharing his basis for those bits.

5. They create a joint secret key from the remaining bits that do match.

The secret which is created is protected against the eavesdropper due to the eavesdropper not being able to determine what was the original qubits that were sent and the basis used by Bob to generate the sifted key. There are also additional measures Alice can take to make sure the protocol is not compromised such as transmitting qubits already in the defined state and not in a superposition, since when the qubit is measured it is supposed to collapse to either 0 or 1 in a true random manner. By doing this she makes sure that these qubits can be used as check qubits to check on the security of the protocol.

In essence, a one time pad allows us to communicate over an insecure channel and its security doesn't depend on any computational assumption. The only problem with this type of cipher is that the parties need to preshare a secret key of the same length of the message before the communication. In classical settings one can distribute keys using a public key cryptography system. However, the security of all the existing public key cipher systems are based on the computational assumption. This implies that with this approach the security of the entire protocol would be based on the computational assumption and there is no need to use a one time pad.

Quantum Key Distribution allows us to share this classical secret key via insecure authenticated channel and its

security doesn't depend on any computational assumption. In this sense, the Quantum Key Distribution can turn the one time pad into a practical reality.

B. Post Quantum Candidates

In this section we will review the current most-promising candidates for PQC and attempt to generalize the key components of each algorithm. These have been categorized into a non-exhaustive list and only two of them are explored. There are a few other PQC Algorithms which are not covered such as the Isogeny-Based SIKE, built on some of the concepts from Elliptic Curve Cryptography, that is aimed at public key exchange,as well as the hash-based and Multivariate-based algorithms which are aimed towards digital signatures.

The current stage in NIST's competition is further development of the software and hardware capabilities of the algorithms and preparations against other possible attacks such as Side Channel Attacks like Timing and Differential Attacks, as well as making the key sizes as small as possible.

Code-Based

We use Error Detecting Code such as ones commonly used for communication between Rockets and Ground Station. The errors are generated from radio interference and radiation which can affect the signals that are received. We can find the errors by using two different methods: Parity bits and Three-times code. The Three-times code is not as efficient as it uses a significant amount of bandwidth. We can use Linear error-correcting code instead of parity in order to produce a much more reliable error correction method. In this process the message is multiplied with a matrix to create a longer bit message, which includes an overhead of a couple of bits. This overhead allows us the ability to correct errors. In general an overhead of n bits can correct $n/2$ errors. In figure 2 below we can expect errors in the communication and use our error correction algorithm to produce the correct message, after which we can inverse the matrix multiplication to find the message.

This is a slow and inefficient process with a long sequence of bits, but if we choose a good matrix this process is computed much quicker. We can also estimate how many errors we can compute with our good matrix. From this insight, we can also turn a good matrix into a bad one using blend matrices, and the bad matrix can be used as a public key while the blend matrices and good matrices together can be used as the private key [5]. The way it works is:

1. Alice generates a good matrix G as well as 2 blend matrices A and B in the form AGB, where the two blend matrices surround the good matrix. This is the private key Alice uses.

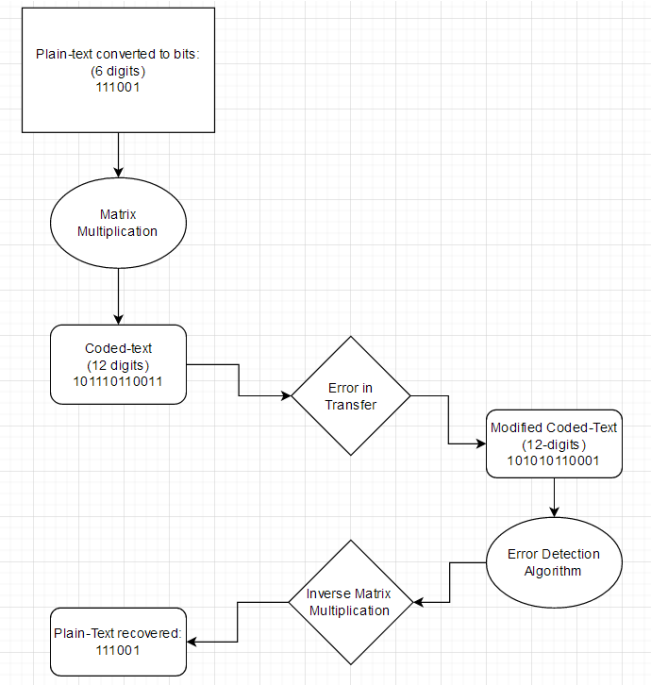

FIG. 2. Linear Error Code Correction Process

2. The product of the blend and good matrices produce a less efficient matrix which is then used as Alice's public key.

3. Bob encodes his message using Alice's public key and after the message is encoded, he generates errors at random onto the ciphertext by the process of bit flips.

4. Alice receives the message and can correct the errors very efficiently using her good matrix.

The Current Round 3 Finalist is the Classical McEliece Algorithm which combines binary Goppa codes from the original McEliece paper with an upscale of security parameters [6]. It has the highest level of assurance and has an exceedingly fast computation time but this is balanced by its larger than most public key sizes. In relation to Side Channel Attacks, the several timing based attacks require access to a lot of the variables which are reasonably difficult to acquire. There are still vulnerabilities in embedded applications as well as the Electro-magnetic field side channels exploitations [7]. These exploits either have to be fixed or the submission is removed from the competition, and as such this is still an ongoing process of refining the algorithm and it's relevant processes.

