Post-Quantum Cryptography

Efeosa Louis Eguavoen
Student Number: 17324649

April 25, 2020

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's 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.

1 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 to do with wireless communications, from connecting your phone to a Wi-Fi modem at home, e-commerce, social networking and cloud computing. Considering the level on interconnectivity we face today, it's imperative that we maintain the ability to communicate across these platforms securely and easily.

1.1 Modern Cryptography

When a user connects to a website using the HTTPS protocol, the user's computer uses Transport Layer Security or 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.

For example, If we had 2 users ,Alice and Bob, and they wanted to send a message m between them. If they both knew an encryption key k_{enc}, Alice could apply a symmetric encryption algoritm using this key to produce cipher text c which Alice can 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 also sends over the internt to Bob proving she has the key. Bob does the same to prove he has the key also. Symmetric encryption ensures the data remains confidential in the HTTPS protocol. It makes sure a third party can't see the message contents, while authentication ensures the messages are authentic from the 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 known as the Advanced Encryption Standard or AES.
For both Alice and Bob to have shared symmetric keys, they hav to use 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. Everyone 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 confience to encrypt with Alice's public key. Real world systems that operate on this principle include RSA and ECDSA(Elliptical Curve Cryptography).

1.2 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 differnet to classical computers because they work on the probability of an objects state rather than the objects definitive state meaning they have the ability to process exponentially more data than their classic counterparts.
Classical computers carry out operations on binary operators that only have 2 states, 0 or 1 called bits as we know. But 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 polaristaion of an photon or the spin of an electron. They don't have a clear position like in bits but instead operate on the fundamental principle of quantum mechanics called quantum super position, 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 it's partner without taking it's measurement as they are no longer independant of each other i.e if we measure the spin of one electron to be up, it's entangled partner will always be spinning down. In this lies the inherent power of quantum computing as we can store a lore more data in a smaller test space and with this data we can compute all outcomes simultaneously as we never collapes the probabilities to a singular state. For example, a 100 qubit computer would have more computing power than all the supercomputers in the world.
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.

1.3 Why It Matters

The most intrinsic communciation 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 dependant 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 it's security dependant on teh 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 factorixation of any positive integer N with high efficency making all 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 to O$n^3 logn$ runtime using 2n + 3 qubits if N=pq fits into n bits. 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 qould 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 $2^1$28 security such as 128bit AES keys.

2 Post-Quantum Methods and Analysis

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

The table above 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 seem to be the most unaffected by quantum algorithms as there's only a quadratic speedup in 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 confidentiallity. Implementing this change is fairly easy and the cost of implementing such changes is hardly noticeable and can be implemented without changing any of the current protocols in place. Similarly for hash functions such as SHA-256, we do not necessarily need to change anything as thier current security level is high enough that with quantum algorithms such as Grover's algorithm, their security level would only recude 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 algotirhms 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 off the papers I have read.

2.1 Lattice-based Encryption

Lattices in this context are basically a grid of regularly spaced points streching to infinity. Lattices are comprised of **vectors** or a points in the lattice made of coordinates. A special vector we take notice of is the **origin** or a vetor where all it's coordinates are 0. We call a vector long if its far from the origin or short if it's close to the origin. As lattices are infinitely large but computers have finite amounts of memory, 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 first choose points that don't lie on a single line through the origin i.e (2,0) and (0,2). Using these points we can generate a third point by:

- First we choose 2 numbers at random i.e 3 and -1.
- We then multiply the co-ordinates of the first point by 3 to get the point (6,0) and the coordinates of the sceind point by -1 to fet (0,-2).
- Adding the results we get the new point (6,-2).

Using this method we can generate an enitre gird of evenly spaced points. The core idea is that by choosing a basis we can choose an entire lattice based on the vectors in the basis. Most importantly this can be stored in memory as a finite object. We say that a basis is short if it consists of short vectors or long if it consists of long vectors.

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

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

While the question appears trivial, we are given long vectors so it's not immediatley 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 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. In this system the public key is a p-coefficent polynomical $h = h_0 + h_1x + ... + h_{p-1}x^{p-1}$ 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 the formula $c = ((hd + e) \bmod x^p - 1) \bmod q$. 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 efficent as a secret public key is generated also.
In terms of security, NTRU 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 of degree of unsolvability 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 entires into the US National Institue 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 efficency 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. Also it's not susceptible to side channel attacks unlike code based encryption.

2.2 Digital Signatures and Key Exchange based on Supersingular Isogenies

Elliptical curve based cryptography can be broken an extentions of Shor's algorithm by replacing parts of the equation with points on an elliptical curve. As a response to this, supersingular isogeny based systems 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. Much alike Diffe-Hellman key exchange, super-singluar isogeny Diffie-Hellman key exhanged(SIDH) has one of the smallest key sizes at 2688bit public keys at a quantum security level of 128bit.
Supersingular isogeny based systems work with graphs whose verices are supersingual elliptical curves and whose edges are isogeines 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 supersingular if it's endomorphic ring is isomorphic to an order in quarternion 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 Supersingular 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 Supersingular 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 Jaoresulting in a complexitiy 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 supression techniques. With these techniques, SIDH has a similar bandwith 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 witha key size of only 564 bytes, making it the most efficent implementation till date.
As for digital signatures, similar results were obtained by researchers working on using supersingular isogenies for digital signatures instead of key establishement. 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, supersingular 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 primatives when combined with preexisting research. The sophistication of the mathematics used in such systems inspires a high amount of confidence and the implementation of statless quntum-resistant digital 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.