Advancing Blockchain Security

Alice Karanja, Ryan Khaleghi, Mooyoung Lee, and Albert Asuncion

***Abstract*—Blockchain came about in the aftermath of the financial crisis of 2008 and became the underlying platform for Bitcoin. Bitcoin is a type of digital currency (or cryptocurrency) which was developed by an unknown person who went by the alias Satoshi Nakamoto. Since its inception, Bitcoin has grown in acceptance and to date has a market capitalization of $60 billion, 48% of the $124 billion cryptocurrency market cap.**

Bitcoin has had its share of vulnerabilities that attackers have exploited. In 2014, Tokyo's Mt. Gox Exchange lost $350 million in bitcoins, a victim of theft. In July of this year, South Korean bitcoin exchange Bithumb, was hacked and 30,000 of its customers' data were compromised.

Blockchain use cases have grown over the years, beyond cryptocurrencies. If blockchain is to become viable and relevant in conducting e-commerce, we propose variants of blockchain based upon application. By classifying transactions according to block size and monetary value, smaller blockchains can be mined under 10 minutes, mitigating double-spending attacks. Our simulations show evidence of reduced processing times as block lengths decrease.

***Index Terms*—Blockchain, distributed ledger technology, cryptography**

I. Introduction

BLOCKCHAIN is a distributed ledger technology which first came into existence in 2008 as the underlying platform for the popular cryptocurrency Bitcoin (BTC). Blockchain is a decentralized transaction ledger network, allowing transactions to be processed without recourse to a central authority thereby eliminating a single point of failure. Blockchain can be likened to an operating system with cryptocurrencies, smart contracts, and other things of value transacting within this system.

Over the years, the use cases for blockchain have grown beyond cryptocurrencies. Mainelli & Smith identified the following expanding applications for blockchain [1].

Table 1- Blockchain Applications

Area	Possible applications
Financial instruments, records, models	Currency, private and public equities, certificates of deposit, bonds, derivatives, insurance policies, voting rights associated with financial instruments, commodities, derivatives, trading records, credit data, collateral management, client money segregation, mortgage or loan records, crowdfunding, P2P lending, micro finance, (micro)charity donations, account portability, air miles and corporate tokens, etc.
Public records	Land and property titles, vehicle registries, shipping registries, satellite registries, business license, business ownership/ incorporation/ dissolution records, regulatory records, criminal records, passport, birth/death certificates, voting ID, health and safety inspections, tax returns, building and other types of permits, court records, government/listed companies/civil society, accounts and annual reports, etc.
Private records	Contracts, ID, signature, will, trust, escrow, any other type of classifiable personal data (e.g., physical details, date of birth, taste) etc.
Semiprivate/ semipublic records	High school/university degrees and professional qualifications, grades, certifications, human resources records, medical records, accounting records, business transaction records, locational data, delivery records, genome and DNA, arbitration, genealogy trees, etc.
Physical access	Digital keys to home, hotel, office, car, locker, deposit box, mail box, Internet of Things, etc.
Intellectual property	Copyrights, licenses, patents, digital rights management of music, rights management of intellectual property such as patents or trademarks, proof of authenticity or authorship, etc.

As these use cases grow for blockchain, the issues of security and trust become increasingly vital. "Trust is the foundation of security", we cannot have security without trusting institutions and network systems. With blockchain and similar distributed ledger technologies however, trust is shifted from people and institutions to trusting in the algorithms behind them. Or, as Antonopoulos puts it, a "new model of trust-by-computation" [2].

In this paper, we address the question of blockchain security. We begin with an overview of blockchain in Section II, outlining key concepts and describing how it works. An evaluation of vulnerabilities is discussed in Section III as well as verified attacks over the last few years. Section IV outlines the results of experiments we conducted to measure processing times at various block lengths as a deterrent to double-spending. We close with the Conclusion.

