American Journal of Electrical and Computer Engineering
2020; 4(2): 81-93
http://www.sciencepublishinggroup.com/j/ajece
doi: 10.11648/j.ajece.20200402.17
ISSN: 2640-0480 (Print); ISSN: 2640-0502 (Online)

Review Article

Cybersecurity in a Post-Quantum World: How Quantum Computing Will Forever Change the World of Cybersecurity

Marc Nahed, Shadi Alawneh*

School of Engineering and Computer Science, Oakland University, Rochester, the United States

Email address:
mmnahed@oakland.edu (M. Nahed), shadialawneh@oakland.edu (S. Alawneh)

*Corresponding author

To cite this article:
Marc Nahed, Shadi Alawneh. Cybersecurity in a Post-Quantum World: How Quantum Computing Will Forever Change the World of Cybersecurity. *American Journal of Electrical and Computer Engineering*. Vol. 4, No. 2, 2020, pp. 81-93. doi: 10.11648/j.ajece.20200402.17

Received: December 11, 2020; **Accepted**: December 18, 2020; **Published**: December 25, 2020

Abstract: High Performance Computing (HPC) has recently been considerably improved, for instance quantum computing has been developed to achieve high performance computation in many areas, such as medical research, artificial intelligence, weather forecasting, etc. But it also poses a significant threat to cybersecurity, requiring changes to data encryption methods. Currently, the most widely used asymmetric algorithms are based on difficult mathematical problems, such as factoring large numbers, which can take thousands of years on today's most powerful supercomputers. The purpose of this paper is to dive into the field of cybersecurity and understand how modern practices will be affected by the advancements of quantum computing. In doing so, a fundamental understanding of modern-day computing, modern-day cybersecurity, and quantum computing will need to be established. This, in turn, will build the foundation to allow for a comprehensive analysis of how powerful quantum-based computing is in comparison to modern-day computing, and how this disruptive technology will ultimately change the field of cybersecurity on a global scale. In addition, current industry cybersecurity best practices will be presented to expose their projected vulnerabilities as well as what can be done in the immediate future to prepare for the ever-rapid advancements in computing. Finally, conclusions will be extrapolated on what is to come for future generations in the ongoing race between computing and cybersecurity.

Keywords: Quantum Computing, Cybersecurity, High Performance Computing

1. Introduction

In theory, the greatest limitation to technology is not having the ability to innovate. Technology is culmination of what already exists. To advance is to have the ingenuity to improve on what is already provided. In practice, the greatest limitation to technology is not having the ability to protect. Technology without boundaries is harmful. Technology without safety is dangerous.

Cybersecurity, as a field of study, is relatively new. It emerged in the early 1970s as a research project led by Bob Thomas, known as The Advanced Research Projects Agency Network (ARPANET). The objective of this project was to create a computer program that was able to travel through the ARPANET network, a supposedly secure network of computers, and leave a small trail behind [1]. The program was called CREEPER and it would print out the message "I'M the CREEPER: CATCH ME IF YOU CAN" on the screen of the computer it infiltrated [1]. The purpose of this project was to prove that the ARPANET network, which, at the time, was implemented by large organizations and government agencies, had major security flaws that can be exploited by well-devised computer programs to steal and manipulate important data. The outcome of this research project sparked the attention of the industry and began to engage scholar and professional alike to start focusing their efforts on improving network security.

However, this awaking came late, as the two decades to follow saw an exponential increase in technological advancements within the field of computing and connectivity, while cybersecurity, in its infant stages, did not have the capability to keep up. The security systems in place to combat

cyberattacks were not sufficient, and as a result, major networks were compromised by cyber attackers in the hunt for highly sensitive data that can be used for extortion and manipulation. During the cold war, the Russians, with the help of German computer hackers, developed a cyber weapon in 1986 to breach the United States military's networks in pursuit of war secrets [1]. Through much success, 400 military computers were breached, including mainframes at the pentagon [1]. Following this attack, in 1988, the United States allocated significant resources to drastically propel the field of cybersecurity forward and into the 21st century [1].

With much focus, significant progress has been made, enabling current cybersecurity measures to surpass modern computing advancement, and making cyber-attacks exponentially more difficult to succeed. Organizations and committees have formed to develop globally recognized cybersecurity standards and protocols that have been proven to be effective in today's and the foreseeable future's computing climate. However, much of the focus on cybersecurity by industry poses a single major flaw. The "established" strategy for future security is contingent on the fact that computing power will grow relatively linear. What happens if computing capabilities grow exponentially? With the introduction to quantum-based computing, providing a solution for exponentially capable computing power, the current state of cybersecurity could be in jeopardy. If measures are not taken early enough to account for quantum computers, the state of security will return to what it was in the 1970s and 1980s.

2. The Modern Computer

Computers are complex. Through the lens of an engineer, the culmination of over a century of refined technological innovation can not easily be explained in a single paper. And fortunately, it will not be needed because what makes quantum computing unique is not rooted in engineering – it is rooted in physics. Therefore, this section will focus on the physics that drives modern-day computers, which will later be compared to the physics that drives quantum computers.

At the core of all computers is a transistor. A transistor is an electronic switch made of semiconductor material that has the ability to change its electrical state if pulsed with voltage [2]. In the absence of voltage, the transistor is nonconductive, impeding current. In the presence of voltage, the transistor is conductive, allowing current to flow. It is a clever on/off switch in which every pulse of voltage represents a single bit of data – 0 (low voltage) or 1 (high voltage) [2]. Therefore, creating a sequence of pulses, allows for data encoding. More so, by strategically combining transistors together creates a logic gate. Logic gates enable computation of which a logical operation is performed on one or more binary inputs with the result being a single binary output. Combining logic gates together forms a circuit, allowing for more complex decision-making. Furthermore, combining circuits together creates an electronic system, commonly known in a computer as a central processing unit (CPU).

To discuss the capability of modern computers is to discuss the speed of which the CPU can process data. The faster the CPU can derive an output from a given input, the more capable a computer is to perform more complex computations in a shorter period of time. This, in turn, allows more advanced technological developments such as machine learning and artificial intelligence to be more effective.

There are three physical limitations that hinder the speed of a CPU: speed of electrical transmission, gate delay, and heat. Through nearly a century of innovation, great strides have been made to improve on the performance of the CPU by finding better techniques to minimize gate delay and heat creation. With that being said, because a computer operates under the natural laws of physics, the one limitation that can not be improved upon is the speed of electrical transmission. Albert Einstein's theory of special relatively dictates that the speed of light can not exceed 300 million meters per second. Therefore, under ideal conditions where gate delay and heat are negligible, the CPU will maximize on its performance based on the speed in which electricity can through the circuit.

3. Quantum Computing

"Quantum computing is the area of study focused on developing computer technology based on the principles of quantum theory, which explains the nature and behavior of energy and matter on the quantum (atomic and subatomic) level," [3]. As discussed previously, modern-day computers process data in an exclusive binary state, which can be either one or zero, also known as a bit. The bit value of zero (0) represents off or false, and the bit value of one (1) represents on or true. In a computer, which is made up of hundreds, if not millions or billions of transistors, can only exist in one state at a time. With the constant evolution of technology, transistors have become much faster in switching between the two states; however, there is still a restraint on how fast they can operate. As transistors become more efficient, they begin to reach the limits dictated by laws of physics. Beyond this point, the only way to improve on the performance is to utilize the theories of quantum physics. With these laws come several quantum phenomena, which include superposition and entanglement. These properties give quantum computing the unique characteristics that can not be taken advantage of by modern-day computers.

In 1925, Niels Bohr and Werner Heisenberg developed the *Copenhagen Interpretation*, introducing the early working principles of quantum physics. They introduced two fundamental concepts, which laid the foundation of which quantum computers operate under.

