Quantum security

By Scott Vore
Department of Information and Computer Science, ICS 495
The University of Hawaii at Manoa, Honolulu, HI

***Abstract*– IBM has coined the term "the quantum decade" to describe their predictions for the 2020s. We are in the early stage of quantum supremacy, and this has caused an influx of media coverage on global cryptographic protocols being broken by quantum computers. The following article is my journey into quantum computing and some of my motivations for why I am interested in this field of computer science. I also cover computer security topics and blockchain technology, as the next evolution of the internet uses blockchains as the foundation for security and scalability. There are no new contributions through experimentation in this article. I aim to provide clarity to areas I found confusing in research and new perspectives to learning complex topics. Quantum computing is the most challenging field in STEM as it is the interdisciplinary field of computer science and quantum physics. The initial motivations for researching concepts around blockchain technology and quantum computing were to think about quantum blockchains to secure all facets of online activity from quantum cryptanalysis. However, the research rapidly evolved into learning quantum information science and technology. I hope to continue this line of research in the future as problems do not simply 'go away' when not actively observed.**

Keywords– Computer security, blockchain technology, cryptocurrency, quantum computing, quantum cryptography, zero knowledge proof

I. Introduction[1]

Fourscore and six years ago, Alan Turing invented a system of computation that laid the foundation for modern computers [1]. He named this invention the automatic machine, later coined the Turing Machine, and the developments and innovation that followed is the culmination of human knowledge and research. Turing machines are the pencil-and-paper equivalent of central processing units (CPUs), and the evolution into the powerhouse computers we have today is astounding. Soon after 1936, Turing brought his concept to physical reality and used it to crack Enigma during WWII. Following this, the computer went through several significant transitions. First was the mainframe era, where computers were tremendously large [2]. Only governments and large corporations used this initial version of computers, but universities began housing these mammoth computers soon after. The computer went through another transformation during this time, where the large mainframe acted as a host to multiple end-systems on a wired network. The next major innovation was ALOHAnet in 1969, which pioneered wireless data transmission. The following decades further enhanced the computer with semiconductors to condense large mainframes into personal computers.

Computers until this point were only used by nerds and geeks[2], but this would quickly change with the World Wide Web, created in 1990. The first website went live on August 6, 1991, but web development was in its very early stages [3]. It took about 15 years before the widespread usage of the internet took over the world. Alongside the rapid growth of the internet, cellular devices saw spectacular innovation to bring the internet to our fingertips. In 2007, Steve Jobs at Apple Inc. rocked the world with the first touch-screen exclusive smartphone, the iPhone I. The history of computers is quite remarkable, and during the same period, quantum physics saw maturity from quantum theory to quantum computers.

In the 1980s and 1990s, quantum information science saw groundbreaking advances in the possibility of computation that can utilize quantum mechanical properties to perform computations [4]. Some of these breakthroughs include using Turing Machines that conform to the behavior of Schrodinger's equation, the Deutsch-Joza algorithm, Grover's algorithm, and Shor's algorithm. These realizations of quantum computing rocked the computer science world as to the capabilities of quantum computers.

Shor's algorithm mainly brought increased media attention to quantum computing as it outlined a method for solving prime factorization in O(log(n)) time. This formulation started the quantum advantage movement, commonly referred to as quantum supremacy. Many network security protocols rely on asymmetric key cryptography to provide encrypted communication, but many of these algorithms use prime factorization for the key to decrypting messages. For example, Transport Layer Security (TLS) for HTTPS-based connections utilizes asymmetric key cryptography for the initial handshake [5]. The client and server calculate a shared secret through modular arithmetic. Thus, with a powerful quantum computer in the wrong hands, the world's network security infrastructure may be broken in a mere fraction of the time it would take with a classical computer. Now that we are in the beginning stages of quantum supremacy, we face a common problem of asking

1 Some pre-requisite knowledge for this paper include linear algebra, discrete mathematics, and foundational knowledge of algorithms. Knowledge of security concepts and cryptography will further aide in digesting this article.

2 A more politically correct term is computer scientist and computer enthusiast respectively, but if you are reading this, then you are some linear combination of nerd and geek.

ourselves ‘can we’ instead of thinking about the consequences of our actions. Now that Pandora is out of her box, computer scientists must push computers beyond this hurdle to uphold the lifestyle we rely on so heavily.

II. Security

Cybersecurity is the most challenging field in computer science. Cybersecurity professionals must have breadth and depth of knowledge in all areas of computer science, as everything relates to security in cyberspace. Cybersecurity is often described as a never-ending race to a finish line that is always on the move, which is an understatement for the complexity of the security landscape.

There are many problems to solve in the security world, but none are more important than password security. Password security is crucial to maintaining privacy, authenticity, and availability of information technology. Most web applications require user account passwords, and connecting to these services can be cumbersome. Furthermore, these passwords are inputted by the user and sent over a network that allows attackers to gain unauthorized access to confidential information. However, this is just one aspect of the whole problem. User-provided passwords are arbitrary, but many security protocols follow strict algorithmic formulations to ensure the protocol is accessible to all systems.