~~V.~~II. OVERVIEW

DEFINITION

Blockchain does not have a universal definition as it has a "number of dimensions, including technological, operational, legal, and regulatory". [3] From a technical standpoint, Kakavand et al describes a blockchain as follows:

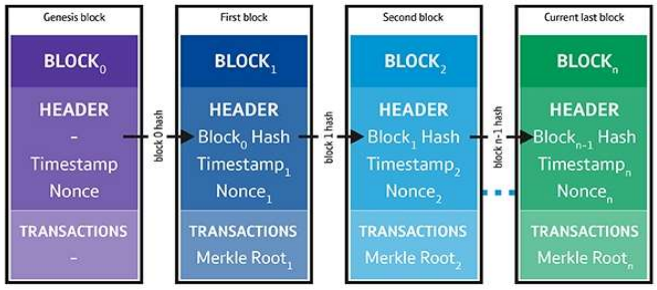
> "a database that consists of chronologically arranged bundles of transactions known as blocks, against which any proposed transaction can be checked with confidence in the integrity of any particular block."

From a transactional perspective, blockchain is a distributed ledger technology (DLT) or mutual distributed ledgers (MDL), as described by Mainelli et al, recording transactions on a shared database (i.e. ledger) operating without a central authority:

> "a type of Distributed Ledger Technology (DLT) that has been defined as a distributed, shared, encrypted database that serves as an irreversible and incorruptible repository of information."

As an innovation, blockchain is "disruptive" [Chirstensen, Innovator's Dilemma], providing an alternative to traditional centralized trust authorities (i.e. intermediaries) for transacting parties, like Alice and Bob. These centralized trust authorities can be financial institutions, escrow agents, lawyers, title companies, etc. Instead, trust is "based on rules that are defined mathematically and enforced mechanically". [3]

Figure 1- The Blockchain

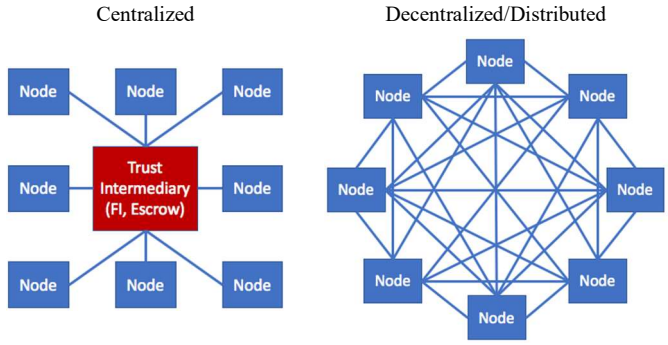


KEY FEATURES OF BLOCKCHAIN

- Decentralized & distributed – Decentralization is implemented through a distributed database that is shared by all nodes of the network. Every node has a full copy of all transactions since genesis (the first block in the chain) eliminating the "single point of failure". Figure 1 illustrates the structure of each block in the chain, each block and chain is copied across all nodes in the network.

In contrast to centralized trust mechanisms where intermediaries are responsible for transaction validation, information security, etc., blockchain decentralizes these functions to all nodes in the network, as shown in Figure 2.

Figure 2 - Trust Mechanisms



- Transparency – Not only are transactions visible to everyone, they are traceable throughout the chain, thereby transparent and verifiable. All the transactions in the ledger can be viewed by all participants of the blockchain.

- Immutable – Despite the absence of a central authority, blockchain is resistant to tampering. This is particularly critical for use cases involving trading of cryptocurrencies. Any changes to the blockchain is made by adding a new block instead of modifying the original block. Once a block is created, it cannot be altered. The mining process, which is discussed in more detail in the next section, ensures blocks are vetted because once a block is added to the chain, it cannot be altered or removed. [4]

For any owner of bitcoins, there is a record on the blockchain that contains the "coins owned", and one half of a digital signature. This digital signature is a cryptographic puzzle that only the owner of the "coins" can solve. This is because the corresponding half is a private key that is contained in their bitcoin wallet. The ledger is secured through cryptography and game theory and only the owner is permitted to change his data. It is considered hack-proof because, a hacker would have to change the same block on every computer that runs the database to make sure the chain remains identical.