1. A particle or system of particles not observed exist in a state of superposition – being in all possible states at once [4].
2. An observed particle or system of particles causes the superposition to randomly collapse to one possible state [4].

In the quantum world, a particle can be in multiple states at the same time, which is known as the property of

superposition. It has the ability to be a photon, an electron, or any other type of particle. In reference to quantum computing, these particles are known as qubits. Based on this property, the same qubit has the ability to be in multiple quantum states at the same time in which it could have a value of one, zero, or a *superposition* of both [5]. This in turn lets one qubit perform two computations in the same step, two qubits can do four, three qubits can do eight, and so on [3]. Following the pattern, the number of computations that a computer can undertake in a single step is 2^n, where n is the number of qubits being utilized [3]. This significantly differs from modern-day computers in which a bit is limited to a single value – 0 or 1 – and therefore, only one computation can be performed in a single step. The restraint with modern-day computing is that, even with parallel processing, the number of computations achievable in each step is linear (i.e., $2n$). As a result, the computing power of a quantum computer far exceeds any modern-day computer's capabilities.

To understand how particles interact with each other in the quantum world, the property of entanglement is used. Two light photons that collide will create a system of particles, acting as one, having equal but opposite spin and charge. The fundamental concept of entanglement is that the system of particles must maintain perfect equilibrium. A particle having a negative charge must be balanced by a particle with a positive charge. A particle that spins to the left must be balanced by a particle that spins to the right. Within a system, two particles are considered to be linked, boundless by space. By measuring the quantum state of one particle will consequently reveal the state of the other particle, regardless of the location of the two particles [5]. As a result, by changing the quantum state of one particle will consequently change the state of the other particles, even if the particles were millions or billions of miles apart from each other. In quantum computing, this allows the manipulation of multiple qubits in a single step; therefore, enabling rapid communication over long distances and powerful computation to be achieved.

A notorious example of this property is defined by Bell's Theorem. Suppose there are two particles, A and B, which are connected through quantum entanglement. By being entangled, the properties of these particles are then correlated. Based on the law of superposition, before measuring the state of a particle, each particle can either be 1/2 or -1/2 [6]. Based on the theorem, by measuring the state of particle A, the state of particle B will also be known, which will be the opposite state of particle A [6]. Therefore, if particle A is measured as 1/2, then particle B will be -1/2, and vice versa. Understanding this concept will allow information to be communicated between the two particles.

Combining the properties of superposition and entanglement lead to a very powerful ideology that is the basis of quantum computers. In a classical computer, a 2-bit register has the ability to store one of four configurations at a given time – 00, 01, 10, and 11. In a quantum computer, a 2-qubit register has the ability to store all four configurations at the same time; therefore, exponentially enhancing the power and efficiency of a classical computer. However, like every great technology, developing a fully functional quantum computer with this capability is no easy task. Quantum computers are a new technology and like any technology, it will take decades of successful iterations to eventually develop a fully functioning computer.

There are three types of quantum computers that were defined to aid in the development of this technology, which include: quantum annealer, analog quantum, and universal quantum. Each type of quantum computer provides a natural progression of development by building upon each other. The quantum annealer is a specialized form of quantum computing with very little proven advantages over other specialized form of conventional computing but allows the fundamental concepts to be exercised [7]. The computation power is equivalent to a modern-day computer and its application is often only used to handle optimization problems. The analog quantum is the most likely form quantum computing that will first show true quantum speedup over conventional computing [7]. The computation power is high, and its applications include aiding in the advancements of quantum chemistry, material science, and quantum dynamics. The universal quantum is the true grand challenge in quantum computing as it offers the potential to be exponentially faster than modern-day computers [7]. The computation power is very high, and its applications include secure computing, machine learning, cryptography, and searching.

When comparing the computation power of a quantum annealer to a modern-day computer, it is surprising to discover that they are the same. How can a quantum computer, operating under the theories of quantum physics not be more powerful than a modern-day computer operating under the constraints of the natural laws of physics? The answer to that lies in the study of quantum supremacy – the ability for quantum computers to outperform modern-day computers. Quantum supremacy states that there is a certain number of qubits that a quantum computer must operate with in order to outperform a modern-day computer [8]. The magic number varies as new discoveries are uncovered; however, it is estimated to lie somewhere between 49 to 72 qubits [8]. In the progression of developing a quantum computer, once this processing power is established and quantum supremacy has been achieved, the classification of the quantum computer will shift from a quantum annealer to an analog quantum computer.

Having achieved this level of quantum computing will fundamentally change the way the world operates. It will bring forth new fields of studies currently unknown to mankind and will exponentially accelerate future technological advancements.

4. Modern-day Cybersecurity

Cybersecurity is a broad field of study, however, only a fraction of it is affected by advancements in computing; the most susceptible being cryptography. Cryptography is the practice of encoding information in order to secure a line of communication between two or more parties through an

untrusted medium. The two major steps of cryptography are encryption and decryption. Encryption is the process of disguising information so that it is unreadable. Decryption is the process of unveiling the encrypted information in order to read it. Both of these are done using a cipher, which is simply an algorithm capable of encryption and decryption, and a key, which is used to set the parameters that will dictate what the cipher does. In the past, the two most well-known types of encryptions operations used were substitution and transposition. Substitution cipher is the method in which the letters of the message (plaintext) are systematically replaced by different letters. Transposition cipher is the method in which the letters of the message are rearranged. Based on the encrypted message, it is up to the hacker to figure out a method of decryption without the knowledge of the cipher or the key, often referred to as cryptanalysis. For modern computers, substitution and transposition ciphers are not difficult to decipher. For this reason, more complex methods for encryption were created.

Modern-day cryptography can be broken down into three main categories - symmetric-key cryptography, public-key (asymmetric-key) cryptography, and hash-based cryptography.

A. Symmetric-Key Cryptography

Symmetric-key cryptography is a method in which the same key is used for both encrypting and decrypting data; therefore, the same key must be known to both the sender and receiver [9]. How the key is shared with both parties will be covered when discussing public-key cryptography. Figure 1 displays the flow of symmetric-key cryptography. In general, this type of cryptography uses complex data manipulation as the basis of the encryption and decryption algorithm [9]. The two main cryptographic standards that fall under this category are Data Encryption Standard (DES) and Advanced Encryption Standard (AES).

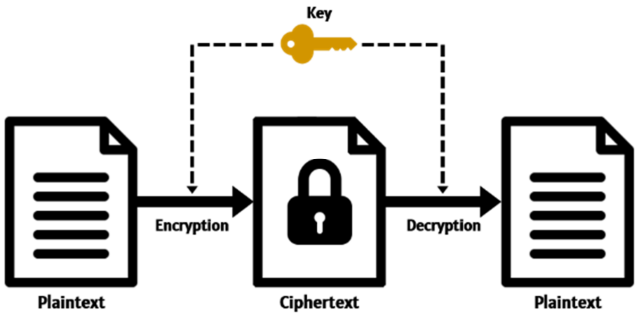

Figure 1. *Symmetric-key Cryptography.*

DES is a very established and widely used block cipher. It uses a 64-bit key – 8-bit parity and 56-bit key – to encrypt and decrypt 64-bit data [10]. The same algorithm and key are used for encryption and decryption. DES encryption consists of three mains steps: initial permutation, 16 rounds of data manipulation, and final permutation [10]. An initial permutation is performed on plaintext, which divides the plaintext into two separate 32-bit permuted blocks, commonly referred to as Left Plain Text (LPT) and Right Plain Text (RPT). LPT and RPT then go through 16 rounds of data manipulation. During each round, Equation 1 and 2 are used.

$$L_n = R_{n-1} \tag{1}$$

$$R_n = L_{n-1}\, f(R_{n-1}, K_n) \tag{2}$$

$n = current\ round, where\ 1 \leq n \leq 16$
$L = left\ permuted\ block\ value$
$R = right\ permuted\ block\ value$
$K =$
$unique\ 48\ bit\ key\ derived\ from\ the\ 64\ bit\ DES\ key.$

In each round, the L block value is the value of the R block from the previous round. The R block value is determined by taking the bit-by-bit exclusive-OR (XOR) of the L block from the previous round with the result from applying the DES cipher *f* to the R block from the previous round and K. After the 16 rounds have been completed, a final permutation is performed to recombine the 32-bit L and R blocks back to a single 64-bit ciphertext.

