CISC-650 – Computer Networks

Final Paper

Due: 12/1/2019

Eric Webb

[ew774@mynsu.nova.edu](mailto:ew774@mynsu.nova.edu)

Dr. Wai Lee

**Topic Title:**

“Strength and Weaknesses in Blockchain Consensus Algorithms.”

**Abstract:**

This academic paper will be based on a technical analysis of popular blockchain consensus algorithms. The methodology for conducting my research will be done via academic and professional literature mostly based from the IEEE and ACM databases. Although some material might not currently be present in those databases, in which case I may get information from another credible source. This paper will begin to discuss pros and cons of popular and upcoming blockchain consensus algorithms. Examples of these would be the Proof of Work, Proof of Stake, Delegate Proof of Stake, Delegate Byzantine Fault Tolerance, and Transactions as a Proof of Stake algorithms. This is all contingent I find professional and academic references on each specific algorithm. In this discussion I will explain high and low level overviews of how these algorithms work along with when and where you would implement one. I will also discuss their vulnerabilities and security exploits if any are known. Out of the topics provided, I believe this one is most relevant to peer to peer communications because blockchain is essentially a peer to peer decentralized digital ledger and is fast growing invigorating technology.

**Structure:**

The structure will begin with a quick abstract and methodology.

It will then begin an introduction into what blockchain is along with a brief history.

I will then begin providing research starting with the PoW and the PoS algorithms.

The next algorithms are contingent on how much research I find on them. I will then give a conclusion of a high level over view of my findings and my personal opinions on the strengths, weaknesses, and prediction in the world market.

**Introduction:**

Since blockchains conception it has become a wide topic of discussion at many levels. Many times with the answers being abstracted away without truly grasping how the underlying architecture works. When blockchain was initiated the general public didn’t quite understand what it was, where it came from, what its future held, and what its future still holds. This narrative will begin to define what the fundamental basis of a blockchain is and how it is defined.

Blockchain has been a confusing topic for most because it is typically used in reference to the crypto currency Bitcoin. While it is true that Bitcoin uses blockchain it is not blockchains only application or specific architecture. There are numerous types of blockchains designed differently and for different purposes. It is widely agreed that blockchain in its most basic form is a decentralized digital ledger. These digital ledgers are essentially databases appending information to them via blocks. The blocks are added to the dataset effectively becoming chains of information. These database are stored within the blockchain community and are not centrally located under one entity. These databases keep a copy of the ledger and agree to make changes to it. That way if one entity has a different ledger the others can compare their copies to see where the change was made. This effectively makes blockchain pretty difficult to manipulate because it would have to effect multiple parties’ ledgers and not just one single node. Because blockchain transactions are stored as copies on multiple nodes within the network this is what makes blockchain public, verifiable, and decentralized. Blockchain is all about consensus and generally aims to answer the byzantine general problem. In this problem multiples armies must gather a consensus on when to simultaneously besiege a city. They can only succeed if they all attack correctly at the same time or else they are destined to fail. Gathering a guaranteed safe consensus solves this problem. Throughout many conversations on blockchain theory there will be many references to the BGP mentioned above and how algorithms aim to solve it. This is because blockchain aims to create verifiable trust between multiple parties as questioned in the BGP. When developing blockchain solutions not only do you have to keep in mind the CIA security triad of Confidentiality vs Integrity vs Accessibility, an architect has to keep in mind Blockchains trilemma of Decentralization vs Consistency vs Scalability. Blockchains will be designed and altered to fit specific needs around these aspects. Being truly decentralized might not be an important factor for your project if you care more about scalablity or vice versa.