- Anonymous – Despite having full copies of the database transactions, none of the users' identities are visible.

- Disintermediation – Eliminates the need for intermediaries, reducing overhead costs and mitigating single point of failure or vulnerability.

- Security – Security is provided in the blockchain by use of computer science and advanced mathematics (in form of cryptographic hash functions) that ensure the integrity of the blockchain. Authenticity is also fulfilled by using a private key that is contained in the user's wallet. Modifying or altering a block would result in a change of its hash function all the blocks following it. This change in the hash function will be detected by the other nodes and cause it to be rejected.

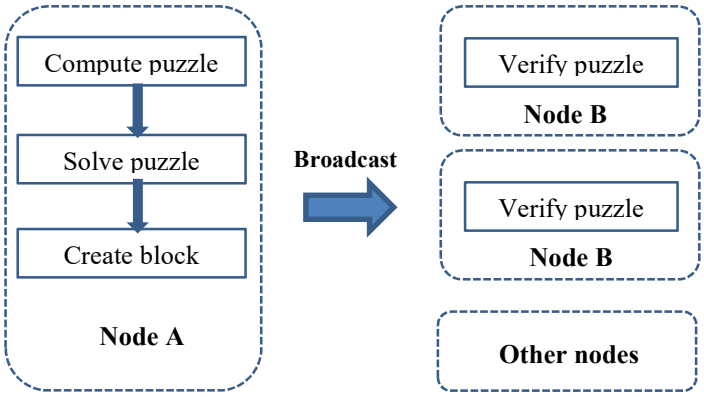
- Open Source – Open source means that has been produced, shared and published freely for anyone and no one company

or person owns or distributes it. In the same way, Bitcoin network is not owned or run by a single company.

- Turing Complete - This refers to the ability to compute anything that is computable provided one has the required resources. This property is usually applicable to Ethereum where one can write contracts that can solve almost any reasonable computational problem.

TRUST MECHANISM OF BLOCKCHAIN [5]

Figure 3 - Consensus Mechanism



- Consensus – In the absence of a central authority, a "crowd-sourced" approach to validating transactions is accomplished through consensus. This consensus mechanism is implemented through a process called mining, whereby transactions are validated and new digital currencies are created as rewards for successful completion of the miner's task.

- Mining – To validate/verify blocks and permanently append them to the blockchain, every transaction requires a miner (i.e. verifier) to present proof of "computational effort". This has come to be known as proof of work or PoW, which amounts to solving a mathematical puzzle. This puzzle is designed to be computationally difficult to solve but much simpler to verify. The puzzle is essentially an inverse hashing operation to determine the nonce which, when entered in an algorithm, is less than a given target value. PoW is represented as:

```
H(prevHash||Tx1||...||Nonce) < Target
```

Bitcoin, for example, relies on a cryptographic hash function, double SHA256 hashing algorithm, wherein the target is a 256-bit number.

The target value is adjusted to increase difficulty approximately every 14 days, motivating miners to come up with ever improving methods for efficiently solving these puzzles. This mining process is the reason behind the 10 minutes it takes to create blocks, i.e. it is the amount of time (cost) it takes to brute force the solution based on the current target difficulty.

Upon completion of the PoW, it is broadcast to the network and consensus is reached, adding the transaction block to the blockchain.

- Proof of Stake (PoS) – This is an alternative to PoW, preferred for the reduced time/cost to process. In PoS, mining involves the verification of ownership of the cryptocurrency and better suited for Smart Contracts used by Ethereum, a variant of blockchain.