The largest concern with DES is that, for modern computers, the input key is not large enough to provide comprehensive security for highly classified information. As an alternative, Triple-DES was developed as a method to overcome the vulnerability flaws that arose from DES. Instead of a 56-bit key, Triple-DES applies the DES cipher algorithm three different times with three different keys. As a result, the combined key size becomes three times greater than the original key size (56-bits) – a key size of 168 bits [11]. DES/Triple-DES is commonly used in embedded systems and network devices, such as SIM cards, modems, and routers.

AES, on the other hand, is the latest standard for encryption in which the algorithm uses either a 127, 192, or 256-bit key to encrypt and decrypt data that is 128, 192, or 256-bit long [12]. The algorithm works by first inserting the data into a matrix and then repeating a number of different cipher transformations over multiple rounds to encrypt the data [12]. The matrix will consist of 4 rows and 4, 6, or 8 columns, depending on the size of the input data, where each position on the matrix represents a single byte of data. The matrix will iterate through 10 to 14 rounds of data manipulation, depending on the key size, to ensure that the plaintext is completely secure. In each round, the data undergoes four different cipher transformations, which include: SubBytes transformation, ShiftRows transformation, MixColunns transformation, and AddRoundKey transformation [12]. SubBytes transformation substitutes the 16, 24, or 36 input bytes of data into the pre-defined matrix structure using a substitution scheme defined by the AES specification. ShiftRows transformation cyclically shifts the bytes of each row in the matrix to the left. MixColunns transformation uses a mathematical function to modify the values of a given column within the matrix. Following this transformation, the data in the matrix is formatted back to a 128, 192, or 256-bit block. AddRoundKey transformation performs a bit-by-bit exclusive-OR (XOR) of the data block and key, resulting in the final ciphertext. Because this algorithm is much more thorough when encrypting and decrypting data, it is often the

preferred choice for "communication and commercial transactions over the internet" [13].

B. Public-Key Cryptography

Public-key (Asymmetric-key) cryptography is a method in which a public key is used when encrypting data and a private key is used when decrypting data; therefore, anyone has the ability to encrypt data, however, only a receiver that possess the private key has the ability to decrypt that data [9]. The public and private keys are mathematically related, although knowledge of one of the keys does not reveal any useful information about the other key [9]. The most popular method of key exchange used is Diffie-Hellman (DH). Figure 2 displays the flow of asymmetric-key cryptography. In general, this type of cryptography uses "mathematical functions that are easy to compute whereas their inverse function is relatively difficult to compute", also known as one-way functions [9]. The three main cryptography standards include Rivest-Shamir-Adleman (RSA), Digital Signature Algorithm (DSA), and Elliptic-Curve Cryptography (ECC).

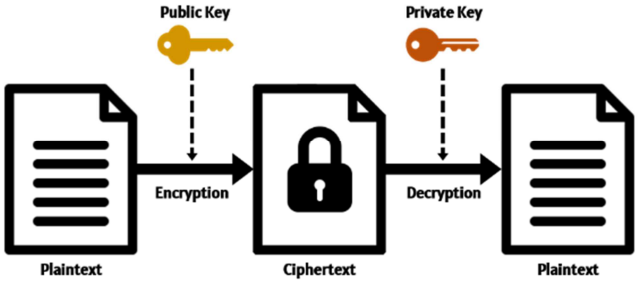

Figure 2. *Asymmetric-key Cryptography.*

DH is a key exchange protocol that enables two parties – a sender and a receiver – with no prior knowledge of each other to communicate securely over a public channel. The problem presented by DH that enables this communication to occur is as follows: Given g, g^a, and g^b, determine g^{ab} [14]. The following is an overview of how DH works [14]:

1. Alice and Bob agree on a finite cyclic group G and a generator g of G
2. Alice randomly chooses an integer, a, computes g^a, and sends the result to Bob
3. Bob randomly chooses an integer, b, computes g^b, and sends the result to Alice
4. Alice computes g^{ba} and Bob computes g^{ab}
5. Shared secret key g^{ab} is established.

RSA is one of the most common types of public-key cryptography that relies on the assumption that factoring large numbers is difficult for modern computers [15]. For two parties to communicate with one another, the receiver must first generate a pair of public and private keys and share the public key with the sender. The public and private keys are generated by the receiver as follows [15]:

1. Randomly choose two large prime numbers p and q, where $p \neq q$
2. Calculate $n = pq$. n is half of the public key.
3. Calculate the totient, $\emptyset(n) = (p-1)(q-1)$
4. Choose an arbitrary integer e such that $1 < e < \emptyset(n)$ and e is a co-prime to $\emptyset(n)$. e is the other half of the public key.
5. Compute d, where $de \equiv 1(mod\ \emptyset(n))$. d is the private key.

Having generated both keys, the receiver will then share the public key (n and e) with the sender. The sender will then proceed to do the following [15]:

1. Convert the message M into a number m such that $m < n$. M is the plaintext.
2. Compute c, where $c \equiv m^e (mod\ n)$. c is the ciphertext.

Once the ciphertext (c) has been generated, the sender will send it the receiver. The receiver will then decrypt the message as follows [15]:

1. Compute m, where $c^d \equiv m(mod\ n)$.
2. Convert the number m back into M

The typical size of the public and private keys is usually 1024-bits or 2048-bits; however, some keys can be 4096-bits long. The reason why these keys are so effective is because to determine the original prime numbers from the total product is far beyond any modern computer's capability, within a reasonable time frame. The one major flaw that exist with RSA is that the strength of the encryption is relies solely on the two random prime numbers chosen. If the numbers chosen are relatively small, the encryption becomes very weak, making it easier for someone eavesdropping to decrypt the data. To resolve this issue, many experts are favoring ECC over RSA, which will be discussed later. Common applications of RSA include web log-in sessions, electronic credit card payments, and mobile security.

DSA, on the other hand, is an algorithm that uses discrete logarithmic computations to generate the public key, private key, and a pair of large integers that act as signatures [16]. Having discussed in detail how the public and private keys are generated for RSA, the same will not be done for DSA as the only difference is which mathematical function was used to create the keys. The typical size of the public key, private key, and signatures range between 1024-bits and 2048-bits [16]. The purpose of the digital signatures is to verify that the data recovered has not been tampered with and that the correct recipient received the data. For DSA, the sender is the one to generate the public and private key. Data is encrypted through the use of a hash-based encryption scheme such as SHA, which will be discussed in detail later in this paper. The output is a hash of the plaintext, the ciphertext. The sender's private key is feed through an algorithm to generate a digital signature, which is appended to the ciphertext and sent to the receiver [16]. The receiver feeds the digital signature and the sender's public key into an algorithm to verify if the signature is valid [16]. If so, the verifier uses the same hash function to generate the plaintext. Therefore, the most common applications for DSA are those that need to guarantee the integrity and authenticity of data, such as e-mail privacy and electronic funds transactions.