A. Why Security is Complex

Cyberspace has reached a point where world powers can wage war through ones and zeros. An attacker halfway around the world can cripple a nation’s infrastructure, which is only the beginning. The space domain has again become the forefront of a cold war in the modern world. Military personnel can now conduct operations without leaving their nation but inflict unfathomable civilian casualties and incalculable environmental consequences through cyberattacks.

The intricacies of cybersecurity impact both the world stage and the individual level. One priority for security protocols and securing software is portability to legacy systems. After all, it would be costly to require new hardware every time a piece of software is updated. Thus, it is not enough to learn math and algorithms to succeed in cybersecurity; one must also learn how all computational devices operate. However, new technology is constantly in development, so security professionals dedicate their lives to protecting everyone’s information. Furthermore, one piece of software can provide many potential attack pathways into an unknowing system.

The internet is moving away from the wild-west era, and cybersecurity professionals are tasked with securing established software while planning for the future. The Hyper-Text Transport Protocol (HTTP) is the backbone of the internet, but the first iteration would send all network traffic in plain text [3]. Thus, an attacker can capture packets between two end-systems and read everything being sent bidirectionally. The next evolution, Web 2.0, was the addition of the TLS handshake to HTTP, named HTTPS. This added layer of security encrypts each packet as it travels between connection points in the network. However, TLS is established between each connection, so packets are only encrypted when actively sent over a network, but at each resource point, it is decrypted. While TLS made it harder for attackers to analyze network traffic, sophisticated tools can crack TLS data.

Web 2.0 also introduced a new problem to the internet, centralization. Centralization refers to “Big Tech,” where private organizations control their software's logging and security procedures. These policies lead to questionable business practices from the big tech companies to draw traffic away from competitors and keep users active for extended periods, regardless of their experience [6]. This only scratches the surface of the problem, but current developments in web3 aim to decentralize the internet, starting with a foundation in blockchain technology.

B. Blockchain Technology

Blockchain technology has become a buzzword since the creation of Bitcoin. The blockchain uses cryptographic algorithms to store valid transactional information on a shared, immutable ledger [7]. Blockchains act as a public logging system to track assets and transactions (txns) tied to the smart contract. The txns are placed in blocks that get added to the blockchain with a block cipher algorithm, and each block has a unique hash to address each block. The blocks consist of four header fields, the previous block’s hash, transactional details, a nonce, and the current block’s hash. These blocks are then added to the smart contract’s Merkle tree, also known as rollups, so finding the block can be done in O(log(n)) time as opposed to O(n) time. The addition of new blocks is calculated by an arbitrary party known as miners. For many cryptocurrencies, miners provide computational power to mint new coins and secure txns. The new coins are added to the liquidity pool for purchase, and new txns are added to the blockchain.

The blockchain provides decentralization by being accessible to anyone that can connect to the network. Anyone can trade assets, mine blockchain platforms, and validate new txns. Miners compete to solve the hash function first, as most smart contracts allow only one person to reap the rewards. However, the miner can only get the reward once their calculations are correctly validated. This validation procedure is called consensus [8]. Consensus algorithms focus on the security and efficiency of validating txns on the blockchain. Proof of work (PoW) is the first consensus protocol, which is done to verify that the miner did the work required to add a new block to the chain. New cryptocurrencies are experimenting with other forms of consensus, such as proof of stake (PoS) and proof of burn (PoB), to address some of the downsides of PoW.

Cryptomining on the PoW protocol is expensive. It requires extremely powerful machines to calculate the hashes in the competitive environment. All other miners who did not solve the hash use a lot of energy and get no rewards. Energy consumption is further expended when the successful miner has their work

verified. This verification is performed by a zero-knowledge proofing system where no knowledge about the answers is communicated between the prover (the miner) and the verifier (a decentralized third-party).

1) Zero-Knowledge Proof (ZKP)

ZKPs are not a new concept. The first formalized writing about ZKPs was in 1985 in [9]. The researchers outlined a system of proving the decidability of NP problems in P time. A point to note is that this does not imply a proof of P versus NP. ZKPs are simply a system for checking the correctness of a solution to a given problem in polynomial time. This notion is further developed by the criterion established in [9]. "[A] proof system for L is zero-knowledge if, for each x$\in$ L, the prover tells the verifier essentially nothing, other than x$\in$ L," is the first defining factor of ZKPs taken directly from the paper. In essence, the proof system cannot prove false theorems and requires the minimum amount of information about the answer for each iteration of the proof. The last criterion relies on the trust of both the prover and verifier. This system has inherent flaws because it is a statistical analysis, where there will never be a 100% guarantee that a bad actor can accurately predict the outcome of each iteration. This also explains why ZKPs are not proofs of P versus NP, as this proof must be simulatable with a deterministic Turing machine. This original paper went through many revisions and spawned numerous continuations by the researchers involved. This team also won the first Gödel's Prize in theoretical computer science in 1993.