Instead of splitting blocks across proportionally to the relative hash rates of miners (i.e. their mining power), proof-of-stake protocols split stake blocks proportionally to the current wealth of miners. The higher your stake, the higher chance you have of validating a block. A validator is rewarded for solving a block with transaction fees instead of coins.

Upon completion of the mining process, the concerned block is irrevocably added to the chain and will henceforth contain the PrevHash, nonce, and T.PrevHash.

STRUCTURE OF A BLOCKCHAIN

Table 2 - Structure of a Block [6]

Size	Field	Description
4 bytes	Block Size	The size of the block, in bytes, following this field
80 bytes	Block Header	Several fields form the block header
1-9 bytes (VarInt)	Transaction Counter	How many transactions follow
Variable	Transactions	The transactions recorded in this block

A block is a "container" of information consisting of data relating to each transaction (see Table 2.) It contains the list of transactions as well as a header which houses all metadata for the block. Only a fraction of the total size of the block is held by the header, the bulk of it consists of transaction data.

Table 3 - Block Header Structure [6]

Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Difficulty Target	The proof-of-work (PoW) algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the PoW algorithm

The block header is broken down into 3 sets of block metadata as summarized in Table 3:

Set 1 – Previous block's hash
Set 2 – Difficulty , Timestamp, Nonce
Set 3 – Merkle Tree Root

TYPES OF BLOCKCHAIN [10]

- Public - Anyone is allowed to set up their computer as a node and is then synced to the blockchain. Every node has the exact copy of the ledger history and is updated with every transaction. This repetition is computationally expensive and causes the process to be slow and wasteful. However, this redundancy is necessary for security of the blockchain especially for cryptocurrency blockchains. As a tradeoff, public blockchains allow for complete transparency and anonymity of the participants. Public blockchains are the best choice for networks that must be decentralized.

- Consortium - A few predetermined nodes verify and add the transactions to the blockchain. This is the equivalent of a council of elders who are known and decide who had read access. Just like a private blockchain, consortium blockchains are more efficient and have higher privacy but without giving all the power to one company.

- Private – Companies or government institutions hold the control to some extent who verify and write the transactions onto the blockchain. This allows for greater efficiency and faster verification of transactions. The company also controls who has read access which allows for greater privacy than public blockchains.

- Permissioned vs Permissionless – in permissioned/ private blockchain has restrictions such that only a few have the privilege to access and validate the transactions on the blockchain. In some cases, a permissioned blockchain can restrict access to creating smart contracts. These are sometimes known as consortium or 'hybrid' blockchains, e.g. Ripple blockchain. Perrmissionless / public blockchain is contrast, are those where anyone can join, verify transactions or create smart contracts, e.g. Blockchain and Ethereum blockchains.

BLOCKCHAIN IMPLEMETATION

Blockchain technology can be applied in three main areas [7]:

- Cryptocurrencies – Bitcoin is by far the largest implementation of blockchain. It is a digital currency (cash system) that is passed from person to person without any middleman like banks. It was designed and developed in January 2009 by Satoshi Nakamoto. It is said that there are only 21 million bitcoins in circulation. Bitcoins can be bought from a dealer or mined by solving complex mathematical puzzles. Transaction are anonymous, meaning that only the participants can see the details of the transaction. Transactions are initiated from the wallet, authorizes by all other bitcoin users and if approved, it is added as a block to the distributed ledger.

- Digital payments – Traditionally, transactions are conducted through a middleman (single authority) like a bank. All transactions must be approved by the central authority and only the participating accounts are updated or affected. In blockchain, the transaction information is transmitted to all the member computers at once in form of new blocks. This is done without relying on a trusted third party. It is done through the use of cryptocurrencies like bitcoin.

- Smart Contracts – a phrase used to describe computer code exchange property, shares or money without the need for middlemen. This is accomplished through smart contract cryptographic systems such as Ethereum encode contracts. Smart contracts run on the blockchain, they run exactly as programmed without any possibility of censorship, downtime, fraud or third-party interference. Traditionally, contracts are written by law enforcement personnel or entities like courts, lawyers or police officers.