ECC is another type of public-key cryptography that has been gaining popularity in recent years as it allows for the use of smaller key sizes than RSA, with increased security over RSA [17]. An elliptic curve is an algebraic curve consisting of a set of points commonly defined by Equation 3, but there are

other equations that can be used which will provide different levels of security [17].

$$y^2 = x^3 + ax + b \quad (3)$$

The shape of the elliptic curve is defined by the values of a and b, in which these parameters must satisfy the requirement presented in Equation 4. The purpose of this requirement is to ensure that the curve is non-singular – no self-intersections, isolated points, or cusps [17].

$$4a^3 + 27b^2 \neq 0 \quad (4)$$

A basic example of a curve can be seen in Figure 3. There are several key properties of an elliptic curve that are utilized by ECC. An elliptic curve is symmetrical over the x-axis, meaning that any point can be reflected over and remains on the same curve [17]. Therefore, if a point exists in the +y plane, then the same point will exist in the -y plane. In addition, a non-vertical line drawn on the curve will intersect the curve in at most three places [17]. As a result, adding or multiplying two points on an elliptic curve will yield a third point on the curve. Therefore, adding two points, A and B will yield a point C on the curve. Furthermore, any small changes in A or B can result in a large change in the position of C.

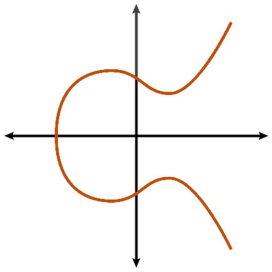

Figure 3. *An Elliptic Curve.*

The difficult problem that ECC is based on is as follows: Given two points, $\boldsymbol{A}$ and $\boldsymbol{C}$, on an elliptic curve, find an integer $\boldsymbol{n}$ such that $\boldsymbol{C} = \boldsymbol{nA}$ [17]. In solving this, first, an arithmetic operation on $\boldsymbol{A}$ must be defined, most commonly addition or multiplication, in which the properties of group law become important. A group is a set of points combined with an arithmetic operation to form a third point such that the four conditions – closure, associativity, identity, and invertibility – known as axioms are satisfied. For ECC, the public key is the value $\boldsymbol{C}$ and the private key is the value $\boldsymbol{n}$. A hacker might know the values of $\boldsymbol{A}$ and $\boldsymbol{C}$, but finding the value of $\boldsymbol{n}$ is difficult to solve without also knowing which operation was used. ECC is primarily used for key generation and paired with DH key-exchange protocol, known as Elliptic Curve Diffie Hellman (ECDH).

C. Hash-Based Cryptography

Hash-based cryptography is a method for encryption and decryption based on the effectiveness of cryptographic one-way hash function. A hash function is a mathematical algorithm that maps data (message) of arbitrary size, typically ranging between 256 to 512-bits, to a bit array of a fixed size (hash output) [18]. There are many different cryptographic hash functions which can be used, the most common of them fall under the Secure Hash Algorithm (SHA) family of hash functions (i.e. SHA-0, SHA-1, SHA-2, or SHA-3) [18].

Assume that the objective is to sign a message of bit length $\boldsymbol{n}$. The private key is created by generating two random bitstrings, each of bit length n, represented as a list. Equation 5 and 6 present the two random bitstrings X_o and X_1, respectively.

$$X_0 = X_0^0, X_1^0, \dots, X_{n-1}^0 \quad (5)$$

$$X_1 = X_0^1, X_1^1, \dots, X_{n-1}^1 \quad (6)$$

The public key is then created by hashing every single bit of the private key's list of random bitstrings. Equation 7 and 8 present the hash outputs of the two random bitstrings Y_o and Y_1, respectively.

$$Y_0 = H(X_0^0), H(X_1^0), \dots, H(X_{n-1}^0) \quad (7)$$

$$Y_1 = H(X_0^1), H(X_1^1), \dots, H(X_{n-1}^1) \quad (8)$$

To sign a n-bit message M, first, the message will need to be broken down to represent M as a sequence of n individual bits, as seen in Equation 9.

$$M_0, M_1, \dots, M_{n-1} \quad (9)$$

Bit-by-bit, the signature key, S, of bit length n is generated by selecting the bit from one of the two bitstrings representing the private key. This is done in the following way: if i is the bit index of message M, for $M_i = 0$, the i^{th} bit of X_0 is used to represent the i^{th} bit of the signature key, S_i. Furthermore, for $M_i = 1$, the i^{th} bit of X_1 is used to represent the i^{th} bit of the signature key, S_i. Having generated the signature key, the sender combines it to the message and sends it to the receiver.

The receiver, already having the public key can then verify the signature to verify that it is valid. This is done by hashing the signature key and using the message to compare the hash output with the public key. For $M_i = 0$, the hash $H(S_i)$ is calculated and compared to the i^{th} bit of Y_0. Furthermore, for $M_i = 1$, the hash $H(S_i)$ is calculated and compared to the i^{th} bit of Y_1. If every single bit of the hashed signature key matches that of the public key, then the signature is valid.

The one major limitation of this method is that the public and private key can only be used to sign one message, known as a one-time signature scheme (OTS). As a result, more complex digital signature schemes such as Merkle's Signature Scheme (MSS) – a one-time signature scheme with a Merkle tree structure – was developed. [18]. A Merkle tree is known as a binary hash tree. The benefit of MSS is that a Merkle tree structure with height h, having 2^h leaves, is used to systematically combine 2^h OTS keys under a single structure and therefore, allowing an effective way to sign 2^h messages [18]. Construction of a Merkle tree starts from the bottom and builds up. The leaf nodes of the tree consist of the public key pair (Y_0, Y_1), generated as the hash of the private key pair (X_0, X_1) [18]. The non-leaf nodes are the cryptographic hash of the two child nodes. The root, known as the Merkle root, is

considered to be the master public key, which is dependent on all the previous nodes in the tree and will be used to authenticate all the leaf nodes. Figure 4 illiterates a Merkle Tree's structure.

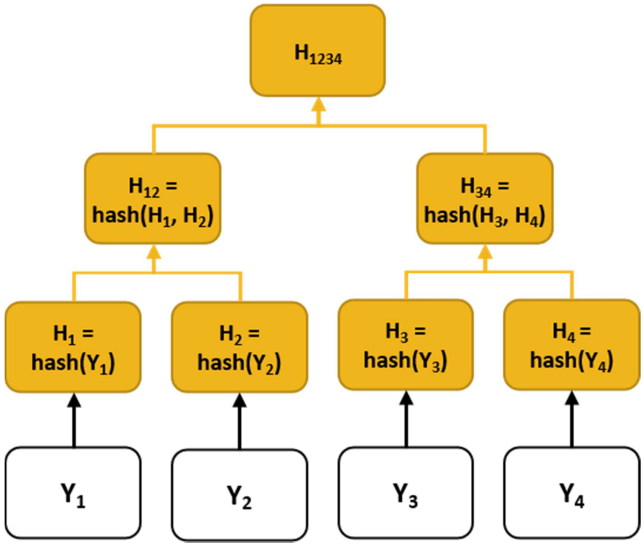

Figure 4. *A Merkle Tree Structure.*

Using MSS, the sender retains all the public and private keys used for signing the message. To sign a the k^{th} message, the signer selects the k^{th} leaf of the tree and uses the corresponding private key to create the signature key. To avoid reuse of keys, keys are used according to the order of the leaves, starting at the leftmost leaf. The sender then proceeds to combine the message, the signature key, and the Merkle proof – proving that the OTS public key being used is contained in the Merkle tree – together and sends the package to the receiver. After receiving the package, the receiver uses the public key to validate the signature and references the Merkle proof to ensure that the public key is contained within the Merkle tree, therefore belonging to the sender.

