Post-Quantum Cryptography

Efeosa Louis Eguavoen
Student Number: 17324649

***Abstract*—In recent times, there has been a surge in interest in the field of quantum computing, a field that is proposed to have the ability to solve problems that are computationally difficult for conventional computing systems to solve rather trivially. Due to this fact, in areas such as modern cryptography that's built on the premise of such mathematical problems, there has been a huge spike in research in post-quantum cryptography or methods that are impervious to attack by quantum computers. Post-quantum cryptography assumes the attacker has access to a quantum computer with enough computational power to crack current systems and tries to create new methods that cannot be breached by those with access to such machines, while being able to integrate with current systems, protocols and networks. The core issue in post-quantum cryptography is creating methods that meet demands for security and usability without a significant loss in speed and confidence.**

I. Introduction

In the last few decades, public key encryption has become an integral part of telecommunications and an intrinsic component of modern communication infrastructure. These systems have been integrated into almost every sector of technology realted to with wireless communications, from connecting your phone to a Wi-Fi modem at home, e-commerce, social networking and cloud computing. The level of interconnectivity we face today is ever growing, it's imperative that we maintain the ability to communicate across these platforms securely and easily.

A. Modern Cryptography

When a user connects to a website using the HTTPS protocol, the user's computer uses Transport Layer Security (TLS) to connect to the web server. This is done to maintain security and privacy as nobody but the user and the server know what data is being transferred, preventing any third party from being able to intercept and snoop on data being sent as it's encrypted by some method. TLS employs a number of cryptographic methods to enable this functionality and to maintain confidentiality and integrity.

Consider the example of two users Alice and Bob who want to send a message m between them. If they both knew an encryption key k_{enc}, Alice could apply a symmetric encryption algorithm to m using this key to produce cipher text c which Alice could then send onto Bob over the internet. Bob would then use a matching symmetric decryption key algorithm using the same k_{enc} to decipher c to plain text. Both parties also know an authentication key k_{auth}. Alice applies a message authentication code using this key auth to the ciphertext to create an authentication tag which she then sends over the internet to Bob proving she has the key. Bob does the same to prove that he too, has the key. Symmetric encryption ensures the data remains confidential in the HTTPS protocol. It makes sure a third party can not see the message contents, while authentication ensures the messages are from the intended source and maintains the integrity of the messages, preventing them being altered by someone snooping or being changed entirely by someone pretending to be the party sending the message. A common example of symmetric key encryption is the Advanced Encryption Standard or AES.

For both Alice and Bob to have shared symmetric keys, they have to use public keys from the filed of public key cryptography. In this field, both parties have two keys; a public key and a private key. The private key is only known to each individual party while the public key is made public and anyone with the key can encrypt messages to that user, whilst that user uses their private key to decrypt their messages.

To ensure that the public key provided by Alice truly belongs to her, a cryptographic function is applied to public keys to create a 'signature' by a trusted party using their private signing key. Anyone can verify this signature using a verification algorithm using the public signing key of the trusted party. This links the public key of Alice with Alice's identity, giving Bob the confidence to encrypt with Alice's public key[1]. Real world systems that operate on this principle include RSA and ECDSA (Elliptical Curve Cryptography).

B. Quantum Computing

Now that we have an understanding of how modern cryptography works, we can now look at how quantum computing works. Quantum computers are rather different to classical computers because they work on the probability of an object's state rather than the object's definitive state meaning they have the ability to process exponentially more data than their classic counterparts.[2]

Classical computers carry out operations on binary

operators that only have 2 states, 0 or 1 ,called bits as we know. However, in quantum computing, operations are carried out on the quantum state of on an object to create a qubit. These states usually take up the characteristics of an unknown property of an object before they've been detected i.e the polarisation of a photon or the spin of an electron. They do not have a clear position like in bits but instead operate on the fundamental principle of quantum mechanics called quantum superposition, where they exist as all possible outcomes simultaneously until they've been quantified much akin to a coin flipping through the air before it lands on a surface.