- Database and record management – The immutable and irreversible properties of the blockchain make it ideal for safe keeping of public records.

- Ethereum – An open-source software application, launched in 2015. It is used to pay for transaction fees and services on the Ethereum network. It is also a programming language (Turing complete) running on a blockchain, helping developers to build and publish distributed applications. The potential applications of Ethereum are wide ranging. It uses Ether, a type of crypto token that fuels the network.

~~VI.~~III. VULNERABILITIES

TYPE 1 – CRYPTOCURRENCY ATTACKS

- 51% Attack – The consensus mechanism implies the agreement of the majority. In other words, if any mining pool controls more than 50% of the total hashing power of the entire blockchain, it could manipulate the transaction history. The 51% attacker essentially becomes the host of the entire blockchain [8]. They can reverse transactions and perform double spending attacks, or use the same coin multiple times. They can interrupt other miners from mining and refuse to confirm transactions from regular users. In addition, ordering of transaction can be modified or transactions excluded altogether.

- Double-spending - This type of attack is a malicious client attempt to transfer coins to multiple vendors before the transactions are fully verified. The average process time of new block, which confirms the transactions, is about 10 minutes with 20 minutes of standard deviation for Bitcoin cryptocurrency. With this much time between the proposal and transfer of payment, an attacker can double-spend before

transactions are fully validated. Another way of double-spending is when a malicious client changes the transaction history and makes a different version of the blockchain, creating a block fork. Since the longest block fork becomes the eventual chain, it leads the original fork to extinction, i.e. extinct fork. [9]

- Selfish mining - An unethical way of abusing mining mechanism. A selfish miner can create a block fork and work privately. The selfish miner keeps working privately until the public block chain length is about to catch the miner's private block chain length, and the selfish miner publishes his branch [10]. This makes the existing public branch invalid and the selfish miner takes all credit for the overlapped works.

- Transaction Privacy Leakage – Although blockchains are designed with anonymity, its privacy protection protocols are not very robust. An attacker uses methods such as 'taint analysis and tracking payments', IP address monitoring, and web-spidering. Furthermore, if an attacker obtains private information from a vendor such as email or shipping address, then it can be linked to the Bitcoin address and the onwer's identity. [11]

- Denial of Service (DoS) – DoS attack is a well-known type of attack in internet based services. DoS attacks on Bitcoin come in many forms. First, by attacking large mining pools, adversaries can temporarily takeover the mining away from the large pools. 60% of the large from the large pools have been attacked using DoS; whereas, only 17% of the small pools have been attacked. Another DoS is delaying the transaction confirmation process so that the attacker gain more time to do double-spend. Thirdly, DoS attacks on bitcoin exchanges, such as the attack on China's BTCCC, the oldest bitcoin exchange. [12]

TYPE 2 – SMART CONTRACT ATTACKS

Smart contracts allow parties to enter contractual arrangements without a trusted third party. Transaction fees are eliminated and parties can easily exchange currency or anything of value. A smart contract can be written in Solidity, Serpent, or LLL, and compiled to run in the Ethereum Virtual Machine (EVM). [13] One known vulnerability with smart contracts involves transaction fees. [14] Attackers can abuse these types of errors to steal money from contracts. DAO is a well-known case of exploiting a glitch in a smart contract. On June 18th, 2016, an unknown attacker ran same transaction code multiple times using reentrancy vulnerability in DAO and came away with 3.6m ether, which was worth about $55m at that time. [15]

Other vulnerabilities of smart contract are summarized below. [13]