A subsequent paper co-authored by Oded Goldreich and Yair Oren titled "Definitions and Properties of Zero-Knowledge Proof Systems" adds more formalizations to how ZKPs interact with other ZKPs. ZKPs can be classified into two categories, auxiliary-input and black-box ZKPs. However, these categories are strictly for understanding the two types of ZKPs. Both sets, along with any intersection, fall within the open set provided by the original ZKP formalism. Thus, ZKPs can encompass both I/O and overhead systems.

a) Intuitive approach to ZKPs: A common way to conceptualize this system is with the following scenario: Bob is colorblind [10]. Alice, Bob's friend, wants to prove that there are different colors and devises a plan to do so. Alice hands Bob two objects of different colors and tells him to put them behind his back. He can choose to swap them or leave them in their current positions. Afterward, Bob moves the objects in view, and Alice will state whether Bob switched them or left them alone. Bob has no way of knowing if Alice is telling the truth about there being more colors but repeating this experiment, Bob realizes that Alice is not lucky; she knows something that he does not. Since Bob has no knowledge of color, Alice can use her knowledge to tell Bob what action he performed but does not share the precise answer. This explanation of ZKPs is famous as it portrays how the system works without being too technical. However, this is only one way of thinking about the system.

The Monty Hall problem is regularly taught in elementary statistics, as it introduces the concept of dependent probabilistic analysis [11]. But, it can help understand ZKPs. Monty Hall was a prominent game show host, Let's Make a Deal. The problem can be stated:

> Monty Hall shows you three doors, two have a goat behind them, and one has a brand new car! You want to choose the door with the car, so you start by selecting one door. Hall then opens one of the unselected doors revealing a goat. Hall then asks you if you want to switch doors or stay with the one you chose.

After doing some quick math, you deduce that you will have a 66% chance of winning the car if you switch to the other door[3].

You may be wondering how this relates to ZKPs in any way, but we just performed a ZKP. Hall takes on the role of the prover, and you, the contestant, are verifying that Hall knows the answer. Hall provides no information on the answer directly, but by opening the doors that have goats behind them, Hall proves the complement. If Hall were to perform the same process with two doors, there would be two outcomes. First, Hall would open the last remaining door, showing the final goat and securing yourself with the car. The second outcome is where you lose since there are no unselected doors with a goat behind them. This problem can also be scaled to n doors, where each iteration, the contestant selects one door, and Hall opens one door with a goat, all the way down to two remaining doors. ZKPs are the heart of consensus algorithms, so a comprehensive understanding of ZKPs is vital to learning all aspects of blockchain technology.

2) The Layers of Blockchain Technology

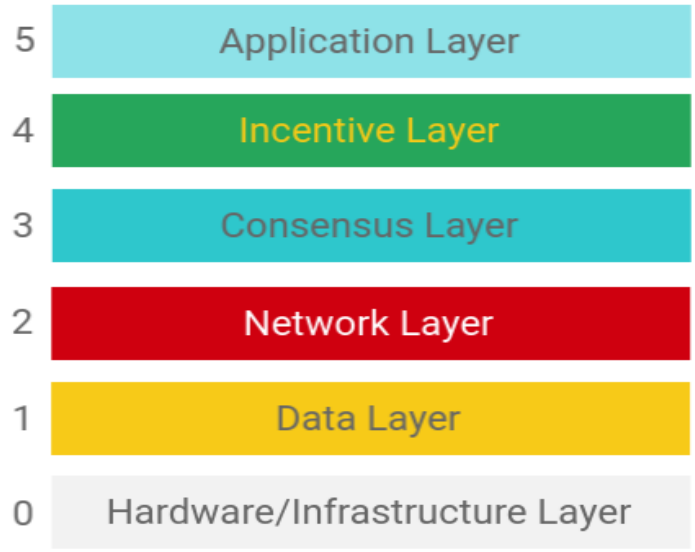

Fig. 1 Blockchain layers based on the OSI model

3 The Monty Hall problem has extensive proofs online, so I will not go into the details here, but they can be found in [11].

Blockchain technology's structure is based upon the OSI model, as the OSI model focuses on scalability and encapsulation and pays a homage to the original model for the internet. There are some similarities, but this model is designed for blockchain technology. For example, in the OSI model, layer 0 corresponds to the physical medium, but layer 0 in Fig. 1 is the infrastructure that hosts the blockchain. Blockchain technology uses cloud computing infrastructure on layer 0, and blockchain technology also uses the "as a service" model from cloud computing. In cloud computing, there are three models for service; infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Blockchain technology uses these concepts for blockchain IaaS (BIaaS), blockchain PaaS (BPaaS), and blockchain as a service (BaaS).

