Bitcoin and Blockchain

Group 7 Report for CSE 434 Fall of 2019 with Instructor Duo Lu

Buck
School of Computing,
Informatics, and Decision
Systems Engineering
Arizona State University
Tempe, Arizona USA
cbuck4@asu.edu

De Mouy
School of Computing,
Informatics, and Decision
Systems Engineering
Arizona State University
Tempe, Arizona USA
ldemouy@asu.edu

Jenner
School of Computing,
Informatics, and Decision
Systems Engineering
Arizona State University
Tempe, Arizona USA
eajenner@asu.edu

Nguyen
School of Computing,
Informatics, and Decision
Systems Engineering
Arizona State University
Tempe, Arizona USA
hnguye38@asu.edu

Sneberger
School of Computing,
Informatics, and Decision
Systems Engineering
Arizona State University
Tempe, Arizona USA
msneberg@asu.edu

***Abstract*—This is a group report due in CSE 434 at Arizona State University in the Fall of 2019 with instructor Duo Lu. The report covers these topics: What is Bitcoin? How does Bitcoin work over the blockchain? What is Ethereum? How is it different from Bitcoin? What are the nodes in a blockchain network? What kind of messages do they communicate? What is Hyperledger? How do these network nodes set up trust in a decentralized network? Is blockchain secure? What are the downsides of blockchain, including security issues facing blockchain.**

Keywords—blockchain, Bitcoin, Ethereum, Linux Foundation, Hyperledger, Sawtooth, Nodes, Hashcash, Ethash. Forking Inconsistencies, Gas Griefing, Reentrancy, Front Running, Integer Over/Underflow, revert Functions

I. Introduction

Blockchain technology is growing in prevalence in the realm of computer science, with new ideas as to how blockchain technology will impact the general public being proposed and developed at a rapid pace. Popularized by the cryptocurrency Bitcoin, blockchain platforms offer the structure needed to create a distributed ledger in the form of a blockchain. “The main feature of blockchain is that it allows untrusted participants to communicate and send transactions between each other in a secure way without the need of a trusted third party” [1]. Blockchain’s immutability, transparency, and auditability make it attractive for many applications, but it is blockchain’s decentralization that breaks new ground. But blockchain has downsides. Even in the face of sophisticated cryptography, blockchain is not 100 percent secure, and blockchain has scalability issues which are directly related to blockchain’s massive use of computerization and this electricity, leading to a detrimental environmental impact.

II. Blockchain

A. *Technical Overview*

Distilled to its essence, a blockchain is a decentralized, distributed ledger [1] stored and serviced by a peer-to-peer network of nodes. Blockchain uses a system of consensus by participating nodes as well as cryptography to ensure that this digital ledger can only be appended to and not edited. A meaningful way to think of a blockchain is to picture a singly linked list, but where there is a formal approval process before a link may be added, and once a link is approved and added it links to the previous link and there is no functionality allowing that link to be altered in any way, or to be moved within the list of deleted from the list. This results in links of the blockchain being approved, permanent and inalterable entries in the distributed ledger represented by the blockchain and stored in total by each node.

A genesis block is the first block in the blockchain, and contains all rules and protocols necessary to formalize the ensuing chain structure.

B. *What are the nodes in a blockchain network? What kind of messages do they communicate?*

Every device participating in the peer-to-peer network of the blockchain is a node. Each node stores part or all of the blockchain.[2] For a node to participate in the blockchain process, it must be continually connected to the peer-to-peer network.

[4] Nodes undertake two functions:

[1] Hyperledger is a collaboration headed by the Linux Foundation joined by Monax and IBM to support blockchain-based distributed ledgers. [2]

[2] That the blockchain is stored at each node leads to one of blockchains technical challenges: scalability. Bitcoin limits its block sizes to one megabyte, but even with that limit Bitcoin can only handle seven transactions per minute, thus limiting its usefulness in high-frequency situations. Additionally, as no blocks ever leave the blockchain, the chain’s size continues to grow. At some points nodes will no longer want to provide the amount of storage needed to maintain the required multiple copies of the blockchain [3].

- Generating and broadcasting a new block requesting to be added to the blockchain; and or
- Validating blocks requesting to be added to the blockchain, and if validated adding them to the blockchain.