Table 4 - Smart Contracts Vulnerabilities

Number	Vulnerability	Cause	Level
1	Call to the unknown	Invoked function not exiting	Script source code (Solidity, etc.)
2	Gasless send	Fee for the fallback function is expensive than 2300 gas limit to send function.	
3	Exception disorders	Inconsistency in exception handling	
4	Type casts	Type handler not showing errors.	
5	Reentrancy	Fallback allow to re-enter function. (e.g. DAO attack.)	
6	Keeping secrets	Secret field can be revealed by cryptanalysis.	
7	Immutable bugs	Consequence of contracts with a bug cannot be corrected.	EVM bytecode
8	Ether lost in transfer	Specifying wrong recipient address.	
9	Stack size limit	Call stack bounded to 1024 frames. Exploit exceptions with higher stack.	
10	Unpredictable state	State of contract from a short branch of a fork can be reverted	Blockchain system
11	Generating randomness	Craft block to bias PRNG for distribution.	
12	Time Constraints	Ability to choose a timestamp by a miner.	

UNDER-OPTIMIZED SMART CONTRACTS

There is a transaction fee to miners to process the smart contracts issued by users. There are many under optimized code that is very inefficient so that the size of the smart contract code is longer or just inefficient than what it should be. The transaction fee is proportional to the size of the bytecodes of the smart contract so the inefficient smart contract overcharges to users.

Table 5 - Under-optimized Smart Contract Types [13]

Num	Pattern	Category
1	Dead code	Useless Code Related Patterns
2	Opaque predicate	
3	Expensive operations	Expensive operations in a loop
4	Constant outcome	
5	Loop fusion	
6	Repeated computations	
7	Comparison with unilateral outcome	

Dead code is the section of a code that will never run but still increasing the transaction cost due to the extra length of code. For example, if there is an if-statement such as 'x >5' and following if-statement start with 'x*x < 20', then the inside if-statement will never run so it is considered a dead code. Opaque predicate is a case where the following statement is always true or false but stating unnecessary codes. For example, with under the 'x>5' if-statement, the following if-statement also stating as 'x>1'. The five expensive patterns can be grouped as an expensive operation. The expensive operations are generally caused by inefficient use of loops. For example, some code can be placed outside of a loop or some loops can be merged to minimized the iterations. 93.5%, 90.1%, and 80% of smart contracts are under-optimized and affected by these 3 types correspondingly: 'Dead code', 'Opaque predicate', and 'Expensive operations in a loop'. [13]

TYPE 3 – PRIVATE KEY SECURITY

Bitcoin and Ethereum utilize the Elliptic Curve Digital Signature Algorithm(ECDSA) in order to authorize a payment. Some implementation of ECDSA is providing not enough randomness during the signature process. This vulnerability in the ECDSA implementation was observed in 2010 from Sony PlayStation, and this weakness also exploited in the Bitcoin system by attackers to steal a private key. [16] Once the private key is lost, attackers will transfer the coins in the account and owner will not be able to trace the attackers.

~~VII.~~IV. TEST DESIGN

Kakavand et al identified performance metrics for evaluating blockchain performance. One test, average transaction validation latency, is described as follows:

> "Average Transaction Validation Latency: the average length of time it takes for a transaction to be validated from the time of submission. This parameter determines how long on average a user needs to wait for their transaction to be validated and placed in a block. Note that the notion of validation and block confirmations might vary for each Blockchain."

We are concerned with the average transaction validation latency because it contributes to the Double-Spending problem identified above. If processing could be made faster, then there would be less time available during processing for attempts to double-spend cryptocurrency.