The service models are interpreted based on Fig. 1, where BIaaS encapsulates layers 0, 1, and 2, BPaaS determines layers 3 and 4, and BaaS allows developers to interact with the blockchain. Layer 0 defines the accessibility to the network, most commonly the PaaS and IaaS service models. This will enable developers to optimize the virtual containers for blockchain services. Layer 1 determines the block cipher algorithm and additional metadata such as wallet addresses and timestamps. The last layer for BIaaS is the network layer, which almost always uses peer-to-peer (P2P) protocols for data delivery. The hierarchy of layer 1 and layer 2 creates a decentralized environment where information is stored across distributed end-systems and are accessed through P2P communication.

BPaaS encompasses the consensus protocols as described in the previous sections. Layer 3 defines the consensus algorithms used for adding blocks to the blockchain. PoW is different from other consensus protocols as the prover computes the new block's hash on layer 1, but the third-party verification happens on layer 3. The slightly unintuitive PoW implementation is to maintain the decentralized nature of blockchain technology. This realization of PoW and mining protocols is the heart of blockchain technology, similar to the transport layer being the heart of the OSI model. Layer 4 is more straightforward than layer 3, as this layer sets the rewards for successfully adding new blocks to the blockchain.

The last layer, layer 5, defines the use of the blockchain platform. BaaS is where the developer provides some applications using blockchain for public consumption. Whether the app is for keeping secure, decentralized logs of txns or to create an economic commodity like Bitcoin and Ethereum. Another form of layering in blockchain technology is focused on the development atmosphere, where layer 0 is BIaaS development, layer 1 is BPaaS, and layer 2 is BaaS. This layering is commonly used when discussing Web3. Web3 is still being fully developed, but current protocols would force Web3 to be a BPaaS where blockchain developers can create their own BaaS to host their business and service their clients. But if interoperability is achievable between different blockchain networks, then Web3 would be a BIaaS model, where developers can host their own blockchain network specialized to their own tasks. The information thus far only grazes the surface of blockchain technology, but is enough to understand how blockchain works and why it is becoming a monumental field of focus in computer science.

3) Decentralized Consensus

Decentralized consensus is the foundation of blockchain technology. The previous section states that layer 2 decentralizes data, and layer 3 provides consensus across the decentralized network. Thus, every system that supports the blockchain's network may be queried for data from the blockchain. This is all done through the ledger stored as a Merkle tree. The Merkle tree is a binary heap containing all block hashes of the network, so accessing information can be optimal. The P2P network protocols provide decentralization, but how do ZKPs provide consensus?

ZKPs are probabilistic, so there are no guarantees that the outcome is 100% accurate. As such, this outcome is the consensus of a decentralized third-party validation, and the block is added to the ledger if the txn is valid. But, there is a question to raise; what happens when a txn is invalid? For PoW, the block is not added to the smart contract's ledger. As punishment for attempting to add an invalid txn to the ledger, nodes on higher layers of the blockchain will disconnect from the miner. So, the miner loses time, energy, and connectivity to the blockchain.

Another popular consensus algorithm used is PoS which takes place entirely on layer 4 of fig 1. PoS consensus is where people invested in a cryptocurrency stake some of their coins with a chance to gain a percentage of the incentive. Stakers provide consensus on what miner gets to mine a specified action. This system was designed to cut down the cost of wasted crypto mining energy since Bitcoin only rewards one miner, but anyone can attempt to mine the blockchain. An interesting component in PoS consensus is when the txn is invalid. The stakers lose their stake. This creates a medium risk, medium reward scenario as the growth is steady, but there is the possibility of losing the trade. PoS and other consensus algorithms bring new problems to the blockchain, such as the 51% attack. This attack brings centralization into the decentralized network, where one entity only needs to own 51% of the third-party consensus group to control the blockchain.

Consensus algorithms are a hot topic with the rapid advancements of blockchain technology, and there are far too many to cover in this article. All consensus algorithms have similarities and differences, and many have risen from previous oversight. These oversights have plagued the internet since its conception, leading to critical security vulnerabilities.

C. ZKPs, Classical Computers, and Quantum Computers

Blockchain technology is great for security, but there are some massive shortcomings. The most significant pitfall is that the consensus protocols trust everyone on the network, as stated when introducing ZKPs. ZKPs are an excellent authentication method, but that is only to ensure the information has not been compromised en route. I covered quite a bit on the security

aspects of blockchain technology in the previous sections, but you may be wondering, how does any of this relate to quantum computing? ZKPs are designed as a black-box system, where you know the input and output from a complex problem, and you have to figure out what it does and if it is correct.

Quantum computers are also described as black-box systems, but quantum computation is capable of much more than what ZKPs cannot do. The power is in the hardware, where quantum engineers use particles ranging from photons to molecules that unlock quantum mechanics for processing. The black-box system for quantum computing is the Quantum Processing Unit (QPU), and this is where the particles reside [12]. The QPU contains cells called qubits, where the particles can perform computations without all of the noise from the outside world. Quantum computing relies on similar concepts to classical computing, such as Boolean algebra, and uses quantum gates to visualize a quantum circuit. However, this is where the high-level similarities start to fade away as quantum gates are applied to the qubits rather than a bitstream passing through a gate with classical computing. Before we dive in deep, there is one fascinating topic to discuss. What is so great about quantum computing?