These superpositions can become entangled with objects using quantum entanglement that enables us to know the outcome of the result of its partner without taking it's measurement as they are no longer independent of each other i.e if we measure the spin of one electron to be up, its entangled partner will always be spinning down. In this lies the inherent power of quantum computing as we can store a lot more data in a smaller test space and with this data we can compute all outcomes simultaneously as we never collapse the probabilities to a singular state. For example, a 100 qubit computer would have more computing power than all the supercomputers in the world.[3]

Using these complex interactions and the mathematics behind them, we can plug them into specialized algorithms that take advantage of these facts to solve problems that are usually NP hard for classical computers to do in relatively trivial time frames.[4]

C. Why It Matters

The most intrinsic communication protocols we employ rely on three cryptographic functions previously explained: public key encryption, digital signatures and key exchange, with symmetric key encryption being another major aspect. These protocols are implemented using Elliptical Curve Diffie-Hellman key exchange, RSA(Rivest-Shamir-Adleman) cryptosystem and AES for symmetric encryption. The security of such systems is dependent on the computational difficulty of theoretical problems such as Integer Factorization and Discrete Log Problem.

Herein lies the key problem. In RSA, the public key is a product $N = pq$ of two secret prime numbers p and q, with its security dependant on the difficulty of finding the two prime factors p, q of N. It was discovered by Peter Shor in 1994 that a quantum computer could find the prime factorization of any positive integer N with high efficiency[5], making this method obsolete and impotent, leaving our major systems open to attack from anyone with a quantum computer with enough power. While the exact costs of Shor's algorithm are still being researched and optimized, we've been able to reduce it to O$n^3 log n$ runtime using $2n + 3$ qubits if $N = pq$ fits into n bits[6]. With some slight alterations to Shor's initial algorithm, we can also break Elliptical Curve Cryptosystems, a popular alternative to RSA.

Grover's algorithm is another quantum algorithm that affects many cryptographic systems. It searches for the roots of a function f that satisfy the equation $f(x) = 0$. If one of every N inputs is a root of f ,then the algorithm finds a root using $\sqrt{N}$ quantum evaluations on the function f on the relevant superpositions of the inputs. It's dependant on the condition that it can be evaluated by a small circuit as this could not require many quantum evaluations and few qubit operations making it very easy to do. While the level of speed up isn't as immense as Shor's algorithm, it threatens to reduce the security of many of our current systems. If the circuit condition is true, it threatens to reduce the security of security systems that aim for 2^128 security such as 128bit AES keys.

II. Post-Quantum Methods and Analysis

The table above[7] provides insights into current cryptographic techniques and their ability to ward off attack from quantum systems. While not all cryptographic algorithms are equally affected, they all require some sort of change to remain viable in a post-quantum world. Symmetric key encryption such as AES[8] seems to be the most unaffected by quantum algorithms as there's only a quadratic speed-up for quantum algorithms in comparison to searches on classical computers. Due to this fact, to maintain safety all we need to do is increase the key size from 128bit to 256bit keys to maintain integrity and confidentiality.To implement such measures would have little to no effect on cost, in addition to it being easy to do without affecting/changing current protocols in place/use. Similarly for hash functions such as SHA-256, we do not necessarily need to change anything as their current security level is high enough that with quantum algorithms such as Grover's algorithm, their security level would only reduce to 128bit from 256bit which is still very acceptable.

In contrast to this, methods such as RSA,ECDH are completely broken by Shor's algorithm and need replacement by algorithms that are unaffected by Shor's algorithm and nobody has found an effective way to break their encryption. The problem lies with choosing secure key sizes for these algorithms so they are unaffected by Grover's algorithms but unlike AES, a penalty is incurred on doubling the key sizes so current research is aimed at reducing these costs or understanding the algorithm in further depth to be able to use smaller key sizes. The following proposed systems have been tested thoroughly or seem to have the most promise based on the papers I have read.

Impact of Quantum Computing on Common Cryptographic Algorithms

Cryptographic Algorithm	Type	Purpose	Impact from Quantum Computer
AES	Symmetric Key	Encryption	Larger Key sizes needed
RSA	Public Key	Signatures, Key Establishment	No longer secure
ECDSA,ECDH (Elliptic Curve Cryptography)	Public key	Signatures, Key Exchange	No longer secure

A. Code-based Cryptography

Code based encryption is one of the most prominent post-quantum cryptography techniques due to it's years of study. The first code based cryptosystem was created by Robert McEliece in 1978 and since that date remains unbreakable.[9]

Code-based cryptosystems are based on the idea that using a word of linear error-correcting code as ciphertext and then adding random errors to it. A basis of the code is used as the the public key allowing anyone to encrypt data. Users who know the decryption key which is a decoding algorithm for the code, can remove the errors and decode the ciphertext to clear text.3rd parties wanting to decode the cipher text are left to solving generic decoding problems, which are hard on average. This includes those with quantum capabilities.

The security of such as system comes from two computational assumptions:

- Generic decoding can not be solved efficiently or with large enough key sizes, the problem becomes essentially unsolvable.
- The public key, a generator matrix, is hard to distinguish from a random matrix.