C. *How do these network nodes set up trust in a decentralized network?*

Transactions made over a centralized network are detached from the users and is instead carried out by a third party at the center, for example a bank. Users must trust this third party to update accounts correctly and not become corrupted by system error or hacking. Blockchain currency transactions are made over a decentralized network, meaning there is no single party responsible for updating and maintaining account information. Every node in the network keeps track of every transaction made so if a single node records incorrect data, it would be invalidated by the other nodes [x].

Verification of Links in a Blockchain

Nodes are only added to a blockchain under a concept called consensus - whereby each node of the blockchain consents to the addition of the block to the chain [6]. In practice, nodes use two surrogate models to assure consensus: Proof of Work, or Proof of Stake [6].

Proof of Work involves a specialized node called a miner which shows that it spent a determined amount of computational resources in the form of solving a cryptographic puzzle. Bitcoin uses a scheme called Hashcash which involves calculating a hash value that matches a Hashcash stamp. Ethereum uses a similar scheme called Ethash which is memory intensive in that it requires mining a one gigabyte data set derived from the headers of approximately 30,000 prior blocks in the chain, but similarly requires a miner to reach a particular hash value [4, 5]. The effective feature of either Proof of Work scheme is that while the Hashcash or Ethash take significant effort to calculate, they can be checked with nominal effort. This requires miner nodes to go to significant trouble to gain consensus for a new node, while all nodes can affirm that the Proof of Work has been done.

Proof of Stake involves a miner node to be chosen at random based on having met a financial or work stake in the blockchain [6]. The idea behind the Proof of Stake is that the approved miner has such a high stake in the blockchain that it is unlikely that the miner would create an invalid block [6]. It is self-evident that the Proof of Stake scheme of consensus is more efficient than Proof of Work, and the computation required by Proof of Work[3] are not required for Proof of Stake, but has a weakness in requiring trust in a particular blockchain node.

III. Different Blockchain Platforms

The first blockchain was developed to support the Bitcoin crypto-currency, however many blockchain platforms have been developed since [2]. Let us review the most popular blockchain platforms [10].

A. *Bitcoin*

Bitcoin is a cryptocurrency and it was proposed by pseudonymous software developer Satoshi Nakamoto in 2009. The idea was to produce a mean of exchange, independent of any central authority, that could be transferred electronically in a secure, verifiable and immutable way [11]. Bitcoin can be sent from one user to another user on peer-to-peer bitcoin network without the need for intermediaries. Transactions are verified by the network nodes through cryptography and recorded in a public distributed ledger called a blockchain.

The original blockchain platform: Bitcoin provides computing capability sufficient to support its namesake cryptocurrency, but because its “stack-based bytecode scripting language” isn’t Turing-complete, the language has limited support for smart contracts and other advanced applications [1]. As mentioned above, Bitcoin uses the Hashcash consensus scheme [8].

B. *Ethereum*

In the beginning, Ethereum was described in a white paper by Vitalik Buterin, a programmer involved with the Bitcoin newspaper to build decentralized applications. Vitalik believed that Bitcoin needed a scripting language for application development. However, the idea are failed to gain agreement, so he proposed the development of a better platform with a more specific general scripting language. Besides that, Ethereum had a long list of the founders. It was founded by "Vitalik Buterin, Anthony Di Lorio, Charles Hoskinson, Mihai Alisie, and Amir Chetrit (the initial 5) in December 2013. Joseph Lubin, Gavin Wood, and Jeffrey Wilke were added in early 2014 as founders [12].

With its support of Turing-complete programming languages such as Solidity, Serpent, and LLL, the Ethereum blockchain platform can support advanced functions that facilitate smart contracts and gambling applications [1]. Code from languages on top of the Ethereum platform are compiled into EVM bytecodes and run in the Ethereum Virtual Machine (EVM). As mentioned above, Ethereum uses the Ethash consensus scheme. In the future Ethereum intends to switch to a Proof of Stake scheme based on weighted voting where nodes will vote on which blocks they think are correct and the majority is determined by the stake of the blocks in the chain [7]. To illustrate a trivial example: if one node has a 51% stake in the chain, and there are 49 other nodes with 1% stake each, the node with the 51% stake will constitute the majority regardless of the votes of the other 49 nodes.

C. *IBM Open Blockchain (OBC)*

Open blockchain is a ledger of digital events. It is called

[3] “According to the bitcoin energy consumption tracker at Digiconomist, bitcoin currently consumes 66.7 terawatt-hours per year. That’s comparable to the total energy consumption of the Czech Republic, a country of 10.6 million people” [9].

transactions that share with different participants, and each of them can have a stake in the system. The participants are the only one who could update the ledger and record. This information will never be altered, and each recorded event needs proof of agreement from participants to verifiable.