**Bitcoin: History.**

Now that this literature has broadly defined the byzantine general problem and blockchains DCS trilemma it will now discuss these applications in their literal references within the cryptocurrency Bitcoin and its Proof of Work Algorithm. When blockchain was conceived it snowballed to market piggybacking off of the popularity of Bitcoin. Bitcoin and blockchain became known worldwide for its revolutionary concepts of a decentralized ledger facilitating financial transactions. Driving this new technology itself was blockchain itself. The creation of this new technology in itself is shrouded in mystery arguably because there is not a conscious agreement on who the pseudonymous entity Satoshi Nakamoto is. This was the alias used on Bitcoins white paper and the world is not sure whether it is one person or multiple entities under a guise of an alias. This has led to the background of blockchain technology and bitcoin mostly to have a low confidence and trust value among consumers. Why would the general public put their trust and money in something that had no definite creator? This would have alarmed most people from any security standpoint.

**Bitcoin: OverView**

**Bitcoin: Proof of Work**

Now that this literature has broadly defined the byzantine general problem and what blockchain widely defines it will now discuss these applications in their literal references within the cryptocurrency Bitcoin and its Proof of Work Algorithm. When blockchain was conceived it snowballed to market piggybacking off of the popularity of Bitcoin. Bitcoin and blockchain became known worldwide for its revolutionary concepts of a decentralized ledger facilitating financial transactions. Driving this new technology itself was blockchain itself. The creation of this new technology in itself is shrouded in mystery arguably because there is not a conscious agreement on who the pseudonymous entity Satoshi Nakamoto is. This was the alias used on Bitcoins white paper and the world is not sure whether it is one person or multiple entities under a guise of an alias. This has led to the background of blockchain technology and bitcoin mostly to have a low confidence and trust value among consumers. Why would the general public put their trust and money in something that had no definite creator? This would have alarmed most people from any security standpoint.

**Bitcoin : Mining**

Going into more detail these miners create candidate blocks that become eligible to add to the blockchain. These blocks must be properly constructed by a miner and has 6 parameters. The first being the version of the Bitcoin software, the second being the hash of the previous block, third is the merkle root which is a hash representation of all the transactions to be added in the candidate block. Fourth we have a timestamp. Fifth is a target hash threshold. This blocks header hashed value must be less than or equal to the networks target hash. Lastly for the sixth parameter there is the nonce value. This is a variable used in the proof of work algorithm for mining. Once those fields are filled out a block is sent to the network to be validated and if the other miners agree it is added to the ledger. These miners compete at the chance to solve hashing functions for a reward on the network. This is all effected by the network difficulty. When network difficulty increases then more hashing power is needed to compete. This is a tad bit confusing from a technical aspect because it is quite backwards on what is really happening in the code. When more hashes are being discovered on the network the algorithm recognizes this and will increase the difficulty on the network. This is accomplished by actually lowering networks target hash. When you lower the networks target hash then blocks have less of a chance of getting a number below that value. This can be misleading because to increase network difficulty you have to decrease the networks target hash value. Conversely, to decrease hashing difficulty the network raises its target hash value so miners now have a greater chance of getting below that number. This is where the nonce value comes into play. Miners compete to find a nonce value that produces a hash that is equal or less to the network difficulty. This means that not only does your machine have to solve the function but its nonce value means it has to be the randomly chosen one in a group to do so as well. When a miner finds this nonce value it is referred to as the golden nonce. This golden nonce reserves the miner the right to add the block to the chain and receive a reward. All this increasing and decreasing of network difficulty is designed to keep mining from forming a linear relation. If more powerful miners were to hit the network there would be a linear relation between the amounts of hash power and how bitcoin much they would get paid out compared to the other miners on the network which would effectively stomp out the little guy and harvest all the coins. Since the network difficulty is used industrial grade ASIC miners cant just quickly mine the whole network because the sudden increase of hashing power would effectively raise the network difficulty. This is the decentralization aimed to stop big business from stomping on the little guy and keep the market competitive although this is not as true to day with the use of ASIC miners and large mining pools.

**Bitcoin : DoubleSha256**

Bitcoins Proof of work algorithm incorporates a double sha256 implementation for cryptographic hashing. This is because new recorded transactions will be added to a block header as a hashed value using sha-256 as the merkel root. Also in the header will be a hashed representation of the previous block. Now that the header has its initial references to sha-256 hash from the merkel root and previous blocks hash it then takes the block header and applies sha-256 again effectively encapsulating the first hashes values in the new sha-256 representation of the block header. If the blockheader has enough zeros in its little endienciy then it will be accepted to the ledger. If the hashed value of the block header does not have enough zeros it will be reject and the nonce value increased to try again.