Qubits are unique because they utilize superposition to compute data at astonishing speeds. The following section explains why quantum computers are exponentially faster than classical computers on complex problems in the coming sections, but the qubit itself already provides a platform for quantum security. A CPU performs actions linearly and performs unnecessary operations constantly. These computations can contain PII or cryptographic keys that get briefly stored whenever a conditional instruction appears. This security vulnerability is one of the most challenging types, a hardware vulnerability. The vulnerability, called Meltdown (CVE-2017-5754), is of no fault to bad engineering, but engineers have worked tirelessly to fix the exploit to no avail [13].

The exploit identifies where the CPU stores conditional expressions and acts as a man in the middle to get the sensitive information. Meltdown targets the OS directly, and Spectre targets network-based applications. Also, Meltdown and Spectre leave a minimal footprint even while active, so detecting the attack is extremely unlikely. However, qubits map the area of a sphere, and the quantum developer gives meaning to these measurements. The black box concept of quantum computing is based on reversible algorithms where many operations are their own inverse.

Additionally, since a qubit maps a sphere, it defines a state space of two dimensions. This can be thought of as each qubit being its own bitstream, so each qubit added to the QPU multiples the space for computation by two. Ultimately, quantum computers overcome Meltdown as qubits do not compute false values as long as the algorithm supplied is correct. Hopefully, this will start to make sense as we go further into the details of quantum computing, but I have found that having a big picture view of quantum computing helps the learning process.

III. Qubit Computation

The future of quantum computing is currently going on right now. I titled this section qubit computation because it is an excellent intermediary between classical and quantum computation. The leap from classical bits (cbits) to quantum bits (qubits) is quite challenging, but the learning curve from quantum circuits to quantum algorithms is steep. Also, understanding how qubits perform computation is crucial to learning the next layer of quantum computing. Therefore, the following information will only cover up to quantum algorithms and briefly describe some fundamental quantum algorithms (mainly because I still don't quite understand them well enough). But, before we get too off track, let's explore the electric world of electrons and photons, and what it means to take a measurement.

A. Spin, Polarity, and Measurements

Qubits get their astounding capabilities from quantum behavior in particles that we manipulate [14]. Quantum computing is at a pivotal point where a lot of research is focused on what type of particle makes the best qubit for a general-purpose QPU. The historical focus was on semiconductor qubits using photons or electrons, but electrons were too easily affected by the natural world. Photons work exceptionally well because they contain a natural polarity dependent on their direction. The poles of a photon are either vertically polarized or horizontally polarized, and they also have circular polarization. The circular polarization can be either in the left or right direction or a superposition.

Superposition is the gateway to quantum behavior, and the definition can be broken down into two parts. First, a network must contain multiple currents from multiple sources. The second part puts a constraint on the values of the currents, but I will define it for qubits, not electrical circuits. Qubits contain a two-dimensional state space represented as a vector. Each value in the two-dimensional vector has a probability amplitude that scales the vector. The second constraint is on the amplitudes. More specifically, they must be orthogonal and scale a normalized vector. An equation can form (under the assumption that amplitude satisfies the constraints);

$$\{\exists\, \alpha, \beta = amplitude \mid \alpha(1) + \beta(0) = \boldsymbol{\psi}\} \qquad (1.1.1)$$

where $\boldsymbol{\psi}$ represents a state of superposition

The set is represented functionally as;

$$H(0) = \boldsymbol{\alpha}(1) + \boldsymbol{\beta}(0) = \boldsymbol{\psi} \qquad (1.1.2)$$

Let's start to analyze what this means. We know quantum computing uses a lot of linear algebra, so we need a way to abstract the linear algebra to make quantum computing faster to process visually. A predominant model for quantum computing is bra ket notation.

1) Bra Ket Notation

Discrete Mathematics uses arrays and matrices to store and index data, and we can define a qubit as an array of size two. Bra ket notation abstracts these state spaces into qubits where a bra ⟨ | defines a row vector and a ket | ⟩ defines a column vector. Qubits may have two values, but we can only measure one. In a row vector, the value of the qubit is the right value;

⟨a| := [1, 0] = ⟨0|

⟨b| := [0, 1] = ⟨1|

For kets, the bottom value defines the qubit value;

$$|a\rangle := \begin{bmatrix}1\\0\end{bmatrix} = |0\rangle \qquad |b\rangle := \begin{bmatrix}0\\1\end{bmatrix} = |1\rangle$$