Lattice-Based

The Lattice is created using parallel, equidistant points in space. To define points in the Lattice Fields we need to have good base vectors, where the vectors are close to orthogonal to each other, while a bad base vector is one that is close to parallel to each other. When we place

a random point in this lattice field, we need to find the closest lattice point to this random point. In a 2D space this is not difficult but this space can extend out of 2D to 64, 128 or 256 dimensions. Thus when we have a good base we can find the lattice point easily, but if we have a bad base we are not able to figure out the point. We can communicate with the points themselves and send across a bad base as the public key and the good base is used as a private key, but this has been proven to be broken, so we need a better methodology. We can expand our Lattice based encryptions to include errors in our bases. This is known as Learning With Errors (LWE) and by viewing these points as systems of equations. As per figure 3 below we have 3 equations with three variables. When

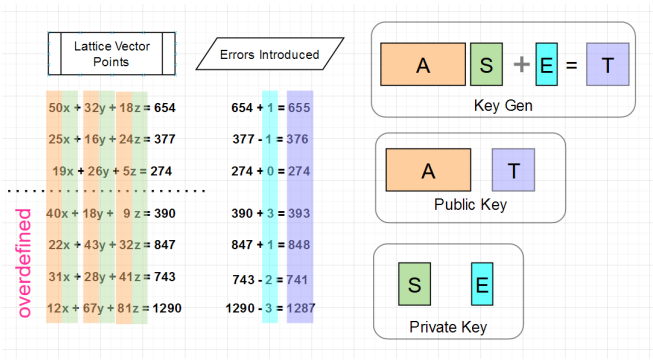

FIG. 3. Lattice Based Key Encryption Process

we over define the variables by adding more equations, we must use overdefined equation systems that are solvable. Now we add errors into our equations that slightly modify it. Thus we can easily spot the added errors, provided we know the variables. But without knowing the variables, it becomes very difficult. Thus we create a trapdoor function similar to RSA. An attacker needs to invert the process in order to find out the variables. When Alice sends over the equations to Bob she sends it with the errors . Bob then, by a random process, picks a few of the equations and adds them. He then sends back an equation which corresponds to either 0, which is the summed equation with a small error, or to a 1 with a large error on the sum. This is an inefficient method but it is the simplest way to explain some of the inner workings of Lattice Based cryptographic techniques. The way the key is distributed is:

1. Alice generates a key as per figure 3 above.

2. Bob Encrypts his message using Alice's public key as well as adding random errors on top of Alice's public key to make sure the message is not decipherable to an eavesdropper.

3. Alice receives the encrypted message and using her private key is able to decode and throw away the errors.

CRYSTAL-Kyber is one of the key-encapsulation mechanism finalists which uses LWE. An advantage of Lattice based algorithms is that they have fast computation time and a small public key, which is a good trade-off for a potential candidate. Kyber's design doesn't have any leaks from Timing based attacks as there are no secret dependent branches. With Differential attacks it is vulnerable to long term keys being attacked and requires further expansion and masking to secure it against it. There are a few other lattice based schemes that are currently in Round 3 such as SABER and NTRU. SABER relies on the hardness of solving Module Learning with Rounding, while NTRU is based on Ring-learning with Errors. [8]

IV. CONCLUSION

In this paper, we explored the quantum capabilities as well as the possible ways to overcome quantum attacks. These measures still require further refinement before they can beused as a new standard for communications. The variants of the Post Quantum Cryptography schemes explored are the Lattice-based CRYSTAL-Kyber and the Code-based Classic McEliece, both of which require more testing and have a long way to go before they are standardised. WIth the next stage of NIST Competition being early 2022, more Cryptographers will attempt to find flaws within each algorithm and help define the new standard.

[1] L. Johnson, *Security Controls Evaluation, Testing, and Assessment Handbook (Second Edition)*, second edition ed., edited by L. Johnson (Academic Press, 2020) pp. 471–536.

[2] J.-C. Deneuville, *Post-Quantum Cryptography: tomorrow's security* (2018).

[3] I. Thomson, Aes-256 keys sniffed in seconds using €200 of kit a few inches away (2021).

[4] D. Bruss, G. Erdélyi, T. Meyer, T. Riege, and J. Rothe, Quantum cryptography, ACM Computing Surveys **39**, 6 (2007).

[5] D. J. Bernstein, T. Chao, T. Lange, I. v. Maurich, R. Misoczki, and E. Persichetti, Classic mceliece: Conservative code-based cryptography (2021).

[6] M. Marcus, White paper on mceliecewith binary goppa codes (2021).

[7] N. Lahr, R. Niederhagen, R. Petri, and S. Samardjiska, Side channel information set decoding using iterative chunking, Cryptology ePrint Archive, Report 2019/1459 (2019).

[8] D. Bernstein, *Visualizing size-security tradeoffs for lattice-based encryption* (2019).

[9] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, *CRYSTALS-Kyber Algorithm Specifications And Supporting Documentation (version 3.01)* (2021).

[10] W. Barker, W. Polk, and M. Souppaya, Getting ready for post-quantum cryptography: 10.6028/nist.cswp.05262020-draft (2020).

[11] .