**Bitcoin : Coinbase Transaction**

The first transaction in the merkel root is called the coinbase transaction and this allows for a transaction to get a reward from if the block is successfully mined. The Coinbase transaction is different in that instead of transactions having input and outputs, this transaction creates bitcoin as a reward for adding a block. The transaction of the coinbase will contain information about the value of the transaction which is 25 bitcoins a block. It will all use the recepients public key to send the value to when confirmed. Also in the coinbase transaction is the fields for previous block hash and index but these are not needed since you are creating new bitcoins. It is popular to see arbitrary data or messages hidden in those values form miners since they are not need or used in the coinbase transaction. Again once the coinbase transaction is completed it is added to the merkel branch to create a merkel tree hashed value to be added to the block header.

**Bitcoin: Halvening.**

Bitcoins proof of work algorithm is designed to only mint 21,000,000 bitcoins in existence. To make acquiring all these coins more difficult a concept of halvening was introduced. When Bitcoin began miners used to get rewarded 50 bitcoins for adding a block. After that it halved to 25 and then to 12.5. The next halvening will make the block rward 6.25 bitcoins everytime a block is added. What this aims to do is decrease supply while keeping demand the same or higher. Now miners will only get half of what they used to for the same amount of work. This will in turn weed out unprofitable miners decreasing network difficulty till profitable again or force miners to sell their Bitcoin at a higher rate to make up for the losses.

**Overview and 51percent.**

When Satoshi Nakamoto first white papered Bitcoin and the Proof of Work algorithm they had an idea of a decentralized digital ledger that would be inept from being controlled by any one single entity. While it is true that this consensus algorithm is optimized for decentralization it is not without fault, and is not as decentralized as it was designed to be. In the beginning the difficulty of the network was low, but as more and more nodes of miners appeared on the network the difficulty began to rise. This meant that you would have to produce more hashing power to compete on the network. Eventually people stopped using their Central Processing Units for computing hasing functions and started using their graphic processing units optimized for solving these functions. This lead to a higher difficulty on the Bitcoin network. Building from that was something that Satoshi Nakamoto did not mention in the whitepaper or maybe intentionally left out, the fact you could create machines optimized for solving the SHA-256 algorithm made up of many tiny specialized CPU’s on one machine. These machines Are called Application Specific Integrated Chip’s or ASIC for short. ASIC miners bring an unfair competitive advantage to the Bitcoin mining network because they typically produce hashing results exponentially faster than traditional CPU and GPU mining rigs. This is because the ASIC is essentially a machine with many CPU’s optimized for solving this one hashing function. This takes away from Bitcoin being truly decentralized because now everyday people must by industrial grade equipment if they want to compete on the Bitcoin network. This has led to the adoption of mining pools, where groups of miners get together and share hashing power at a chance of breaking a block and splitting the profits. These mining pools have become so large that they jeopardize Bitcoins credibility of being decentralized because they make up such large chunks of the network. This exposes one of the known vulnerabilities in Bitcoins Proof of Work architecture. This flaw is called the 51% vulnerability. The 51% vulnerability essentially means that if 51% of the network agrees on a lie then that lie becomes the truth. The Bitcoin whitepaper did not account for this because the Proof of Work algorithm was intended to be a fair competition with CPU miners but was then taken advantage of with the production of mining pools and ASIC miners. Now theoretically if mining pools were to get together they could all agree on a lie essentially making it true if they composed of 51% more of the market or not.