Recently open sourced as a part of IBM's participation in the Linux Foundation's Hyperledger program, OBC is intended to automate business processes "by deploying business rules as smart contracts on the blockchain" [7]. OBC lacks support for confidential and private transactions and requires an authority to allow nodes to join and as a result is distinguishable from Bitcoin and Ethereum [7].

D. Intel Sawtooth Lake

Intel was creating a new modular platform called Sawtooth Lake. They said that it is the second coming of blockchain and its use for developing and maintaining distributed ledgers. A distributed ledger has three main components [13]:

- A data model that captures the current state of the ledger
- A language of transactions that change the ledger state
- A protocol used to build consensus among participants around which transactions will be accepted by the ledger.

According to Intel, they implemented the data model and transaction language in a "transaction family" in Sawtooth Lake. They offer three transaction families for efficient building, testing and deploying a marketplace for digital assets [13]:

- EndPointRegistry - A transaction family for registering ledger services.
- IntegerKey - A transaction family used for testing deployed ledgers.
- MarketPlace - A transaction family for buying, selling and trading digital assets.

By using these three transactions families, Intel designed an "out of the box" ledger. So now, they can start implementing a fully functional marketplace for digital assets.

Sawtooth Lake is similar to Bitcoin but provides additional functionality by way of the Python API [4], as well as providing for two consensus schemes aimed at different applications:

- Quorum Voting using multiple rounds of votes to achieve consensus, and
- The resource-intensive Proof of Elapsed Time scheme developed by Intel [7].

E. Differences Between Bitcoin and Ethereum

The primary differences between Bitcoin and Ethereum as mentioned above are that Ethereum, unlike Bitcoin, supports Turing-complete languages allowing more sophisticated transactions such as smart contracts to be added to the blockchain, and that Bitcoin uses Hascash for its consensus by Proof of Work model while Ethereum uses Ethash.

It has been said that:

> Ethereum and OBC are far in front of the other platforms in terms of usability. Both have multiple methods for interacting with the platform with even more planned for the near future. OBC's plans to allow for contracts to be programmed in Java and Javascript (two fairly common languages) put it slightly ahead of Ethereum which requires knowledge of Solidity or Serpent, which are specific to the Ethereum VM

[7]. Thus, it appears that the flexibility Ethereum exhibits in allowing the development of sophisticated algorithms on top of its blockchain structure give it a decided edge over the earlier Bitcoin.

Ethereum is reported to have an edge in documentation, and as a part of the Linux Foundation's Hyperledger Project its prospects for further development and documentation appear to edge the legacy Bitcoin

F. Blockchain Is Being Put To Many Uses

Today, numerous computer scientists and policy makers work on the projects aimed at solving the most serious problems with the help of the blockchain technology and its offspring smart contracts. Applications include:

- Digital Identity: details about your life
- Banking: transfer money with no or little fee
- Tax Records: organizing records
- Insurance: organization of policies
- Real Estate and Land Titles Recording: public records
- Supply Chain: food safety
- IoT (Internet of Things): reduction of friction in a smart house or factory
- Authorship and Intellectual Property Rights: maintaining a transparent registry
- Life Science and Health Care: on demand health records
- Gaming and Gambling: honest record-keeping
- Educational Records: permanent, immutable, searchable transcripts

[11]. Surely, many new applications will be found where applying blockchain technology appears to provide value and efficiency.

IV. Security Issues Facing Blockchain

Development platforms for blockchain are always evolving. This technology is still at a nascent stage. The developer community is trying to assure that blockchain meets its promise of secure transactions. But the correct framework on Ethereum and other similar frameworks, though secure enough to be used in the real world with real transactions, do not have well-defined and well-documented standards to write code. This has led to many vulnerabilities in poorly written code leading to attacks. To make developers more confident in writing code for Blockchain, there is a need for standardization and documentation of the frameworks on which they run their programs.

Ethereum and other platforms' digital signature protocols rely on the secured privacy of the signer's private key, thus any breach of the security of these keys would allow malicious actors to undertake seemingly unassailable signatures on behalf of another party. This is the age-old problem of keeping something under lock and key safe from those who wish to take it.

It has also been rumored that the NATIONAL SECURITY ADMINISTRATION has inserted a backdoor of sorts in the elliptical curve cryptography protocol used by Ethereum, which raises a whole group of concerns related to privacy and government interference in commerce, especially in totalitarian regimes.