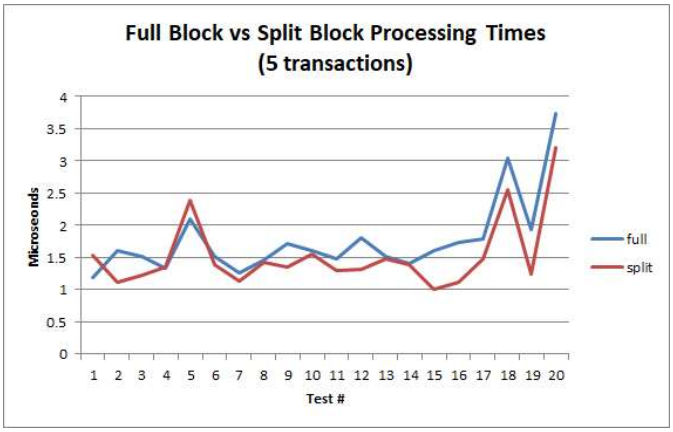
We theorized that we could solve this problem by breaking up transactions by transaction amount, and processing them separately. Transactions in the market vary widely in price, but typically there are many more small transactions than large ones. We seek to show that due to the nature of the blockchain encryption algorithm, splitting transactions by transaction amount into different chains and then processing them separately is much faster than processing them all together.

For our model we created a test blockchain in python. The approach we utilized was adapted from @ECONMUNSING's Build Your Own Blockchain: A Python Tutorial. [17] We wanted a wide range of transactions, so we created a set of 10,000 transactions between Alice and Bob with random exchanges of -5,000 to 5,000 bitcoins between their two wallets. For the base case, we then proceeded to form blocks of 5 transactions each, and process them into one blockchain. For the test case, we separated the transactions into two sets: one set of transactions between -1,000 and 1,000 bitcoins, and one set of transactions greater than 1,000 or less than -1,000 bitcoins. We then processed these into blocks of 5 transactions and turned them into 2 blockchains. We then compared the processing times for both sets of processes. We did these 20 times, and recorded the data. We then did this again for 100,000 transactions and a block size of 10 transactions to see what the effects of changing these variables would have on the model.

~~VIII.~~V. RESULTS AND ANALYSIS

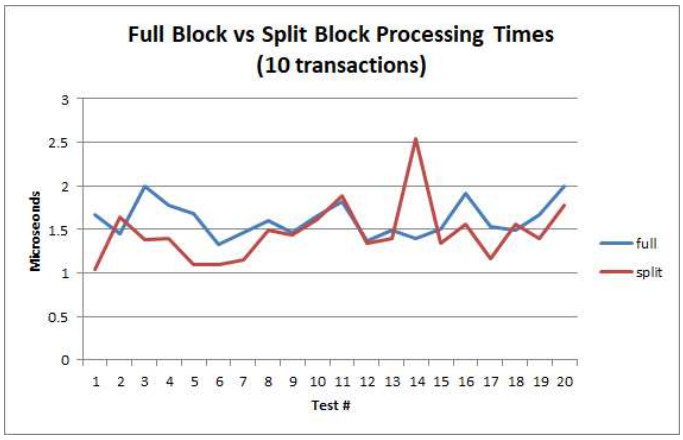
The result for our split block analysis for a block size of 5 transactions, as shown in the graph below, was an average decrease in processing time of 24% for the test case vs. the base case. In some instances, the time savings was more significant, and one instance was very minimal. This variability is due to the differing data size of the blocks and the variance in processing the hash algorithm.

Figure 4 - Processing Time (TX = 5)



Increasing the size of the blocks from 5 transactions to 10 resulted in a decrease in performance and decrease in time savings compared to the single block. The advantage in processing, as shown in the graph below, averages 14%.

Figure 5 - Processing Time (TX=10)



It is worth noting the one outlier in our sample that was significantly worse for the split block, and several points that were very close. This evidence shows that increasing the block size in our simulation has a significant negative effect on the ability to reduce time spent on processing. Given that this is a very small number of transactions per block for this efficiency gain, further analysis could find a different methodology for determining an optimal block size and divide the transactions into a sufficient number of chains so as to optimize performance. This method provides a path forward to reduce transaction time and limit the vulnerability to attacks such as double spending.