**Etherium: Smart Contracts**

The birth of bitcoin and the proof of work consensus was the breakthrough phase in its technological evolution but soon after that came the replication phase. Where other blockchain developers aimed at building off the weaknesses of Bitcoin to develop a more advanced blockchain for modern society. This led to the birth of alternative coins called Altcoins. From the birth of these alternate coins came an advancement known in blockchain technology as decentralized applications and smart contracts. The technology allows for blockchain to not only be just a store of value but to actually deploy and run applications in a decentralized fashion. From a broad perspective a smart contract is back end server code that is ran on the decentralized blockchain network. The front end will have libraries that make reference to the back end API’s that can automate applications without a facilitator. For example, traditionally you would need a real estate broker to play the middle man in purchasing your home. In this instance if you have the money the deed for the home would be issued to you automatically from the smart contract without the need of a broker collecting fees. This is a broad example and the beauty of how smart contracts can create trust between entities and cut out the middleman in transactions. This was all made possible from a different type of consensus algorithm compared to Bitcoins traditional Proof of Work and it is called the Proof of Stake. The Proof of Stake eliminates using hashing algorithms that are taxing on the system and allow for transactions that can quickly and easily be verified. This is done through a staking architecture where nodes will put something of value at stake. If they are caught lieing or doing something fraudulent they will lose their stake and no longer be able to verify transactions. This is in contrast to the Proof of Work using copious amounts of resources just to create a digital store of value. Proof of Stake uses a little bit of resources by instead of working for the transactions they put money up front to back up they will verify the transaction correctly. One major competitor in the market is Etherium. Etherium is known for deploying smart contracts on its network using a language called “Solidity”. Solidity smart contracts give programs the fallibility to create secure transactions automatically without a third party verification system. One example of this could be an online slots machine that pays out winnings instantaneously to your cryptocurrency wallet without having to log into the proprietary websites exchange. A broader comparison would be that Proof of Stake is like a vending machine, where Proof of Work is like a cashier. The vending machine being less taxing but an upfront cost, while the cashier is literally working for their pay. This is achieved in Etheriums Proof of Stake consensus through the use of validators. Validators are nodes on the network whose votes weigh in proportion to how much stake they put up. Validators will then earn rewards for successful transactions that are processed and added to the chain. In a typical PoS consensus out of the group of validators a pseudo-randomly selected miner will process the transaction and get a reward. Of course there are known design flaws in Etheriums Proof of Stake design, first being the opposite of Bitcoins 51% network issue. Instead of owning 51% of the network someone can own 51% of the coins that are minted to be able to create malicious intent on the network. They can do this because if they own 51% of the coins they can stake their 51% percent on the network and control the majority stake. Another exploit is the “Nothing at Stake” exploit where essentially miners can put up their stake one time and be able to stake multiple forks at the same time. So if someone creates a fork of the code and creates their own blockchain similar to the original it can mine on that forked network as well without have to put up any up to stake first.

**Etherium: Sharding**

In Etherium’s Proof of Stake consensus it aims at creating a scalable applications without the loss of security by a concept of sharding. In the Proof of Work consensus all nodes store and process all transactions. This is provides a large amount of security but does not leave a lot of room for scalability. Sharding aims at grouping together nodes of the blockchain to validate transactions in independent groups. Over all increasing the blockchains throughput. A very simplified mathematical example of this would be if you had a PoW chain that has 100 nodes verifying every transaction a second versus a PoS consensus using ten groups of ten nodes solving ten transactions simultaneously a second. This is an example of how giving up security can increase throughput and scalability. The design of sharding uses validators to added information to the main chain. Typically there will exsist validators who are proof of stake nodes. Some of these nodes will be randomly chosen to validate shards. One of the chosen group will be the catalyst randomly chosen to trigger the transaction adding the shard to the main chain. This will only be triggered if the remaining specially chosen validators agree in two thirds majority with validator sharing the shard data back to the main beacon chain.