Future technology such as quantum computing and its potential to unlock new avenues of brute-force and other calculations could eventually provide a crack in the security provided by cryptographic and digital signature protocols currently deemed secure. Although cryptography has made great strides since RSA was developed in the 1970s, history has proven that no cryptographic solution is secure *forever*.

Simulation with the statistical model checking tools reveal scenarios where the blockchain can be breached by hackers:

> **Scenario 1:** hacker retrieves the name from the pending transactions data when the user transaction is not mined yet. Here the hacker has an average of 12% to hack the register, succeeding when both the hacker and the user transactions are in the pending transaction list. Due to the random mining, the hacker transaction can be mined and executed first
> **Scenario 2:** He gets the name from the network, that is, directly from the user call interaction with the blockchain. Statistical Model Checking (SMC) probability evaluation feature with parameters $\alpha = 0.1$ and $\delta = 0.1$ is used [14]. Here the hacker has an average of 25% to hack the register, succeeding by intercepting the transaction while the user sends the register call transaction

[12]. Since in scenario 1, the hacker should wait for the user's transaction to be in the pending transaction list, it explains the smaller chance for the hacker to succeed than in scenario 2. In order to avoid such attacks, a rigorous registration process should require two steps. First the user registers the hash of the name and then only he registers the actual corresponding name in a second transaction.

Developers working with blockchain face many different kinds of problems. Blockchains can have problems due to their internal structure, or poor development practices by their developer. Examples of issues with blockchain may include the following:

1. *The Majority Attack*

During the Proof of Work phase, miners may utilize differences in computing power or work together to achieve a majority of the work [12], and thus be able to gain control over the blockchain network in weighted voting schemes as previously mentioned in the section on Ethereum. By gaining control over the network they can decide how blocks are to be modified and deny the work of other miners within the network, thus giving them total control.

2. *Forking inconsistencies*

If there is a change to the way that consensus is formed in the network, it is possible that certain nodes will not receive the update rules, and thus create a split in how the chain is being processed. These splits are called forks which may occur in two types: hard or soft [12].

A hard fork is a change to a protocol that renders older versions invalid. They are needed to change defining parameters such as the block size and other factors, but any change to these factors can cause blocks to be rejected by older versions and can cause problems. Having a hard fork can be messy and risky as well. The reason for this is because bitcoins exchanged in a newly changed block could be exchanged again on an old block [15].

On the other hand, soft forks do not carry the double-spend risk that plagues hard forks since it can read both old and new version blocks. With soft forks, if a protocol is changed so it tightens the rule, but does not affect the structure of it, the new version blocks will be accepted by the old version nodes. For example, if the community reduced the block size to 0.5MB from the current limit of 1MB, the new version nodes would reject 1MB blocks, and would build on the previous block. Introducing this 1MB limit was done though a soft fork since bitcoin didn't initially have a block size limit [15].

3. *The scale of the blockchain*

Because it requires a substantial amount of computation to verify and maintain blocks in a chain, and these grow as the size of the chain increases. The performance of the network can degrade over time. Blockchain developers have investigated a variety of techniques to reduce the performance issues of these networks. Unfortunately, there is still much room for improvement in consensus models. A new approach to consensus developed by Naoki Shibata [16] is called Proof of Search and will be discussed later in this paper.

4. *Current regulations and rules*

Because blockchains are not controlled by any government entity, and instead are a distributed network of peers, this poses problems for agencies such as the IRS to track transactions between various entities, thus leading to government crackdowns in various nations across the globe [12].

5. Cost of infrastructure

As every miner in a blockchain is computationally competing to finish the Proof of Work and Proof of Stake tasks in a block [12], this has created an arms race by miners which has resulted in shortages in several markets such as GPUs and high thread count CPUs, leading to massive price increases for the consumer market.

Additionally, there are security issues that may result from poor coding practices:

1. Reentrancy on a single function

In reentrancy attacks an attacker calls the same function multiple times in separate threads, this means that a function may not complete before another copy of the function is run, which may result in the data being changed before the first function resumes. Allowing an attacker to for example make multiple withdrawals if each function is paused before reaching code ending the transaction. This has been exploited in the past and during the DAO attack allowed attackers to steal $50 million worth of ethercoin [17].

2. Front running

In front running an attacker utilizes knowledge about what actions haven't been performed yet and modifies the block. If it is known that delivery hasn't occurred yet an attacker may directly modify the structure of the block to set delivery as occurred and bypass payment [17].