The first problem is NP-complete and is hard on average. The second problem is much more open. To ensure security, the system must be instantiated properly.McEliece suggested using the family of binary Goppa codes as the indistinguishability assumption still holds. Other families of codes such as low-density parity check codes and concatenated codes the asssumption doesn't hold, making the system insecure.

There are no real world systems using code-based primitives as of yet, this is due to a few major factors:

- Digital signature systems are not entirely practical yet.
- The public key size is large and becomes even larger for quantum proof security.

Despite these drawbacks, it remains one of the most promising areas of research as it's security is well understood, has withstood 40 years of scrutiny and testing and is computationally efficient in terms of encryption and decryption.

B. Lattice-based Cryptography

Lattices in this context are basically a grid of regularly spaced points stretching to infinity. Lattices are comprised of **vectors** or points in the lattice made of coordinates. A special vector we take notice of is the **origin** or a vector where all its coordinates are 0. We can define a vector as a long vector if it is far from the point (0,0). Inversely, a short vector is one that is close to the origin. Lattices can stretch to inifinity but computers have a limit to the amount of data they can store ,so we represent a lattice as a **basis** or a small collection of vectors that can be used to reproduce any point in the grid that forms the lattice.
For example in the case of a 2D lattice, we choose 2 points that are not on a line that bisects the origin i.e (2,0) and (0,2). We can generate a new point using the two previoulsy selected points. To do this we:

- Choose 2 numbers at random i.e 4 and -2.
- Multiply the first point by 4 to create the point (8,0) and then the second point by -2 to get (0,-4).
- Adding the results we generate the point (8,-4).

Using the previously outlined method, we can create an entire grid of points. The main idea here is that by selecting a basis, we can essentially generate an entire lattice based on the vectors in the basis. More importantly, this can be stored in memory as it's finite. Similar to long and short vectors, a basis is considered short if it's composed of short vectors and long if composed of long vectors.

Based on the above information, a number of hard lattice problems were created that form the basis of lattice cryptography. The **Short Vector Problem** is arguably the single most important problem in lattice based cryptography. It asks us to suppose:

- We are given a long basis for some lattice L.
- The short vector problem asks us to find a grid point in L as close as possible to the origin point.

While the question appears trivial, we are given long vectors so it's not immediately clear how to combine them to generate a point with small coordinates. Also we are dealing with much higher dimensions than the example given, instead of 2 dimensions we may be given 10,000 dimension vectors. So finding a combination of the basis vectors that generates a small vector for all 10,000 coordinates is rather difficult, so much so that we don't even know how to do it quantum computers, never mind classical computers.

One real world example of post-quantum lattice

based encryption is NTRU[10]. In this system the public key is a p-coefficient polynomial $h = h_0 + h_1x + ... + h_{p-1}x^{p-1}$ with each coefficient in the set $(0, 1, ..., q-1)$. A ciphertext is another polynomial c. The sender then follows the above procedure to generate c according to the formula $c = ((hd + e) \text{ mod } x^p - 1) \text{ mod } q$. The Lattice L is a 2p-dimensional lattice containing a point close to (0,c). The problem follows the short vector problem as the attacker must find the d,e given c and public key h. Decoding of the algorithm is efficient as a secret public key is generated also.
In terms of security, NTRU[11] has yet to been broken but there are potential avenues of attack as the nature of the systems can be used against them, causing some systems to be broken by extensions of Shor's algorithm. This can be easily rectified by changing $x^p - 1$ with $x^p - x - 1$.

Lattice based systems are fairly well understood and studied going as far back as the 1800's so we can have a high degree of confidence in the degree of intractability of these problems. Due to this fact a large number of the most promising post-quantum cryptography methods are based on lattice-based cryptography and a large number of the entries into the US National Institute for Standards in Technology are based on lattice systems. The versatility of such systems allow them to replace a large number of our current systems from encryption to key exchange. The efficiency of such systems is rather high in comparison to other quantum methods such as code based encryption as key size created by NTRU is much smaller than a key created by the McEliece's system. Additionally lattice based cryptosystems not susceptible to side channel attacks unlike code based encryption.