**Etherium ERC20**

**Neo: Delegate Byzantine Fault Tolerance as Practical Byzantine Fault Tolerance.**

Now that we have discussed Etheriums Proof of Stake implementing the Casper protocol we will discuss another flavor of Proof of Stake used by a blockchain company called Neo. Neo uses an implementation of PoS via its consensus deamed the Delegate Byzantine Fault Tolerance. The Point of a DBFT is that the blockchain should solve the byzantine general problem by continuing on and be resilient to issues brought up in the BGP. This is done by in the following process. Owners of the coin make transactions where are stored on book keeping nodes called delegates. These delegates verify the transactions and out of them a speaker is chosen. The speaker then verifies the transactions given by the delegates and compares them to the current hash rate for a new proposed hash rate. If the delegates agree with the speaker’s transactions and new proposed hashed rate at a two thirds majority then the transaction is added to the main chain. According to the Neo documentation the the speaker of the round is responsible for a broadcast called a prepare-request to initiate a newly proposed block. After getting the response the Speaker creates a prepared proposal block and if verified by 2/3 majority it will then move on to the Persisting stage where it will publish a new block and enter the next round of consensus. If a consensus is not reached or agreed upon then the nodes will initiate a change view proposal This will enter a new view with a new speaker and restart the consensus. This is an example of how Neo’s delegate byzantine fault tolerance performs byzantine fault tolerance by view changing of speakers and consensus creating redundancy if a block were to fail.

**Neo: NEP-5**

Prior to Proof of Stake consensus there was the traditional account balance model used in Bitcoins Proof of Work. The transactions were verified on the chain by directly recording each accounts assets. Neo is widely popular because you can easily earn Gas for processing transactions on the Neo network from staking. To execute a transaction a fee in Gas must be payed. For this example we will show how unspent transaction output is used in Neo’s transaction asset Gas. Lets say user A earns ten gas, they then send five gas to user B and five gas to user C. The first transaction will see User A have an Output of ten gas and be the original transaction index since this is where the gas was created. The Second transaction will see User A now only have an output of five gas while user B will now have an out put of five gas as well, this will be the second transaction index from user A’s initial creation of gas in the first index. Lastly, in the third transaction you will no longer see User A in the outputs because they have used up their unused outputs. The remaining outputs are user B and user C’s five Gas waiting to be spent. This is a broad over view of how unspent transaction output is used in Neo’s Byzantine fault tolerant network.

There are certain questions one must ask yourself when implementing a blockchain solution. Such as, it the data I am using going to be share across multiple parties? Decentralized ledgers are records that are stored on multiple nodes with different parties agreeing to changes. This creates a situation where any one can read or make changes to the database. In a centralized operation, if you were to want to keep a database with all your top secret information off of a network on a single system then a blockchain solution would not be ideal and traditional database would be recommended. That way only you would have access to the confidential data inside. The beauty of sharing information between parties is it eliminates the distrust between them because data is transparently stored on the decentralized ledger. A far use case would be if all big business was stored on a decentralized ledger then fraud would be exponentially more difficult because all transactions in and out would be monitored by all parties on the blockchain. If one company where to try and manipulate the data other companies could review their digital ledgers to the point where communication error occurred and effectively point the finger back at fraudulent company. These concepts are factors to think about when considering a blockchain solution and whether information should be central governed or not.

Next question is whether data should be dynamic and needs and auditable history. Blockchains are immutable, meaning that once information is added to the ledger it cannot be changed. This immutable data is left as an audit trail for other entities to verify.so if you don’t want your transaction to have a paper trail or want the contents of its history to be changed then a blockchain is not a solid option.

Another issue on when choosing to deploy a blockchain solution is speed. If a high performance that is dependent on millisecond transactions then it is best to lean towards a centralized system. Blockchains are typically still pretty slow in comparison to traditions model-client architecture. If you are customer waiting to verify a debit card transaction you probably are not willing to wait 15 minutes for the transaction to go through. This obviously creates the problem of speed.