3. Integer Overflow and Underflow

Because integers, like all numerical data types in the context of computers, are finite and thus may overflow or underflow, developers must not assume that a particular variable is safe against memory manipulation of these values which can result in situations like a value being 0 on an unsigned int being converted into the maximum value of the int by subtracting 1, or similarly being at the maximum value of the int and adding 1 and getting 0 [17].

4. Using revert functions

In contracts that have a refund mechanism without checks, an attacker may call refund multiple times to steal more money than they initially put in [17]. This can additionally be combined with a re-entrancy attack to achieve the same effect if threading has not been properly considered.

5. Insufficient gas griefing

An Attacker may inject a null contract into another contract in the chain, once this contract calls the null it will fail, and may cause the contract holding it to fail. Performing appropriate checks when calling contracts may alleviate this vulnerability [17].

6. The "Proof-of-Search" Consensus Model

Proof of Search is a new consensus model developed by Naoki Shibata based on Proof of Work. The primary goal of Proof of Search is to reduce the computational workload by using otherwise wasted computing time to search for a solution to an optimization problem [16]. For the purposes of Proof of Search the solution does not need to be exact as it is using leftover cycles.

Proof of Search introduces the concepts of the evaluator, searcher, client and job [15]. The evaluator computes the value of a solution to an optimization problem that could potentially be solved. Given the same input, the evaluator must always output the same value across any computer for every instance of the problem. Any node in the network can act as a client and submit a job to the system. A job is a request to search for a solution to an optimization problem, including an evaluator and all other information required for a searcher to search for a solution with a randomized search algorithm [15].

Similar to Proof of Work, miners must find a nonce to make a proposed block's hash value match with a required number of zero bits. However, due to how nonce values are generated in Proof of Search, not every nonce is valid. A nonce in Proof of Search includes a solution candidate and the solution value provided by the evaluator. In order to generate a valid nonce, miners evaluate solution candidates until the block's hash value matches the correct number of starting zero bits, proving many candidates have been evaluated. The node that finds the best solution is rewarded by the client and a new block is added [15].

The minimal Proof of Search scheme works similarly to Proof of Work and is limited in its use of a single fixed evaluator, hindering its ability to find a good solution. Proof of Search should incentivize miners to provide good solutions while disincentivizing searching for a solution for a problem that already has a good solution or pursuing a solution for a problem with a low solving value. All this must be done while preventing malicious miners from being rewarded. To accomplish this, the client pays for the job which is then used as the incentive for finding the best solution [15].

Proof of Search can be made more efficient by allowing the submission and execution of multiple jobs. To incentivize miners further, the probability of earning a reward increases with computation power used on the job. This is done by adding mini-blocks in between blocks that each correspond to a different job. Mini-blocks are made up of a nonce, the ID of the miner, and the hash value of the last block. A valid nonce showls a solution candidate. When a valid nonce is found, the mini-block and its nonce are broadcasted over the network. Once every mini-block job is carried out, the block is added [15].

IV. Environmental Impact of Blockchain Technology

Another major setback for current Blockchains is the computational need to participate in the network. To make Blockchain to be widely accepted, the main problems to be addressed are the robustness of the code, scalability issues and computational needs of the network. Until recently participating in a blockchain meant that a miner had to build complex computational systems with huge power demands, many GPUs, a server grade processor and network adaptor to handle the computational load to be part of a network and also be profitable in mining.

Some of the developments in Ethereum to tackle the scaling issues are:

1. *Raiden Network:* It is a network built on top of the Ethereum ecosystem which allows the actual transfer sequence, where the value is transferred from one entity to another, to happen off the main blockchain network. This avoids the consensus bottlenecks in the peer-to-peer payment networks.

2. *Plasma:* It is a framework that allows hierarchical setup of blockchains for Smart Contracts. We can have several contracts running in this blockchain in a hierarchical setup where we can have some contracts as children of other smart contracts. So, the entire network will contain a root blockchain and then several blockchains for parents and children. The idea to use this setup is that it will be much easier to manage dependent smart contracts. With this setup, the children can periodically update the parent to maintain data integrity and concurrency.

3. *Sharding:* This technique divides the network into shards and the validators in the network only need to worry about validating a subset of the transactions belonging to their shard. Here, shard is just a portion of the network, which includes a certain set of validators and certain transactions for the shard. By dividing the work among different shards, developers have been able to get higher throughput from the blockchain network.