C. Digital Signatures and Key Exchange based on Super singular Isogenies

Elliptical curve based cryptography can be broken by extensions of Shor's algorithm by replacing parts of the equation with points on an elliptical curve. As a response to this, super singular isogeny based systems[12] were created to replace the elliptical curve based systems as they are designed to be impervious to attack by an adversary with access to a quantum computer. Similar to Diffie-Hellman key exchange, super singular isogeny Diffie-Hellman key exchanged(SIDH) has one of the smallest key sizes at 2688bit public keys at a quantum security level of 128bit.
Super singular isogeny based systems work with graphs whose vertices are super singual elliptical curves and whose edges are isogenies between those curves. An isogeny between curves E' and E is a rational map which is also a group homomorphism. The set of isogenies mapping a curve E to itself forms a ring under called the endomorphism ring. A curve E is super singular if it's endomorphic ring is isomorphic to an order in quaternion algebra and ordinary otherwise.

The security of such systems are heavily based on the following 2 problems which are believed to be intractable even by quantum computers:

- **Computational Super singular Isogeny problem**: Let ϕA: $E_0 \rightarrow E_A$ be an isogeny whose kernel is (R_A) where (R_A) is a random point with order l_A^{eA} . Given E_A,ϕA(P_B),ϕA(Q_B), find a generator of (R_A)
- **Decisional Super singular Product (DSSP) problem**: Let ϕ: $E_0 \rightarrow E_3$ be an isogeny of degree l_A^{eA} . Given(E_1,E_2,ϕ') sampled with probability 1/2 from one or the other of the following distributions, determine which distribution it is from.
  – A random point R of order l_B^{eB} is chosen and E_1 = E_0/R , E_2 =E_3/ϕR , andϕ': $E_1 \rightarrow E_2$ is an isogeny of degree l_A^{eA}.
  – E_1 is chosen randomly among curves of the same cardinality as E_0, and ϕ':$E_1 \rightarrow E_2$ is a random isogeny of degree l_A^{eA}

The best attack for these problems is solving the related claw finding problem as proposed by Dr Jao[12] resulting in a complexity of O($p^{\frac{1}{4}}$) for classical computers and O($p^{\frac{1}{6}}$) for quantum computers.

As for the performance of such systems,6144bits are necessary to be transmitted for a security level of 128bit. This key size can be further reduced using key suppression techniques. With these techniques, SIDH has a similar bandwidth requirement as current Diffie-Hellman exchange systems. In contrast NTRU has a larger key size. In 2016, Microsoft researchers showed SIDH can be ran in constant time with a key size of only 564 bytes, making it the most efficient implementation till date.
As for digital signatures, similar results were obtained by researchers working on using super singular isogenies for digital signatures instead of key establishment. The majority of costs of implementing the systems can be precomputed offline, meaning the signing algorithm needs to only evaluate a hash function on the data and output an appropriate response for the signature.

Based on the presented information, super singular isogeny based encryption seems to be another viable encryption system to replace current systems with strict key size requirements. They seem to be able to cover the full range of cryptographic primitives when combined with pre-existing research. The sophistication of the mathematics used in such systems inspires a high amount of confidence and the implementation of stateless quantum-resistant digital

Overview of Post-Quantum cryptosystems

Approach	Advantages	Disadvantages	Example
Code-Based Encryption	Fast encryption,High Confidence	Large Key size	McEliece's code based systems
Lattice Based Encryption	Short ciphertext and Keys, fast encryption	More analysis necessary	NTRU
Super singular Isogeny based cryptosystems	Small key size, fast decryption	More Analysis necessary	SIDH

signature schemes based on super singular isogenies with very small key sizes by researchers shows the technology is a viable and not just theoretical in nature. The major obstacle facing this field of post quantum cryptography is the lack of research as it's still an emerging field that's in its infancy in comparison to other fields such as lattice based cryptography, so more work needs to be done to verify the integrity of the field.

III. Discussion

The future of post-quantum cryptography is very bright, but some major obstacles still stand in the way before it can be fully adopted.

A. Standardization

One of the foremost problems is the lack of standardization of post-quantum techniques. This is important as systems often require everyone to be using the same systems for them to work as intended. Thus for widespread adoption of these techniques, the establishment of an industry standard is necessary. There are a number of bodies attempting to create standards in this field such as the Internet Engineering Task Force who are in the process of standardizing hash-based signature systems. On the other hand, bodies such as the National Institute of Standards and Technology are actively looking for candidates for standardization by running an open call for submissions starting in 2017, scrutinizing candidates over the course of 3-5 years, in an effort to find and standardize methods in each field[13], similar to how AES was previously standardized. Once these standards are established, they wish to compare them to existing standards and based on the current threat level of quantum computers, may depreciate or withdraw the affected standards.
Despite a lack of standards, there have already been some real world deployments of post-quantum cryptography. One such example is Google's application of the 'NewHope'[14] lattice-based cryptosystem. A fraction of all Chrome users connecting to Google sites traffic were encrypted with the cryptosystem and the experiment proved to be a success and viable for future implementation. From the results of the experiment they saw:

- A slow down of 20ms for the slowest 5%
- A slow down of 150ms for the slowest 1%

From the results, most users experienced no difference in normal usage and only a small set of users experienced any slow down, due to increased message sizes, as NewHope is computationally inexpensive.
While it's exciting to see we do have the ability to switch to post-quantum methods, much more testing is necessary before we can have full confidence in switching to such systems.