**Chapter 3: Problems in Blockchain Security.**

When implementing new technology you must keep in mind the concept of zero day exploits, because the technology is so knew there could be a multitude of issues still to be discovered. Still for the most part if implemented correctly and adhere to secure practices and will find its place in the world. In the security realm nothing is impenetrable, even multilayered security can have its flaws and this certainly holds true in the case of blockchain. Being a new construct always comes with some kickback. Due to its rapid development many crucial mistakes were taken advantage of in the crypto currency market. Although blockchain itself was secure, the way businesses utilized it was questionable. One of the most infamous examples of a cryptocurrency hack was the incident that happened at Mt.Gox. The Mt.Gox hack at a high level was due to poor software development methodologies involving the development of blockchain applications. Another issue was that certain standards were not yet created in the blockchain community to adhere to security. It is still contested on what truly happened in the Mt.Gox hack but the underlying basis is that wallet private keys were not yet encrypted at the time, so someone was able to access wallets private keys in clear text. This in turn led to the standard practice in blockchain to encrypt wallet private keys when at rest and is a prime example of how the blockchain space and software in general evolves to meet the needs of security.

Another topic of discussion is the concept of environment costs, mining takes extreme power consumption through the use of electricity and the raw materials used to create mining hardware that supports the network. In its current state blockchain solves complex algorithms with large amounts of computing power to provide security. This could cause a problem if you are intending to deploy a large network. Each node verifying hashes is using equipment and energy which can quickly add up. Bitcoins mining nodes have been known to use more electricity than some small countries. So if you are trying to be environmentally conscious with your network, you would not deploy a large scale blockchain application.

One of the biggest issues facing blockchain today is its complexity for end users to understand. At its core users must understand public key infrastructure (PKI).The concept of wallets and having a public address seem foreign to the everyday person. It will take time for society to understand that your routing and account number are your public keys in a traditional banking environment, while your password to your account is essentially your private key in that situation. Of course this the same in crypto with your public address as your public key, and then your private key often being a mnemonic phrase or password. Certain growth has been made in this field such as cloud wallets to store your coins with ease, disconnected physical wallets for security, and updated software wallets with nice graphical user interfaces. The initial command line interface architectures were hard to learn and lead to a slow adoption growth. On top of the deep underlying technical architecture behind blockchain the average person fails to realize its real world use potential. The most popular concept the hit the ground running during blockchains conception is that it would be a disruptor in the traditional banking system.

Most people who have even heard the terms bitcoin or blockchain is its theoretic real world use of being able to create a worldwide decentralized ledger for financial transactions. Going beyond being a bank killer most do not realize its true technological core or other real world use cases. One could argue that blockchains lack of public thorough understanding and common nomenclature is a problem it is still facing today.

The next issue with blockchain technology is that since it is a new idea, its interoperability and standardization comes into question. With so many new players coming into the blockchain market, there needs to be a standardization of technology and how they interact. This has been in issue moving forward in blockchain design. Creating blockchains that can communicate freely with other blockchains becomes a cumbersome design along with getting a wide spread community of players to agree on a single standard. Creating standardization could help with application development, validate proof of concepts, as well as helping with integration. The lack of interoperability and standardization is a problem in blockchain development.

One of the more recognizable issues in blockchain development is that it is relatively slow compared to some legacy transactions systems. As a network grows with more miners validating transactions it then takes more confirmation to make a change on the ledger. This effectively creates a more secure network at the expense of speed.

So the larger a network is the more secure it will become while simultaneously becoming slower. This creates a huge scalability problem when creating a large blockchain applications for instant transactions. Of Couse advancements have been made in this field and every day companies are working toward making blockchains more instant without giving up speed. One example of this is the development of the Proof of Stake algorithm compared to the tradition Proof of Work consensus mechanisms. The Idea that consensus algorithms still need to be improved for speed and security is just another problem facing blockchains mass adoption.