The main aim for the Blockchain community is to make mining accessible to the large masses so that many people can participate in the mining process. This will facilitate the distribution of load over a bigger network because of increased availability of processing devices. This would also mean that the network would comprise of relativity less powerful machines (like a laptop or home desktop), which are already in abundance, connected to the biggest network in the world - the World Wide Web. This would encourage more miners and ultimately help in reaching the goal of a blockchain network which is having a completely decentralized system to handle transactions, by cutting down the middle-men and creating complete transparency in all the transactions in the network. Finally, the immutable nature of blockchain prevents the remedy of vulnerability in existing blocks and chains. This is why it is important to understand blockchain, and for engineers working on these technologies to focus on keeping the widespread, utilitarian technological aids secure from onset.

V. Conclusion

The emergence of blockchain technology, along with the surge of cryptocurrencies, has revolutionized networking systems, with its transparency, security, and immutability. As blockchain continues to become more efficient and accessible, it will integrate into other industries for a multitude of different applications. Implementing a blockchain network over third-party bureaucratic systems will greatly reduce time and effort spent in data upkeep and transaction, along with a lesser probability of error. Blockchain is a technology that users can trust to efficiently and effectively use to keep and transact data over the network, even if users do not trust each other.

[1] Maher Alharby, Aad van Moorsel. 2017. Blockchain-based Smart Contracts: A Systematic Mapping Study. In: Fourth Int'l Conf. on Computer Science and Information Technology (CSIT-2017), 2017.

[2] "Blockchain." Retrieved October 27, 2019 from: https://en.wikipedia.org/wiki/Blockchain

[3] M. Yusuf. "A comprehensive list of blockchain platforms." Retrieved September 9, 2019 from:: https://www.technoduet.com/a-comprehensive-list-of-blockchain-platforms/

[4] M. Macdonald, Lisa Liu-Thorrold, R. Julien. 2017. The Blockchain: A Comparison of Platforms and Their Uses Beyond Bitcoin. 10.13140/RG.2.2.23274.52164.

[5] Christian Searer. February 1st, 2018. Building a Network of Trust using Blockchain Technology. Retrieved from: https://medium.com/regen-network/building-a-network-of-trust-using-blockchain-technology-1745b295c6c7

[6] Zheng, Zibin, et al. "An overview of blockchain technology: Architecture, consensus, and future trends." 2017 IEEE International Congress on Big Data (BigData Congress). IEEE, 2017.

[7] Yaga, D., P. Mell, N. Roby, and K. Scarfone. "Blockchain Technology Overview. National Institute of Standards and Technology Internal Report 8202, 66 pages." (2018).

[8] S.S. Yau, slides used for CSE 543 as ASU Fall 2019.

[9] Hashcash. Retrieved from: https://en.wikipedia.org/wiki/Hashcash October 27, 2019.

[10] Bitcoin is an energy hog. Where is all that electricity coming from? Retrieved from: https://www.vox.com/2019/6/18/18642645/bitcoin-energy-price-renewable-china October 27, 2019.

[11] Danda B. Rawat, Vijay Chaudhary, Ronald Doku. 2019. Blockchain: Emerging Applications and Use Cases. In: arXiv:1904.12247vl [cs:CR]

[12] Tesnim Abdellatif, Kei-Leo Brousmiche. 2018. Formal Verification of Smart Contracts Based on Users and Blockchain Behavior Models. In: 2018 9th IFIP Int'l Conf. on New Technologies, Mobility and Security (NTMS), February 26-28, 2018

[13] Iuon-Change Lin, Tzu-Chun Liao. 2017. A survey of blockchain security issues and challenges. International Journal of Network Security. 19. 653-659

[14] Anonymous. 2019. Ethereum Smart Contract Best Practices. Retrieved from: https://consensys.github.io/smart-contract-best-practices/known_attacks/

[15] Naoki Shibata, August 6th, 2019. Blockchain Consensus Formation while Solving Optimization Problems. Retrieved from: http://arxiv.org/abs/1908.01915v1

[16] Noelle Acheson, March 16th, 2018. Hard Fork vs Soft Fork. Retrieved from: https://www.coindesk.com/information/hard-fork-vs-soft-fork

[17] Tatsiana Yablonskaya, August 31st, 2017. Intel announces distributed ledger platform named "Sawtooth Lake. Retrieved from: https://www.coinspeaker.com/intel-announces-sawtooth-lake-project/#