Both bras and kets can be transposed when mathematics requires specific conditions. Bra ket notation allows a more intuitive analysis of quantum computing through mathematics, and these can be expanded to perform the rigorous linear algebra on the vectors and matrices to ensure correctness. Now we can better define equation 1.1.2 as;

$$H|0\rangle = \boldsymbol{\alpha}|1\rangle + \boldsymbol{\beta}|0\rangle = \alpha\begin{bmatrix}0\\1\end{bmatrix} + \beta\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}\beta\\\alpha\end{bmatrix} \quad (1.2.1)$$

The amplitudes for a qubit in superposition define the value's probability of being measured. But, specific constraints are placed on the amplitudes for the qubit to be in a superposition. The amplitudes have to be orthogonal. Two values are orthogonal if they are perpendicular in a Cartesian system. The expression for this condition is;

$$|\boldsymbol{\alpha}|^2 + |\boldsymbol{\beta}|^2 = 1 \quad (1.2.2)$$

derived from the Pythagorean Theorem. The last component is the qubits must be a normalized vector. This is formally superposition, and the resulting qubit is an orthronormal base. Quantum computing uses the complex plane to visualize the data in a two-dimensional plane, so understanding complex linear algebra will assist in learning those gates.

Equation 1.3 defines the Hadamard matrix. This operation is also defined by the matrix;

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix} \quad (1.2.3)$$

The values computed on |0⟩ are thus calculated as;

$$H|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} * \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} \quad (1.2.4)$$

The result signifies that both states in the state space receive some current, which puts the qubit into superposition. Superposition is relatively easy to represent mathematically, but engineering systems that suspend qubits in superposition are the real challenge. This state is unstable with electrons because they use spin to generate a pole. This spin can easily decohere with natural interactions, so electrons would be highly ineffective in quantum communication.

Photons work exceptionally well as qubits, but this medium requires mammoth super cooling machines to run standard operations. Quantum hardware is going through rapid changes, and the largest QPUs are constantly changing titles. People are currently waiting at the starting line to jump on quantum computers, which means there needs to be a high emphasis on security rather than rushing to completion and picking up the pieces later. But the most crucial component in hardware is measurement.

2) Measurement

The act of measuring a qubit is quite challenging in practice. First and foremost, measurement is one of a few non-reversible quantum operations. The concept of measurement can be tricky for everyone working towards or in quantum computing. One way to describe measurement is through a thought experiment based on the electron spin behavior is one way to describe measurement. This is modeled through a quantum clock that takes boolean queries and returns true if the quantum clock's pole is pointed in the direction of the question and false if otherwise. The clock starts in a superposition, and the first query to 1 outputs a false. The returned false value indicates that the pole is instead pointing at 7. Measuring the same action again, we receive the same answer. After the original query, the quantum state is static until it becomes superposed again. To do so, simply ask one query in a different direction. If the first question is then asked another time, the answer is based on a linear distribution of the two-dimensional state space.

If the second measurement is perpendicular to the first, the repeated first query will be a completely randomized result. This is another area of quantum computing that outclasses classical computing, which means real security improvements from classical systems. Random number generators are commonly used in computers, but discrete math restricts the ability to get truly randomized outputs.

B. Quantum Circuits

Quantum circuits are used to visualize a quantum process, focusing on the poles of the qubit. The quantum wire describes the passage or duration of time over computation. Each operation modifies the qubit over time, but many gates are commutative. Quantum circuit simulators often use multiple diagrams and models to visualize the qubits during computation. The most common visualization tool is the Bloch sphere, which showcases how the qubit poles change as gates are applied to it.

1) Single Qubit Gates

Quantum gates share little resemblance to classical gates. The two classical operators, NOT and AND, are universal. All computational logic can be created from a sequence of NOT and AND operations. The same is not entirely the same for quantum gates. The quantum equivalence to the classical NOT gate is the X gate, but there is no quantum gate equivalent to the AND gate.

This is because quantum gates operate on qubits, which cannot overwrite other qubits. This further explains the quantum black box concept, where the number of input streams equals the output streams.

There are two quantum gates that perform modifications on qubits that bring universality to quantum computation. However, there are some definitions required before understanding these quantum gates. Another way superposition is represented is;

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle \quad (1.3.1)$$

This representation of the Hadamard gate showcases the qubit is in a state where its phase can be manipulated to map the area of the bloch sphere. The Z gate alters the qubit phase, and the Z gate defines a unique value when used on a qubit in superposition.

$$ZH|0\rangle = |-\rangle \qquad ZH|1\rangle = |+\rangle \qquad (1.3.2)$$

This operation performs a NOT on a qubit in superposition. Since the H gate is its own inverse, applying the H gate again would give the inverse of the input with 100% probability in theory. This provides a new dimension for computation that classical computers cannot compute on. A quantum circuit with 6 qubits can compute on 64 different points compared to classical computers that can only compute on 6 unique points. However, due to decoherence and other unsolved questions, qubits require a substantial network of error correcting qubits to provide a single high functioning qubit.