D. Chinese Cryptography

To be comprehensive, it is important to note that China, unlike the majority of the world, has its own set of cryptographic standards in which they mandate the use of for any technologies sold or services provided within their borders. Similar to the National Institute of Standards and Technology (NIST) in the United States of America, China's Office of Commercial Cryptography Administration (OSCCA) issues and regulates the country's commercial cryptographic standards. China's cryptographic standards are fundamentally the same as the cryptographic standards previously discussed in this paper, however, they have taken them a step further by adding additional complexity to allow for increased security. Overall, OSCCA has published five cryptography standards. This paper will not discuss these in detail but will provide a relation between Chinese cryptographic standards and globally reorganized cryptographic standards, as seen in Table 1.

Table 1. *A Comparison of Cryptography Standards.*

OSCCA Published Standards	Type of Cryptographic Standard	Globally Recognized Standards
SM2	Public-key Cryptography	ECC
SM3	Hash-Based Cryptography	SHA
SM4	Symmetric-key Cryptography	AES
SM9	Public-key Cryptography	ECC
ZUC	Symmetric-key Cryptography	AES

Although cybersecurity is not perfectly robust, several effective methods of cryptography exist today that successfully provide comprehensive security of valuable data. However, the standing on how secure these methods are when implemented is all contingent on the capabilities of modern computing. As more power computers become available, the efficacy of the existing methods could prove to be ineffective.

5. Cybersecurity after Quantum

Having discussed modern-day cybersecurity and the capabilities of quantum computing in comparison to modern-day computing, a comprehensive look can be done to determine how the field of cybersecurity, and more specifically cryptography, will be affected by quantum computing. Fundamentally, cryptography bases its security on the inability to devise an efficient solution to complex problems and furthermore, the inability for a person or machine to compute quickly enough to brute-force attack. Modern-day computers are quite fast, but even with their speed, cryptographic standards with large enough keys size will take decades if not centuries to break.

As previously mentioned, modern-day cryptography falls under three different categories: symmetric-key cryptography, public-key cryptography, and hash-based cryptography. Symmetric-key cryptography and hash-based cryptography rely on different forms of complex message manipulation to secure data. Public-key cryptography relies heavily on difficult mathematical problems to secure data. For the foreseeable future, all three of these cryptography strategies have the ability to remain secure through the use of sufficient key sizes. As computing power advances linearly, the keys can proportionally increase in size to add additional complexity. A simple solution and one that has been used for decades. However, with an exponential increase in computing power, this strategy will fall short as quantum computers enable the use of more efficient algorithms such as Grover's algorithm and Shor's algorithm.

A. Quantum Algorithm's

In 1996, Lov Grover, a computer scientist, devised a brute-force quantum algorithm capable of determining, with high probability, the unique input to a black box function that produces a particular output value, known as Grover's

algorithm [20]. Most commonly, this algorithm is known as a quantum searching algorithm with an efficiency of $O(\sqrt{N})$ when searching through an unsorted database of *N*-items, in comparison to a classical searching algorithm with a linear efficiency of $O(N)$ [20]. However, when discussing this algorithms application for black box functions, such as cryptography, Grover's algorithm can more accurately be described as a highly effective algorithm for inverting functions. As a result, Grover's algorithm when applied to any modern symmetric-key or hash-based cryptographic standard will drastically weaken its security. Symmetric-key and hash-based cryptography are black box functions, meaning that how data is encrypted and decrypted through data manipulation is forever unknown. What makes Grover's algorithm effective, but not robust, is the ability to brute-force attack; it can iterate through numerous key values with relatively high efficiency to determine the correct private key value. Therefore, the reason why Grover's algorithm weakens, but does not break, symmetric-key and hash-based cryptography is because without ever knowing how the data is manipulated, the only attack strategy is dependent on time. This means that the if the cryptographic standards were to become more complex, then Grover's algorithm would need more time to work. As a result, the strategy used to combat this attack could be to increase the key sizes of current cryptography standards, which will increase the time needed for Grover's algorithm to work.

In 1994, Peter Shor, a mathematician from the Massachusetts Institute of Technology (MIT), devised a quantum algorithm for solving integer factorization problems, known as Shor's algorithm [21]. It was formulated to solve the following problem: Given an integer *N*, find its prime factors. A well-devised algorithm that utilizes the quantum Fourier transform (QFT) for factoring and computing the discrete logarithm of real numbers with high efficiency [21]. As a result, Shor's algorithm when applied to any modern public-key cryptographic standard will break it, regardless of its key size. This is because Shor's algorithm aims to determine the solution to the complex mathematical problem presented by the different public-key cryptographic standards Once the solution has been obtained, the cryptographic standard is no longer effective. Therefore, the only way to combat this algorithm is to create much more difficult mathematical problems that would take Shor's algorithm exponentially more time to solve.

With the ability for quantum algorithms to comprise the security of modern-day cryptography, two new fields of cryptography have emerged in the pursuit for greater security, which include quantum cryptography and post-quantum cryptography.

B. Quantum Cryptography

Quantum cryptography aims to utilize the theories of quantum physics to create cryptosystems that are resilient against quantum computers. Currently, the most robust and popular quantum protocol is Quantum Key Distribution (QKD), which utilizes concepts from the Heisenberg's uncertainty principle and the no-cloning theorem to allow for two parties to communicate together securely over an unsecure channel [22]. In short, Heisenberg's uncertainty principle states that the position and momentum of a particle cannot be simultaneously measured with high precision. The no-cloning theorem states that "it is impossible to create an independent and identical copy of an arbitrary unknown quantum state" [23].

QKD is not used for encryption or data transmission, but rather for key generation and distribution. The benefit of QKD is that keys are generated randomly, meaning that there is no possible way of determining any unique characteristics from the key itself [22]. Keys are then encoded bit-by-bit over single photons and transmitted as a stream of photons via a quantum channel, such as fiber-optic or free space optics (FSO) [22]. A hacker trying to intercept the photon stream in the quantum link will be unsuccessful. This is because any interruption or modification to the photons will alter the encoded state of the photon and therefore, causing detectable error. Using the quantum entanglement and superposition, the sender and receiver can set up a system to detect eavesdropping over the quantum channel. Based on the level of error that was caused by eavesdropping, the two parties can determine if the key has been compromised. If so, the sender and receiver can terminate their communication. The one major hurdle of QKD is that photon transmission is limited to approximately 60 miles, in which a network of trusted nodes needs to be created to allow keys to be shared over long distances and with multiple users. The two main protocols of utilizing QKD include BB84 and E91.

Data, on the other hand, is encrypted using any form of quantum-resilient cryptographic standard and transmitted over the classic channel. A hacker will be able to intercept the message, but with a randomly generated key and no viable way to intercept it, decrypting the message will be virtually impossible.

C. Post-Quantum Cryptography

This field of cryptography builds upon the current cryptographic standards by researching more complex data manipulation techniques and more difficult mathematical problems that are resilient against quantum algorithms. There are many different approaches to cryptographic standards that are being considered, all which look promising, but are still in their infant stages of development. These standards include lattice-based cryptography, extended hash-based cryptography, supersingular elliptic curve isogeny cryptography, and code-based cryptography.

Lattice-based cryptography presents a series of problems involving lattices. A lattice (L) is any regular spaced set of points (P) in an n-dimensional space with a periodic structure as seen in Figure 5 [24]. For the sake of further discussion, a basis (B) is a small collection of vectors in a lattice. The same lattice can have multiple basis. The assumption that led researchers down this path of exploration was that lattice problems are very difficult to solve and therefore, would be resilient against quantum computers. To this day, this assumption is upheld as there is currently no quantum algorithms for solving lattice problems with the efficiency required that would compromise security. Several attempts dating back to the 1990s have been made to solve lattice

problems through Shor's algorithm or other related quantum algorithms; however, little success has been made [24]. A few of the most popular lattice problems being considered for use in the field of cryptography are as follows:

1. Shortest Vector Problem (SVP): Find the shortest nonzero vector in L as close as possible to the origin.
2. Closest Vector Problem (CVP): Given a point P not in L, find a vector in L that is closest to P.
3. Shortest Independent Vectors Problem (SIVP): Given a lattice L of dimension n, find n linearly independent vectors $v_1, v_2, \dots, v_n$ such that $\max\|v_n\| \leq \lambda_n$, where λ_n is the nth successive minimum of L.