~~IX.~~VI. CONCLUSION

In our paper, we have examined the rapidly expanding and quickly developing world of the blockchain and sought to find a solution to a common problem in blockchain design. The blockchain technology has been heralded by many as "the next Internet" in terms of the impact it may have on the world, with decentralized, government-independent transactions, currency, contracts, and more replacing traditional systems. Despite the secure nature of the cryptographic algorithms used to encrypt the blockchain, vulnerabilities exist in the networking and transaction model used. We think that our analysis and solution for separating transactions for faster processing can help fix one of those vulnerabilities, and it has provided a path forward for a complete solution.

APPENDIX A

BLOCKCHAIN SIMULATION

(THIS APPENDIX IS PROVIDED AS A SEPARATE ATTACHMENT, APPENDIX_A.HTML)

REFERENCES

[1] Mainelli, M., & Smith, M. (Winter 2015). Sharing ledgers for sharing economies: an exploration of mutual distributed ledgers. *The Journal of Financial Perspectives: FinTech. Volume 3 – Issue 3*, p. 13.

[2] Antonopolous, A. Bitcoin security model: trust by computation. *O'Reilly Radar. February 20, 2014*.

[3] Kakavand, Hossein and Kost De Sevres, Nicolette and Chilton, Bart, The Blockchain Revolution: An Analysis of Regulation and Technology Related to Distributed Ledger Technologies (January 1, 2017). Available at SSRN: https://ssrn.com/abstract=2849251

[4] Pilkington, Marc, Blockchain Technology: Principles and Applications (September 18, 2015). Research Handbook on Digital Transformations, edited by F. Xavier Olleros and Majlinda Zhegu. Edward Elgar, 2016. Available at SSRN: https://ssrn.com/abstract=2662660

[5] X. Li, P. Jiang, T. Chen, X. Luo, Q. Wen, A survey on the security of blockchain systems, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.08.020

[6] A. Antonopolous, Mastering Bitcoin, Sebastopol: O'Reilly Media Inc, 2014, p 163-164.

[7] Ammous, Saifedean Hisham, (August 8, 2016) BlockchainTechnology: What is it Good for? [https://ssrn.com/abstract=2832751

[8] Dean, 51% attack (2015). URL http://cryptorials.io/glossary/51-attack/

[9] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Čapkun, "Misbehavior in Bitcoin," ACM Transactions on Information and System Security, vol. 18, no. 1, pp. 1–32, 2015.

[10] P. Franco. Understanding Bitcoin Cryptography, Engineering and Economics. Hoboken, Wiley, p.156, 2014.

[11] M. Conti, S. Kumar E, C. Lal, S. Ruj. "A Survey on Security and Privacy Issues of Bitcoin." 2017. arXiv:1706.00916v2

[12] O. Oluwoye, Xiang, J. Fu, Y. Fu, B.Herbert. "Digital Cryptocurrencies: The Design and Network Analysis of the Bitcoin Infrastructure." Digital Cryptocurrencies: The Design and Network Analysis of the Bitcoin Infrastructure, pp. ProQuest Dissertations and Theses, 2016.

[13] T. Chen, X. Li, X. Luo, X. Zhang, "Under-optimized smart contracts devour your money", in: IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 442–446, 2017

[14] N. Atzei, M. Bartoletti, and T. Cimoli, "A Survey of Attacks on Ethereum Smart Contracts (SoK)," Lecture Notes in Computer Science Principles of Security and Trust, pp. 164–186, 2017.

[15] Anon, 2016. Theft is property; The DAO. The Economist, 419(8995), p.n/a.

[16] H. Mayer, Ecdsa security in bitcoin and ethereum: a research survey, 2016.
URL http://blog.coinfabrik.com/wp-content/uploads/2016/06/ECDSA-Security-in-Bitcoin-and-Ethereum-a-Research-Survey.pdf

[17] "Build Your Own Blockchain: A Python Tutorial," @Ecomunsing, 30-Aug-2017. [Online]. Available: http://ecomunsing.com/build-your-own-blockchain. [Accessed: 04-Dec-2017].