Another gate important to the quest for universal quantum gates is the S gate. Since qubits operate in three dimensions, a gate such as X and Z performs a 180° rotation on some axis. The S gate performs a 90° rotation on the same path as the Z gate. The following describes the matrices for these operations;

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \qquad (1.3.3)$$

The S gate is a good start to finding universal quantum gates, but the S gate is fixed to performing a 90° rotation so this cannot provide universality.

The S gate can also be represented as the square root of the Z gate [15]. This follows mathematically as $S^2 = Z$. There are many gates that are some polynomial of Z, and this leads to a general definition of the phase manipulation on the Z plane. The same is true for the X plane as the X gate manipulates points perpendicular to the Z plane. These two gates are defined as;

$$R_X(\boldsymbol{\theta}) = \begin{bmatrix} cos\frac{\theta}{2} & -icos\frac{\theta}{2} \\ -isin\frac{\theta}{2} & cos\frac{\theta}{2} \end{bmatrix} \quad R_Z(\boldsymbol{\Phi}) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi/2} \end{bmatrix} \qquad (1.3.4)$$

These gates are also defined as R_{θ} and $R_{\boldsymbol{\Phi}}$. The X and Z gate along with the Y gate make up a set of gates called Pauli gates, defined by their Pauli matrices. The Y gate is a combination of an X and S gate but only works on superposition qubits. This is represented by the matrix;

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad (1.3.5)$$

The last single qubit modifiers are based on classical gates, the identity gate, constant-0, and constant-1. The identity gate has been touched upon previously when talking about the unitary property of the Hadamard gate. The identity gate uses a 2 x 2 identity matrix and can also be represented as a quantum wire. Constant-0 and constant-1 are non-reversible computations, restricting the qubit to a constant. This can also be done with a measurement gate, depending on the state of the qubit right before it is measured.

a) Representing multiple qubits with bra ket notation: There is one vital concept we must cover before we start working with multiple qubits. The combination of two qubits into one bra or ket is done by taking the tensor product of the two vectors. This operation is not covered in linear algebra, but is easy to compute. The tensor product for qubits is defined and calculated as follows;

$$|b\rangle \otimes |a\rangle = \begin{bmatrix} b_0 \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \\ b_1 \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} b_0 a_0 \\ b_0 a_1 \\ b_1 a_0 \\ b_1 a_1 \end{bmatrix} = |ba\rangle \qquad (1.4.1)$$

The qubits are commonly listed in standard cbit order, where the most significant qubit grows to the left.

2) Entanglement and Teleportation

Quantum gate matrix multiplication is a powerful way to showcase the obscurity of entanglement. Particles in superposition can become entangled, where each particle cannot be independently represented or factored without affecting the other particle. In other words, when two particles are entangled, measuring one will also measure the other. Measurement is difficult because it irriversibly affects the qubit, so there is no way to know if the qubit has been measured by an unknown party. This problem is an outcome of the no cloning theorem, where it is impossible to create an identical copy of a particle in superposition. However, entanglement creates a weird interaction, where particles behave identically no matter the distance between them. These particles are not copies of the other, but behave uniformly.

The CNOT gate performs a controlled-not on two qubits. One qubit is the control-bit and the other has a NOT performed based on the control-bit. The matrix for CNOT is;

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (2.1.1)$$

We'll construct the bell state, so we will put the control bit in superposition to entangle two qubits;

$$|\psi\rangle \otimes |0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}\otimes\begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0\end{bmatrix}^T = |\psi 0\rangle \quad (2.1.2)$$

We then compute the inner product on the transpose of 2.1.2 and 2.1.1 as;

$$CNOT|\psi 0\rangle = \begin{bmatrix}\frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} & 0\end{bmatrix}\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix} = \begin{bmatrix}\frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}}\end{bmatrix} (2.1.3)$$

The final product contains two qubits that cannot be factored independently. The two qubits are now entangled, so a measurement on one qubit will also correspond to a measurement on the other.

a) Quantum circuit representation: Quantum circuits are an abstraction from the bra ket mathematical formulations. The model is pretty intuitive when you know the gates. We define a quantum wire as a line with a labeled qubit. The line represents time over the whole computation on the qubit, and gates get added to the wire.

This is the representation of a qubit in superposition using a quantum circuit. The qubits start with a value of $|0\rangle$ in a quantum circuit, so the qubit is currently evaluated as $|+\rangle$. A measurement on the qubit would have α probability of being a 1 and β probability of being a 0. The bell state is represented by;

b) Quantum teleportation: Quantum entanglement brings a method of interaction that occurs faster than the speed of light. Entangled qubits cannot be factored individually, but increasing the space between entangled particles does not affect the correspondence in behavior. One important point to note is that no information cannot be passed between these two entangled particles as measurement decoheres the entangled state of the two qubits.