It is important to note, when discussing lattice problems, that lattices can be multi-dimensional in which the complexity of the lattice problem increases as the dimensions of the lattice increase.

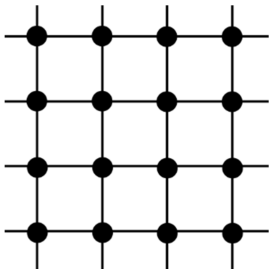

Figure 5. *A two-dimensional lattice.*

eXtended Merkle Signature Scheme (XMSS) builds upon the Merkle Signature Scheme (MSS) discussed earlier as a form of quantum-resilient hash-based cryptography. A core difference between XMSS and MSS is that XMSS uses Winternitz one-time signature (WOTS), a special implementation of OTS with the intention to minimize space with the tradeoff of time. To sign messages, OTS keys focus on the comparison of individual bit in which the private and public key are comprised of a pair of random bitstrings and their hash, respectively [19]. WOTS, on the other hand aims to sign the message using a large subset (i.e., four-bit nibbles or eight-bit bytes) [18]. Furthermore, to create the set of private keys, WOTS only generates a single random bitstring and proceeds to hash it repeatedly to generate the additional bitstrings required. Equation 10 and 11 presents the two bitstrings $\boldsymbol{X_o}$ and $\boldsymbol{X_1}$, respectively, representing the private key subsets for WOTS.

$$X_0 = X_0^0, X_1^0, \dots, X_{n-1}^0 \tag{10}$$

$$X_1 = X_0^1, X_1^1, \dots, X_{n-1}^1$$

$$= H(X_0^0), H(X_1^0), \dots, H(X_{n-1}^0) \tag{11}$$

For WOTS, the public key can be generated as a final hash of the last private key bitstring generated. This, in turn, requires the storage of only the one initial random bitstring created, but will require the computation of the additional bitstrings to be done as needed, therefore, sacrificing time. This method, however, comes with a major vulnerability. Because the private keys and public key are all related, anyone who has access to the original message and signature key can manipulate the data as a technique of forgery [18]. To resolve this, the sender will need to calculate and sign the checksum of the original message. "The structure of this checksum is designed to prevent the attacker from incrementing any of the bytes, without invalidating the checksum" [18]. Overall, the benefit of this XMSS is that it is simple to implement, resists side-channel attacks, and is suitable for compact implementations.

Supersingular Isogeny Diffie Hellman (SIDH) is a quantum robust key-exchange protocol that builds upon the concepts of ECDH discussed earlier. The added complexity from ECDH is provided through isogeny – a morphism of algebraic groups [26]. With modern-day ECC, the security stemmed from the difficultly to determine the integer value n, given two points on the same elliptic curve, A and C, such that $C = nA$. ECC isogeny conceptually is similar, with some added complexity. Suppose there exist two elliptic curves, E1 and E2. A function is created to map a point A in E1 to a point C in E2. This function is an isogeny. Figure 6 provided an overview of ECC isogeny.

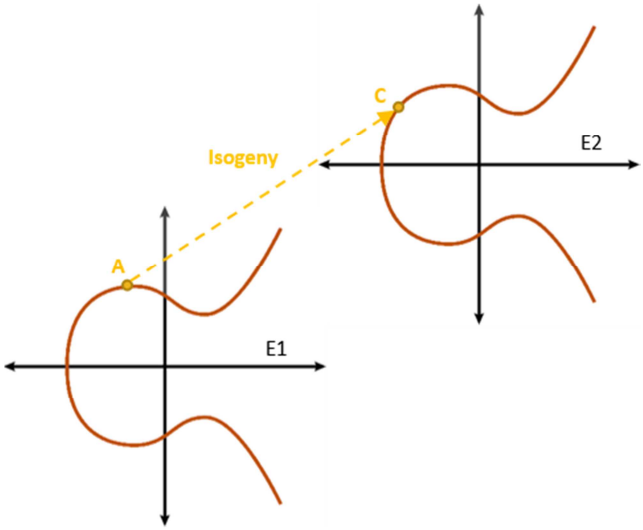

Figure 6. *Overview of ECC Isogeny.*

For SIDH, the public key is the elliptic curve. The shared secret between the sender and receiver is the isogeny function. For key exchange, the sender and receiver mix their isogeny function with the public key to create a secret curve [26]. SIDH in the field of quantum-resilient cryptography has received much praise, as it is believed to have perfect forward secrecy.

Code-based cryptography is a method of encryption and decryption based on error-correcting code (i.e., Goppa Code) [27]. Error-correcting code is an algorithm for detecting and correcting errors in data caused by *noise* from an unreliable communication channel. Error-correcting code was initially intended as a method of data recovery; however, in the exhaustive hunt for quantum-resilient cryptography, it was discovered that noise and noise-removal algorithms could be used as a method for encryption and decryption. This idea extended the field of cryptography into coding theory,

resulting in development of several promising cryptosystems: two of them being McEliece cryptosystem and Niederreiter cryptosystem. McEliece cryptosystem is a public-key cryptosystem based on the idea of intentionally adding "random" errors to the code as a method of encryption. Its security relies on the hardness of decoding a linear code. Goppa Code is currently the most favored decoding algorithm, as it has a very efficient. Goppa Code is a type of error-correcting code based on modular arithmetic, "which is when a series of numbers increases towards a certain number and upon reaching said number, starts back over at 0 again" [27]. Below is an overview of how McEliece cryptosystem works. The following steps summarize how keys are generated [27]:

1. Alice selects an error-correcting code capable of correcting t errors
2. Let G be a generator matrix multiplied by a scrambler matrix, S, and a permutation matrix, P, Alice computes $\hat{G} = SGP$
3. Alice's public key is $(\hat{G}, t)$ and private key is (S, G, P)

The following steps summarize encryption [27]:

1. Bob encodes a message, m, using $\hat{G}$ from Alice's public key
2. Bob couples the encoded message with a vector, z, containing t amount of errors
3. Bob sends the following: $c = \hat{G}m + z$

The following steps summarize decryption [27]:

1. Alice computes $\hat{c} = cP^{-1}$
2. Alice uses an error-correcting code to decode $\hat{c} \rightarrow \hat{m}$
3. Alice computes $m = \hat{m}S^{-1}$

Niederreiter Cryptosystem is another popular type of code-based cryptography, which is a variation of McEliece Cryptosystem that uses a parity check matrix for encryption instead of a generator matrix [28]. The favoritism towards this cryptosystem over McEliece cryptosystem is due to its efficiency, which has an encryption speed 10 times faster than McEliece [28].

6. The Impact of Quantum

Current predictions estimate that quantum computers will be established before organizations and government agencies are prepared to handle quantum-based cyber-attacks, which will result in a short period of vulnerability. Figure 7 displays a graphical timeline of how soon these organizations will need to prepare [29]. Based on Figure 7, three critical questions can be derived. First, what is the shelf life of the information currently in need of protection – x years? If the information in question will go out-of-date around the time that quantum computers become established, then it may not be necessary to enhance the security protecting that information. Second, what is the migration time of the system – y years? More so, how long will it take to deploy a system that is resilient against quantum-based cyber-attacks. Depending on the complexity of the system, this process could take anywhere from a couple of years to over a decade. Third, what is the collapse time in which the traditional methods of cybersecurity will no longer be effective – z years? Based on the answers to these three questions, it is crucial that the outcome abides by the following rule: $x + y < z$ [29]. If not, then for every day or month or year that $x + y$ exceeds z is the time in which sensitive information and the infrastructure that protects it become vulnerable to cyber-attacks. More critically, if $y > z$, then the entire system will collapse, which will leave very little room for recovery. Therefore, by planning for the future and making sure that the plan abides by the rule presented in this paragraph, organizations can ensure that their information is secure from harmful cyber-attacks.