**Chapter 4: Issues with Proof of Work Consensus.**

Upon blockchains conception was the first time the world would see the proof of work algorithm deployed on a public network. Arguably the beauty of the proof of work algorithm is it takes up resources just for the sake of taking up resources. As in it is designed to take up a lot of energy and computational power. The hashing algorithms are designed in a way that it is very taxing on the system. Network miners will use time and energy in hopes of solving a hashing function. If they solve it they add a block to the chain and get a reward. This is why it is called proof of work because you are working for a reward. Proof of Work was initially intended to be decentralized but is not as decentralized as it is intended. To begin on this issue special computer chips were designed specifically for calculating these hashing functions. These specialized chips are called application specific integrated chips and have dominated the Proof of Work market. This forces everyday miners to invest in high end equipment and effectively raises the difficulty rate for mining new blocks. Because Proof of Work algorithms can be dominated by ACIC miners it can lead us to the next issue with PoW. The 51% problem. If 51% of the network agrees on lie, then it becomes the truth. This would not be an issue if it was truly decentralized but considering there are large mining pools who own good chunks of the network, they could theoretically team up to create false transactions. This of course is an adherent flaw to the PoW algorithm and is a good example of how this new concept didn’t out way the advancements in physical hardware.

**Chapter 6 Proof of stake**

Moving forward from the PoW the Proof of Stake consensus came into being.

PoS allows validators to lock up funds in escrow. After that they start validating blocks. If they think a block should be appended they will add it to the chain. If it is appended successfully they will get a reward. If they are caught lying they will lose their funds in escrow and their validating positions. This is the concept of Staking your funds in an escrow account to be able to validate blocks on the network. This makes validation way more resource conservative. The PoS does come with its own issue to called the “Nothing at Stake” exploit. If there is a true primary chain and a faux branched chain a validator can put its escrow on both chains effectively winning either outcome. Because it can get a guaranteed pay out from either chain this is why it is called the nothing to stake issue. The Proof of Work algorithm mitigates this because miners will mine the longest chain because it is more profitable and risk free to do so. So here we see how the Proof of Stake protocol can be susceptible to malicious forks although advancements in this architecture are still underway.

**Chapter 7 Delegated byzantine fault tolerance.**

Another popular consensus algorithm is Delegate byzantine fault tolerance. This builds from the staking of the PoS algorithm. DBFT makes nodes that are staking to vote for a speaker to represent its changes to the main chain. Delegates are chosen and they choose a speaker to communicate with the main chain. The delegates are the book keepers who communicate to the speaker who communicates to the main chain. If the delegates all vote that the request and response between all parties are correct they will add the transaction to their records. If a speaker acts maliciously they are voted off their speaker position. If a node is caught acting nefariously they lose their stake. This is a great advantage to speed and scalability but it is not truly decentralized in nature since you need to have a node with an escrow account to be able to vote for delegates who vote for a speaker.Making it have a high entry barrier creating a buffer for decentralization adoption by everyday users.

**Work Cited.**

**References:**

Chen, W., Wu, J., Zheng, Z., Chen, C., & Zhou, Y. (2019). *Market Manipulation of Bitcoin: Evidence from Mining the Mt. Gox Transaction Network*. Paris, France: IEEE. doi: 10.1109/INFOCOM.2019.8737364

Duy, P. T., Hien, D. T. T., Hien, D. H., & Pham, V.-H. (2018). *Proceedings of the Ninth International Symposium on Information and Communication Technology*. New York NY: ACM. doi: 10.1145/3287921.3287978

Sharma, R., & Chakraborty, suchetana. (2018). *2018 International Conference on Advances in Computing, Communications and Informatics (Icacci)*. Bangalore, India: IEEE. doi: 10.1109/ICACCI.2018.8554369

Zhang, R., Xue, R., & Liu, L. (2019). *Security and Privacy on Blockchain*. New York, NY: ACM New York. doi: 10.1145/3316481

Zhao, W., Yang, S., & Lou, X. (2019). *Icbct 2019 Proceedings of the 2019 International Conference on Blockchain Technology* . New York, NY: ACM New York. doi: 10.1145/3320154.3320162

Zohar, A. (2019). *Recent trends in decentralized cryptocurrencies.* (49th ed.). New York, NY: ACM New York. doi: 10.1145/3055399.3079074

**Certification of Authorship**



Submitted to: Professor Wai Lee

Student’s Name: Eric Webb

Date of Assignment: 12/8/2019

Title of Assignment: Final Paper

Certification of Authorship: I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas, or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for this course.

Student's Signature: ERIC WEBB