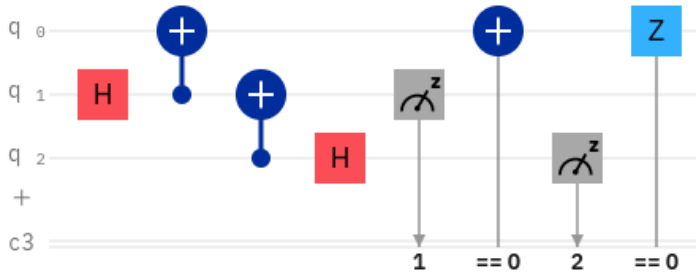

Figure 2 Quantum teleportation circuit

Quantum teleportation uses two parties, Alice and Bob, where Bob manipulates q_0, and Alice can control q_1 and q_2. The teleportation is initiated by entangling q_0 and q_1 and then separating them between the two parties. Alice then entangles q_1 and q_2 and performs some arbitrary computation. In this system, Alice can teleport two cbits of information to Bob by performing a measurement on q_1 and Bob performing a conditional X gate on q_0 depending on its measured state. Then, Alice performs another measurement on q_2 and Bob performs a conditional Z gate depending on the outcome. This results in four possible outputs, each with about the same probability, on $|000\rangle$, $|010\rangle$, $|100\rangle$, and $|110\rangle$.

While Bob's qubit will always measure 0, he can measure his qubit after Alice performs her measurement, allowing Bob to learn 2 bits of information. Quantum teleportation opens the door to highly advanced and secure communication between two parties, but the hardware is still a long way from this point.

IV. Conclusion

Learning ZKPs provides a multifaceted approach that can benefit the learning process of blockchain technology and quantum computing. Concepts around ZKPs are very apparent in blockchain technology, as consensus algorithms are ZKP systems. The relationship between ZKPs and quantum computing is not as strong in the foundational quantum computing topics, but there are areas ZKPs share strong connections with more advanced topics. Self-learning these concepts has been challenging, but I noticed stronger retention of the material than learning traditionally.

References

[1] A. Turing, *On Computable Numbers, with an Application to the Entscheidungsproblem*, Reading, 1936. [Online] Available: https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf

[2] The University of Rhode Island, "History of computers," University of Rhode Island. Available: https://homepage.cs.uri.edu/faculty/wolfe/book/Readings/Reading03.htm (Accessed Apr. 5, 2022).

[3] J. F. Kurose, K. W. Ross, *Computer Networking: a top-down approach*, 8th ed. Hoboken, NJ. Pearson, 2021.

[4] Wikipedia. "Timeline of quantum computing and communication," Wikipedia. Available: https://en.wikipedia.org/wiki/Timeline_of_quantum_computing_and_communication (Accessed Mar. 26, 2022).

[5] C. Kemmerer, "The SSL/TLS Handshake: an Overview." SSL. Available: https://www.ssl.com/article/ssl-tls-handshake-overview/ (Accessed Apr. 4, 2022).

[6] B. Perrigo, "Inside Frances Haugen's Decision to Take on Facebook." Time. Available: https://time.com/6121931/frances-haugen-facebook-whistleblower-profile/ (Accessed Apr. 20, 2022).

[7] IBM, "What is Blockchain Technology?" IBM. Available: https://www.ibm.com/topics/what-is-blockchain (Accessed Apr. 10, 2022).

[8] Disruptr, Tokens-economy. "Blockchain Consensus Encyclopedia," Tokens-economy. Available: https://tokens-economy.gitbook.io/consensus/blockchain-consensus (Accessed Apr. 15, 2022).

[9] S. Goldwasser, S. Micali, C. Rackoff, *The Knowledge Complexity of Interactive Proof-Systems*, Reading, 1988. [Online] Available: http://crypto.cs.mcgill.ca/~crepeau/COMP647/2007/TOPIC01/GMR89.pdf (Accessed Feb. 18, 2022).

[10] Wikipedia, "Zero-knowledge Proof," Wikipedia. Available: https://en.wikipedia.org/wiki/Zero-knowledge_proof (Accessed Feb. 16, 2022).

[11] J. Frost, "The Monty Hall Problem: A Statistical Illusion," Statistics by Jim. Available: https://statisticsbyjim.com/fun/monty-hall-problem/ (Accessed Feb. 20, 2022).

[12] A. Matuschak, M. Nielsen, "Quantum Computing for the Very Curious," Quantum Country. Available: https://quantum.country/ (Accessed June 2021).

[13] Graz University of Technology, "Meltdown and Spectre," Meltdownattack. Available: https://meltdownattack.com/ (Accessed July 2021).

[14] C. Bernhardt, *Quantum Computing for Everyone*. Cambridge, MA, USA: The MIT Press, 2019.

[15] Alphabet X, Microsoft, "Quantum Computing," Brilliant. Available: https://brilliant.org/courses/quantum-computing/#chapter-information (Accessed January 20, 2022)[4].

4 Brilliant courses are great but requires an account and payment for full access. I think it is worth the investment.