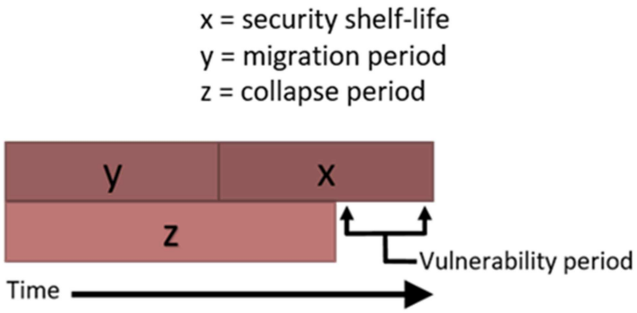

Figure 7. *Diagram of Future Planning.*

Currently, most industries use a combination of different cryptographic algorithms, known as cipher suites, to protect their networks. Table 2 displays the top three cipher suites used per industry sector. Through interpretation of the table, it can be seen that the most popular cipher suite being used is ECDHE-RSA-AES256-GCM-SHA384 [30]. ECDH and RSA, which are public-key cryptographic standards, are used for key generation and exchange. AES GCM – AES with Galois/Counter Mode –, a symmetric-key cryptographic standard, is used for encryption and SHA is used for hashing [30]. Based on what was discussed previously, it is understood that all these standards, although highly effective now, are vulnerable to future quantum cyberattacks. Therefore, it is no question that these industries, and many others, will need to invest time and resources into shifting their security infrastructure to become more quantum resilient. The question is how long these industries will have before it is too late.

Table 2. *List of Different Industries and their Most Used Cipher Suites [30].*

Industry	1st Cipher	2nd Cipher	3rd Cipher
Health	ECDHE-RSA-AES256-GCM-SHA384	ECDHE-RSA-AES128-GCM-SHA256	ECDHE-RSA-AES256-SHA384
News	ECDHE-RSA-AES256-GCM-SHA384	ECDHE-RSA-AES128-GCM-SHA256	ECDHE-RSA-AES256-SHA384
Retail	ECDHE-RSA-AES256-GCM-SHA384	ECDHE-RSA-AES128-GCM-SHA256	ECDHE-RSA-AES256-SHA384
Technology	ECDHE-RSA-AES128-GCM-SHA256	ECDHE-RSA-AES256-GCM-SHA384	ECDHE-ECDSA-AES128-GCM-SHA256
Business	ECDHE-RSA-AES256-GCM-SHA384	AES256-SHA256	ECDHE-RSA-AES128-GCM-SHA256

It is quite difficult to determine when quantum computers will achieve quantum supremacy and begin to pose a serious threat to the current security infrastructure. However, based on the trend of progress made, an understanding can be obtained on how fast this technology is advancing and how soon the industry can expect to see a breakthrough in this space. Table 3 provides a list of achievements throughout the past three decades towards quantum computers.

Table 3. *Quantum Computer Achievements.*

1985	David Deutsch describes the first universal quantum computer.
1998	First working 3-qubit NMR computer. First execution of Grover's algorithm.
2000	First working 7-qubit NMR computer demonstrated. First execution of Shor's algorithm.
2006	First working 12-qubit NMR computer.
2007	D-Wave Systems demonstrates use of a 28-qubit quantum annealing computer.
2008	D-Wave Systems claims to have produced a 128-qubit computer chip.
2009	First universal programmable quantum computer unveiled.
2011	D-Wave One introduced as the first commercially available quantum computer.
2012	1QB Information Technologies (1QBit) founded as world's first quantum computing software company.
2015	D-Wave Systems announced that it had broken the 1000 qubit barrier.
2017	D-Wave Systems Inc. announces general commercial availability of the D-Wave 2000Q quantum annealing computer. Microsoft reveals an unnamed quantum programming language. Programs can be executed locally on a 32-qubit simulator. Intel confirms development of a 17-qubit superconducting test chip. IBM reveals a working 50-qubit quantum computer.
2018	Google announces the creation of a 72-qubit quantum chip, called "Bristlecone" Intel confirms development of a 49-qubit superconducting test chip
2019	IBM announces the world's first commercially available integrated quantum computer, Q System One

Based on the information presented in Table 3, a number of trends can be observed. The idea that a quantum world exist beyond the physical word was first brought to light in the mid-1920s. It was not until the mid-1980s, however, where David Deutsch theorized that quantum physics can be applied to computers as a method to enhance its performance; however, it was not until the late 1990s in which this theory was applied. Up to end of the 20th century, the progress was slow but was monumental in laying down the groundwork for future innovation. In the turn of the century, quantum computers began to pick up significant traction and major advancements began to take form. Up until 2006, there was much work being done with no major milestones that deemed significant. However, nearly every year following produced major accomplishments leading up to 2019 where IBM revealed the first every working 50-qubit quantum computer. This development, although in its infant stages, was crucial as it becomes the first quantum computer with the computing power to satisfy the theory of quantum supremacy. Given the current trend outlined in Table 3, if progress continues to grow at the same rate, it is only a matter of a few years to a decade before quantum computers begin to surpass modern-day computers.

Given this information, it would be alarming to find out that within the next decade, the security infrastructure currently in place will be compromised. However, the information previously provided does not paint the whole landscape as there are numerous hurdles to be overcome in order for quantum computers to be reliable and achievable. To achieve quantum properties, the Niobium (Nb) needs to be cooled down to extreme levels, ideally equivalent to or colder than the temperature of outer space (-455°F). The environment to which the computer resides in must be kept constant as any noise or change in temperature will disrupt the qubits' entangled state, causing significant errors in calculation, known as quantum decoherence [31]. As of now, to properly operate a quantum computer, a highly skilled team of mathematicians, physicists, and engineers are required as well as highly advanced, state-of-the-art equipment, costing hundreds-of-millions to billions of dollars. A situation in which the majority of organizations and governments are not able to acquire, let alone a common hacker.

Furthermore, several hurdles exist that need to be overcome in order for quantum computers to be effective in preforming reliable computations. One such hurdle is based on the properties of collapse, outlined in the *Copenhagen Interpretation*. It is understood that measuring a state of a qubit will reveal a random state of the particle's infinite stats. For calculation, the question becomes, how would one know if the collapsed state is the correct answer? If not the correct state, which state is and how long will it take to collapse to that correct state? As a result, robust algorithms still need to be developed to filter through all possible states and yield a single correct answer. Similar to the field of cybersecurity, there is still much progress to be made before the capabilities of quantum computers can be effectively utilized.

7. Conclusions and Recommendations

The field of cybersecurity has evolved significantly over the past five decades and has been highly effective in combatting cyber-attacks derived from modern-day computers. However, as quantum computers continue to advance, eventually the current security infrastructure will not be sufficient.

Modern-day cryptography bases its security on the inability to devise an efficient solution to complex problems and furthermore, the inability for a person or machine to compute quickly enough. With the development of quantum computers, both of these will be resolved through the implementation of quantum algorithms, including Shor's algorithm for solving integer factorization

problems, breaking public-key cryptography, and Grover's algorithm as an effective algorithm for brute force-attacks, weakening symmetric-key cryptography.

As a result, more complex cryptography solutions are required to be resilient against quantum based cyber-attacks. This has led to the creation of quantum cryptography, which utilizes the concepts of quantum physics to create cryptosystems that are resilient against quantum computers, and post-quantum cryptography, which builds upon the current cryptographic standards by researching more complex data manipulation techniques and more difficult mathematical problems that are resilient against quantum algorithms.

Given the current security landscape of most industries and the development trend of quantum computers, it is imperative that major strides are taken in the near future to create more quantum-resilient infrastructures. The transition will be difficult and costly, however if not done in time, the result of a quantum based cyber-attack on a major organization of government will be crippling.

References

[1] D. Murphey, "A History of Information Security," 03-Jul-2019. [Online]. Available: https://www.ifsecglobal.com/cyber-security/a-history-of-information-security/#:~:text=Cybersecurity's history began with, small trail wherever it went. [Accessed: 20-Nov-2020].

[2] "Transistor," *PCMAG*. The Computer Language Co Inc.

[3] B. Pawliw, "What is quantum computing?" *WhatIs.com*, 08-Jun-2020. [Online]. Available: https://whatis.techtarget.com/definition/quantum-computing. [Accessed: 04-Dec-2020].

[4] "Copenhagen Interpretation." [Online]. Available: http://abyss.uoregon.edu/~js/21st_century_science/lectures/lec15.html. [Accessed: 01-Dec-2020].

[5] "Quantum Computing 101," *Institute for Quantum Computing*, 16-Oct-2020. [Online]. Available: https://uwaterloo.ca/institute-for-quantum-computing/quantum-computing-101. [Accessed: 04-Dec-2020].

[6] A. Z. Jones, "Everything You Need to Know About Bell's Theorem," *ThoughtCo*, 02-Apr-2018. [Online]. Available: https://www.thoughtco.com/what-is-bells-theorem-2699344#:~:text=Bell's Theorem was devised by, than the speed of light. [Accessed: 04-Dec-2020].

[7] J. Desjardins, "The 3 Types of Quantum Computers and Their Applications," *Visual Capitalist*, 13-Mar-2020. [Online]. Available: https://www.visualcapitalist.com/three-types-quantum-computers/. [Accessed: 04-Dec-2020].

[8] M. Giles and W. Knight, "Google thinks it's close to 'quantum supremacy." Here's what that really means.," *MIT Technology Review*, 31-Aug-2020. [Online]. Available: https://www.technologyreview.com/2018/03/09/144805/google-thinks-its-close-to-quantum-supremacy-heres-what-that-really-means/. [Accessed: 04-Dec-2020].

[9] G. C. Kessler, "Types of Cryptographic Algorithms," *An Overview of Cryptography*, 30-Nov-2020. [Online]. Available: https://www.garykessler.net/library/crypto.html#:~:text=Three types of cryptography: secret, public key, and hash function. [Accessed: 01-Dec-2020].

[10] National Institute of Standards and Technology, 1999. *Data Encryption Standard (DES)*. Information Technology Laboratory.

[11] Singh, G., & Supriya, S. (2013, April 18). A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. *International Journal of Computer Applications, 67* (19), 33-38. doi: 10.5120/11507-7224

[12] National Institute of Standards and Technology, 2001. Announcing the Advanced Encryption Standard (AES). Information Technology Laboratory.

[13] G. J. Simmons, E. Gregersen, P. Jain, and W. L. Hosch, "AES." Britannica, 22-Jul-2009.

[14] J. Buchmann, *Introduction to Cryptography*, 2nd ed. Springer-Verlag New York, 2004.

[15] A. Katz, A. Ng, P. Bourg, E. Ross, and A. Kau, "RSA Encryption," *Brilliant Math & Science Wiki*. [Online]. Available: https://brilliant.org/wiki/rsa-encryption/. [Accessed: 04-Dec-2020].

[16] National Institute of Standards and Technology, n.d. *Digital Signature Standard (DSS)*. Gaithersburg, MD: Information Technology Laboratory.

[17] G. C. Kessler, "Cryptographic Algorithms in Action," *An Overview of Cryptography*, 30-Nov-2020. [Online]. Available: https://www.garykessler.net/library/crypto.html#:~:text=Three types of cryptography: secret, public key, and hash function. [Accessed: 04-Dec-2020].

[18] M. Green, "Hash-based Signatures: An illustrated Primer," *A Few Thoughts on Cryptographic Engineering*, 18-Apr-2018. [Online]. Available: https://blog.cryptographyengineering.com/2018/04/07/hash-based-signatures-an-illustrated-primer/. [Accessed: 05-Dec-2020].

[19] D. Li and Y. Liu, "Introduction to the Commercial Cryptography Scheme in China," in *atsec China*, 06-Nov-2015.

[20] Grover, L., n.d. *A Fast Quantum Mechanical Algorithm For Database Search*. [online] Murray Hill, NJ: Bell Labs. Available at: <https://arxiv.org/pdf/quant-ph/9605043.pdf> [Accessed 4 December 2020].

[21] "Shor's Algorithm," *IBM Quantum Experience*. [Online]. Available: https://quantum-computing.ibm.com/docs/iqx/guide/shors-algorithm. [Accessed: 04-Dec-2020].

[22] n.d. *What Is Quantum Key Distribution?* [ebook] Cloud Security Alliance. Available at: <https://www.quintessencelabs.com/wp-content/uploads/2015/08/CSA-What-is-Quantum-Key-Distribution-QKD-1.pdf> [Accessed 4 December 2020].

[23] "The no-cloning theorem," *Quantum Information Portal and Wiki*. [Online]. Available: https://www.quantiki.org/wiki/no-cloning-theorem#:~:text=The no cloning theorem is, quantum computing and related fields. [Accessed: 04-Dec-2020].

[24] Micciancio, D. and Regev, O., 2008. *Lattice-Based Cryptography*. [online] New York City, NY: New York University. Available at: <https://cims.nyu.edu/~regev/papers/pqc.pdf> [Accessed 4 December 2020].

[25] Huelsing, A., Butin, D., Gazdag, S., Rijneveld, J. and Mohaisen, A., 2018. *XMSS: Extended Merkle Signature Scheme*. [online] Internet Research Task Force (IRTF). Available at: <https://tools.ietf.org/html/rfc8391#section-3> [Accessed 4 December 2020].

[26] B. Buchanan, "Supersingular Isogeny Diffie-Hellman (SIDH) for Post Quantum Computer Key Generation," *Medium*, 22-Mar-2020. [Online]. Available: https://medium.com/coinmonks/supersingular-isogeny-diffie-hellman-sidh-for-post-quantum-computer-key-generation-6742d2ea78dc. [Accessed: 04-Dec-2020].

[27] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, 5th ed. London: CRC Press, 2001.

[28] Wang, W., Szefer, J. and Niederhagen, R., n.d. *FPGA-Based Niederreiter Cryptosystem Using Binary Goppa Codes*. [online] New Haven, CT, USA: Yale University. Available at: <https://eprint.iacr.org/2017/1180.pdf> [Accessed 4 December 2020].

[29] C. Moskowitz, "How Quantum Computing Could Change Cybersecurity Forever," *Scientific American*, 05-Oct-2016. [Online]. Available: https://www.scientificamerican.com/article/how-quantum-computing-could-change-cybersecurity-forever-video/. [Accessed: 04-Dec-2020].

[30] William J. Buchanan, Alan Woodward & Scott Helme (2017) Cryptography across industry sectors, Journal of Cyber Security Technology, 1: 3-4, 145-162, DOI: 10.1080/23742917.2017.1327221

[31] "The Role of Decoherence in Quantum Mechanics," *Stanford Encyclopedia of Philosophy*. Center for Study of Language and Information, Stanford University, 03-Nov-2003.