B. Integration

Another issue is the integration of post-quantum cryptosystems into current systems. Deploying new systems incur costs such as the time required to communicate new keys, signatures and so forth. Also we need to factor in the average user's ability to use and run such systems based off their budget. It's redundant to have systems that only a minority of the population can use or access due to budget reasons or lack of computational power.
We also need to be wary that the mathematical models of attacker capabilities match that of real world adversaries. For example 'side-channel' attacks where an attacker observes physical outputs of a cryptographic system at work, such as power consumption, cache access or timing. There needs to be thorough testing of finding the maximum security achievable under real world constraints so we can have an idea how these systems actually perform and are attacked in the real world.
While some state of the art systems are quantum proof, they are often held back by things such as large key sizes or an excessive amount of computational power needed for them to be implemented. One such example is the McEliece's code-based system that is less intensive to compute than current elliptical curve cryptosystems but is held back by large key sizes.

IV. Conclusions

The field of post-quantum is still in its infancy but there has been a huge amount of progress made in the field in a relatively short amount of time. Many of the methods outlined in this paper will more than likely be adopted in one form or the other but with altered parameters once they've been tested more thoroughly. But it's also likely that some of these methods may be broken new undiscovered algorithms after extensive testing but as we learn about these methods, we can continually improve these algorithms. Bodies such as NIST and IETF leading the push for standardization advances the field, creating new designs, more optimizations and ultimately more attacks as we push for systems we can reliably use and can maintain integrity and confidence. Once systems are well studied, we can adopt these systems into existing frameworks to

create safer systems for all. While much more work is necessary before we can reach that milestone, it is truly an interesting time to witness the progress of post-quantum cryptography, with an eye to see what the future holds in the development of this area.

REFERENCES

[1] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017.

[2] H. Buhrman *et al.*, "Quantum computing and communication complexity," *Bulletin of the EATCS*, vol. 70, pp. 131–141, 2000.

[3] T. Greene, "Here's why 100 qubit quantum computers could change everything," 2018.

[4] J. Gruska, *Quantum computing*. Citeseer, 1999, vol. 2005.

[5] M. Kaplan, G. Leurent, A. Leverrier, and M. Naya-Plasencia, "Breaking symmetric cryptosystems using quantum period finding," in *Advances in Cryptology – CRYPTO 2016*, M. Robshaw and J. Katz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 207–237.

[6] S. Beauregard, "Circuit for shor's algorithm using 2n+ 3 qubits," *arXiv preprint quant-ph/0205095*, 2002.

[7] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, *Report on post-quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016, vol. 12.

[8] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, "Applying grover's algorithm to aes: quantum resource estimates," in *Post-Quantum Cryptography*. Springer, 2016, pp. 29–43.

[9] N. Sendrier, "Code-based cryptography: State of the art and perspectives," *IEEE Security Privacy*, vol. 15, no. 4, pp. 44–50, 2017.

[10] J. Hoffstein, J. Pipher, and J. H. Silverman, "Ntru: A ring-based public key cryptosystem," in *International Algorithmic Number Theory Symposium*. Springer, 1998, pp. 267–288.

[11] J. Bi and Q. Cheng, "Lower bounds of shortest vector lengths in random ntru lattices," *Theoretical Computer Science*, vol. 560, pp. 121–130, 2014.

[12] Y. Yoo, R. Azarderakhsh, A. Jalali, D. Jao, and V. Soukharev, "A post-quantum digital signature scheme based on supersingular isogenies," in *Financial Cryptography and Data Security*, A. Kiayias, Ed. Cham: Springer International Publishing, 2017, pp. 163–181.

[13] G. Alagic, G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, Y.-K. Liu, C. Miller, D. Moody, R. Peralta *et al.*, *Status report on the first round of the NIST post-quantum cryptography standardization process*. US Department of Commerce, National Institute of Standards and Technology, 2019.

[14] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange—a new hope," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 327–343. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim