International Journal of Aquatic Science
ISSN: 2008-8019
Vol 15, Issue 01, 2024

Empowering Voters: The Evolution of E Voting with Blockchain Technology

Amisha bharadwaj[1], Aakansha Kamble[2], Ananya Dangare[3], Prashik Koche[4], Nayan Shivankar[5], Sanket Satone[6], Dr. Amit Thakare[7]

[1,2,3,4,5,6]Students, Dept of Computer Engineeering, Bapurao Deshmukh College of Engineering Sevagram, Wardha.
[7]Asstt. Professor, Dept of Computer Engineeering, Bapurao Deshmukh College of Engineering Sevagram, Wardha.

Abstract: Building an electronic voting system that satisfies the legal requirements of legislators has been a challenge for a long time. Distributed ledger technologies is an exciting technological advancement in the information technology world. Blockchain technologies offer an infinite range of applications benefiting from sharing economies. This paper aims to evaluate the application of blockchain as service to implement distributed electronic voting systems. The paper elicitates the requirements of building electronic voting systems and identifies the legal and technological limitations of using blockchain as a service for realizing such systems. The paper starts by evaluating some of the popular blockchain frameworks that offer blockchain as a service. We then propose a novel electronic voting system based on blockchain that addresses all limitations we discovered. More generally this paper evaluates the potential of distributed ledger technologies through the description of a case study, namely the process of an election and implementing a blockchain-based application which improves the security and decreases the cost of hosting a nationwide election.

1. INTRODUCTION

In every democracy, the security of an election is a matter of national security. The computer security field has for a decade studied the possibilities of electronic voting systems [1], with the goal of minimizing the cost of having a national election, while fulfilling and increasing the security conditions of an election. From the dawn of democratically electing candidates, the voting system has been based on pen and paper. Replacing the traditional pen and paper scheme with a new election system is critical to limit fraud and having the voting process traceable and verifiable [2].

Electronic voting machines have been viewed as flawed, by the security community, primarily based on physical security concerns. Anyone with physical access to such machine can sabotage the machine, thereby affecting all votes cast on the aforementioned machine.

Enter blockchain technology. A blockchain is a distributed, immutable, incontrovertible, public ledger. This new technology works through four main features:

The ledger exists in many different locations: No single point of failure in the maintenance of the distributed ledger.

There is distributed control over who can append new transactions to the ledger.

Any proposed “new block” to the ledger must reference the previous version of the ledger, creating an immutable chain from where the blockchain gets its name, and thus preventing tampering with the integrity of previous entries.

A majority of the network nodes must reach a consensus before a proposed new block of entries becomes a permanent part of the ledger.

Blockchain as a Service

The blockchain technology was introduced in 2008 when Satoshi Nakamoto created the first cryptocurrency called Bitcoin. The Bitcoin blockchain technology uses a decentralized public ledger combined with Pow (Proof-ofWork) based stocastic concensus protocol, with financial incentives to record a totally ordered sequence of blocks, the blockchain. The chain is replicated, cryptographically signed and publicly verifyable at every transaction so that no-one can tamper with the data that has been written onto the blockchain. The blockchain structure is an append-only data structure, such that new blocks of data can be written to it, but cannot be altered or deleted. The blocks are chained in such a way that each block has a hash that is a function of the previous block, providing the assurance of immutability. Whereas the Bitcoin blockchain publishes all elements of the entire chain, in general other types of blockchain can be public, private or consortium based. Public blockchains grant access to read and ability to create a transaction to any user on that network. This type is mostly used for cryptocurrencies (e.g., Bitcoin, Ethereum, Dogecoin and Auroracoin). Consortium blockchain is a "partially decentralized" blockchain [17], where the consensus process is controlled by a pre-selected set of nodes. Imagine a consortium of 15 financial institutions, each of which operates a node of which 10 must sign every block in order for the block to be valid. The right to read the blockchain can be public or restricted to the participants. Private blockchain limits not only the write access but the read access as well, to specific participants who can verify their transaction internally. That makes the transaction on a private network cheaper, since they only need to be verified by few nodes that are trusted and with guaranteed high processing power. Nodes can be trusted to be very well-connected and faults can quickly be fixed by manual intervention, allowing the use of consensus algorithms which offer finality after much shorter block times.

Smart Contracts: Smart contracts are trackable and irreversible applications that execute in a decentralized environment (e.g., blockchain). Once the smart contract has been deployed nobody can edit the code or change its execution behavior. Smart contract execution guarantees to bind parties together to an agreement as written. This creates a new powerful type of trust relationship that does not rely on a single party. Smart contracts enable better management for realizing and administering digital agreements because they are self-verifying and selfexecuting.

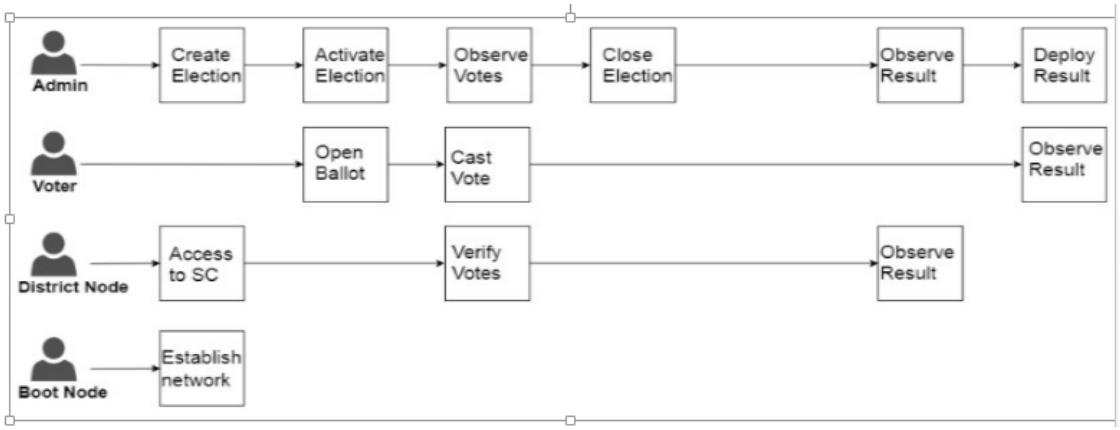


Fig. 1: Election roles and process

Non-Interactive Zero-Knowledge proof: Another concept that is not directly related to blockchain but can be seen as an essential component for satisfying some of the requirements of building an e-voting system on a blockchain is zero-knowledge proof. A zero-knowledge proof is a cryptographical method by which one party, the prover, can prove to another party, the verifier that the prover knows a value x, without revealing any information other than the fact that the verifier knows the value x. A simple example which was first demonstrated live by Konstantinos Chalkias and Mike Hearn Using the example of "Two balls and the colour blind friend", the ZKP works as follows: The prover has two balls, one red and one green, and otherwise identical. The verifier (the friend) is colour-blind. To prove that they are in fact differently coloured, you give the balls to your friend, who hides them behind his back. Your friend then decides whether to switch the balls between hands or not, and then reveals one of the balls. The prover declares if the balls were switched. By repeating this process, the prover can prove that he can correctly identify the balls, as the verifier confirms that the likelihood of repeated success is halved each time. A non-interactive zero-knowledge proof, or NIZKP for short, is a variant of zero-knowledge proofs in which there is no interaction between the prover and verifier. Blum, Feldman and Micali showed that a common reference string shared between the prover and verifier is enough to achieve computational zeroknowledge without requiring interaction. The Fiat-Shamir heuristic however showed that NIZKPs could also be obtained in the random oracle model, which in practice can be used as a cryptographic hash function instead which enables any user to prove their identity and the authenticity of their message without a shared public key. This scheme is ideally suited for microprocessor-based devices such as smart cards, personal computers and remote control systems. The FiatShamir heuristic therefore provides a simple yet efficient and secure method to authenticate and verify eligible individuals for a voting system while guaranteeing voters privacy.

Blockchain as a Service for E-Voting

Election Roles: elections in our proposal enable participation of individuals or institutions in the following roles. Where multiple institutions and individuals can be enrolled to the same role.

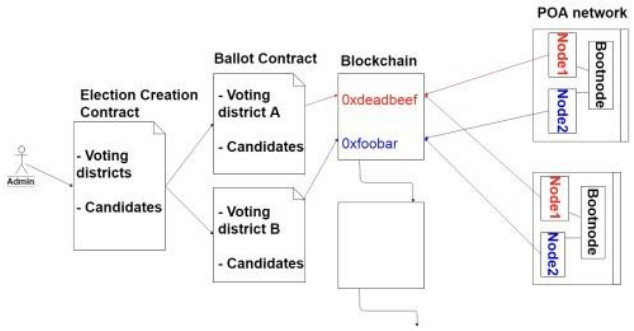
1. **Election administrators:** Multiple trusted institutions and companies are enrolled with this role. The election administrators specify the election type and create aforementioned election, configurate ballots, register voters, decide the lifetime of the election and assign permissioned nodes.
2. **Voters:** For elections to which they are eligible for, voters can authenticate themselves, load election ballots, cast their vote and verify their vote after an election is over. Voters can be rewarded for voting with tokens when they cast their vote in an election in the near future, which could be integrated with a smart city project.
3. **District nodes:** When the election administrators create an election, each ballot smart contracts, representing each voting district, are deployed onto the blockchain. When the ballot smart contracts are created, each of the corresponding district nodes are given permission to interact with their corresponding ballot smart contract. When an individual voter casts his vote from his corresponding smart contract, the vote data is verified by all of the corresponding district nodes and every vote they agree on are appended onto the blockchain when block time has been reached.
4. **Bootnodes:** A bootnode helps the district nodes to discover each other and communicate. The bootnodes do not keep any state of the blockchain and is ran on a static IP so that district nodes find its peers faster.

Election Process: each election process is represented by a set of smart contracts, which are instantiated on the blockchain by the election administrators. A smart contract is defined for each of the voting districts of the election so multiple smart contracts are involved in an election. For each voter with its corresponding voting district location, defined in the voter registration phase, the smart contract with the corresponding location will be prompted to the voter after the user authenticates himself when voting.

Evaluating Blockchain as a Service for E-Voting

The three blockchain frameworks that we consider for implementing and deploying our election smart contracts. Those are Exonum, Quorum and Geth.

Fig. 2: Election as a smart contract



The following are the main activities in the election process:

A. Election Creation

Election administrators create election ballots using a decentralized app (dapp). This decentralized app interacts with an election creation smart contract, in which the administrator defines a list of candidates and voting districts. This smart contract creates a set of ballot smart contracts and deploys them onto the blockchain, with a list of the candidates, for each voting district, where each voting district is a parameter in each ballot smart contract. When the election is created, each corresponding district node is given permission to interact with his corresponding ballot smart contract.

B. Voter Registration

The registration of voter phase is conducted by the election administrators. When an election is created the election administrators must define a deterministic list of eligible voters. This requires a component for a government identity verification service to securely authenticate and authorize eligible individuals. Using such verification services, each of the eligible voter should have an electronic ID and PIN number and information on what voting district the voter is located in. For each eligible voter, a corresponding wallet would be generated for the voter. The wallet generated for each individual voter should be unique for each election the voter is eligible for and a NIZKP could be integrated to generate such wallet so that the system itself does not know which wallet matches an individual voter.

C. Vote Transaction

When an individual votes at a voting district, the voter interacts with a ballot smart contract with the same voting district as is defined for any individual voter. This smart contract interacts

with the blockchain via the corresponding district node, which appends the vote to the blockchain if consensus is reached between the majority of the corresponding district nodes.

D. Tallying Results

The tallying of the election is done on the fly in the smart contracts. Each ballot smart contract does their own tally for their corresponding location in its own storage. When an election is over, the final result for each smart contract is published.

E. Verifying Vote

Each individual voter can go to his government official and present their transaction ID after authenticating himself using his electronic ID and its corresponding PIN. The government official, utilizing district node access to the blockchain, uses the blockchain explorer to locate the transaction with the corresponding transaction ID on the blockchain. The voter can therefore see his vote on the blockchain, verifying that it was counted and counted correctly.

A. Exonum:

Looking at the Exonum blockchain, it is robust end to end with its full implementation done with the programming language Rust. Exonum is built for private blockchains. It has a customized Byzantine algorithm that is used to achieve consensus in the network. With that consensus algorithm, Exonum can support up to 5000 transactions per second. Unfortunately, the limitation of the framework is that Rust is the only programming language in the current version, which limits the developers to the constructs available in that language. To solve this limitation, Exonum is planning to introduce Java-bindings and platform-independent interface description to make Exonum more developer-friendly in the near future.

B. Quorum:

Is an Ethereum-based distributed ledger protocol with transaction/contract privacy and new consensus mechanisms. It's a Geth fork and is updated in line with Geth releases. Quorum changed up the consensus mechanism and aimed more towards consortium chain- based consensus algorithms. Using this consensus allows it to support from dozens to hundred transactions per second.

C. Geth:

Go-Ethereum or Geth is one of three original implementations of the Ethereum protocol and it runs smart contract applications exactly as programmed without possibility of downtime, censorship, fraud or third party interference. This framework supports development beyond the Geth protocol, and is the most developer-friendly framework of the frameworks we evaluated. The transaction per second(transaction rate) is dependent on whether the blockchain is implemented as a public or private network. Because of these capabilities, Geth was the framework we chose to base our work on, any similar blockchain framework with the same capabilities as Geth should be considered for such systems.

Design and Implementation

To introduce a method of secure authentication, our proposed system is designed to use electronic ID authentication via Auðkenni [25], which is an Icelandic service provider for identity verification. Auðkenni utilizes the Nexus software and RFID scanners. When a user registers for an electronic ID, a user chooses a PIN number for its corresponding ID consisting

of 6 numbers. A user will therefore identify himself in the voting booth by scanning his ID and providing his corresponding PIN number to authenticate himself to the system.

1) Any computer in any voting district can be used by any eligible voter to vote, since the wallet for the corresponding voter has information on which voting district the voter is supposed to vote from. For a user to successfully authenticate, a valid ID and PIN number needs to be presented at a voting district using a card reader and the nexus software.

2) If the authentication is successful, the corresponding smart contract is prompted for the ongoing election. The ballot for the aforementioned election is a smart contract which has a list of the candidates a voter can choose from.

Fig. 3: Voter authenticates himself and casts vote

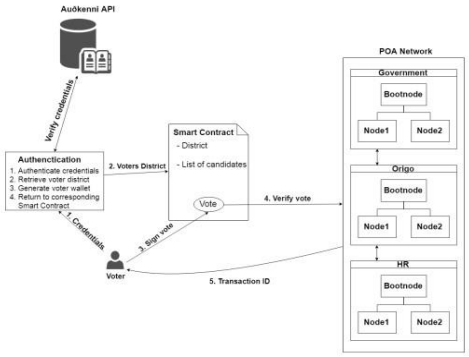
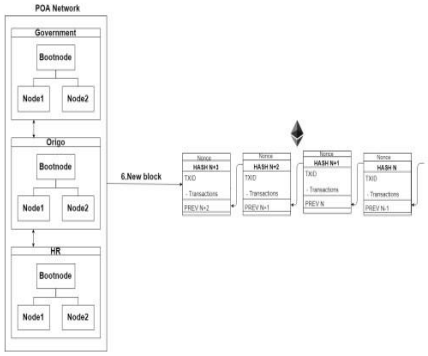


Fig. 4: Block added to the blockchain



3) When a voter has selected a candidate and casts his vote, the voter proceeds to sign his vote by re-entering the corresponding PIN number for his electronic ID.

4) After the voter has signed his vote, the vote data proceeds to be verified by the corresponding district node, which the voter is interacting with the smart contract through. If the aforementioned district node accepts the vote data, the vote data must be agreed upon by the majority corresponding district node.

5) All transactions which were received and verified in the ongoing block time are deployed onto the blockchain after the block time has reached its time limit (see Figure 4). With each new block added to the blockchain, each district node updates his copy of the ledger.

TABLE III: Framwork Evaluation

	Exonum	Quorum	Go-Ethereum
Consensus	Custom-built BFT algorithm	QuorumChain, IBFT and Raft-based consensus	PoW, PoS and PoA
Transactions p/s	up to 5000 transactions p/s	Dozens to hundreds	Depends
Private support	Yes	Yes	Yes
Smart Contract Language	Rust	Solidity	Solidity
Programming Language	Rust	Go, C, JavaScript	Go, C, Javascript
Decentralized	Yes	Partially	Optional

6) If the majority of district nodes agree upon the vote data, consensus for the particular vote has been reached. The user then receives the transaction ID for the corresponding transaction of his vote in the form of a QR-code and the option to print the transaction ID. When the vote is casted and has been verified, a function in the smart contract adds one vote to the party which

was voted for. This functionality of the smart contract structure is utilized to determine the election result in each of the voting districts. Figure 3 is a visual representation of the steps we just elaborated.

Security Analysis and Legal Issues

In this section we analyze the security of the proposed evoting system and the main legal issues

A Security analysis:

1) DDoS: To successfully DDos a distributed system such as we have proposed, the attacker must DDoS every single bootnode in the private network. The individual or institution would be immediately located if that would occur. Each node is implemented with a Byzantine fault tolerance algorithm, which helps locating failed nodes in the system.

2) Authentication vulnerability: Each individual is identified and authenticated by the system by presenting an electronic

ID from Auðkenni and the corresponding 6-digit PIN in the voting booth. Without supervision, an individual could vote for multiple people, if the individual had knowledge of the PIN for each corresponding electronic ID he has. To further address this vulnerability in the near future, a biometric scan could be introduced.

3) Sybil: Sybil attack [26] is known against centralized systems, where an individual creates a large amount of nodes in an attempt to disrupt network operation by hijacking or dropping messages. Since our proposal is running in a private network no individual has the access to create one. Even the consensus protocol that is used in our system is prone Sybil attacks. Private blockchains solve many of today's security problems using strong cryptography features and the limited access to the ledger, without negating the transparency aspect the blockchain technology offers.

B Legal issues:

1) Remote voting: Remote elections provide no coercion resistance because of the non supervised factor in a remote election. Remote elections can therefore not guarantee the privacy that people have when they cast their vote in a voting booth. Family members or a coercer can watch over your shoulder while you're voting, which could lead to a misconfigured results. If elections are hosted on a website for example it could easily be taken down by people with good hacking skills and the mindset to do so. People could identify themselves as another person and therefore vote for another person and even multiple people.

2) Transparency: In the today's election scheme, no method of transparency can be offered to participants of the election. When an individual places his ballot in the box at his voting district, there is no guarantee from the scheme that his vote was counted and counted correctly. Any individual vote can be misplaced, counted incorrectly because of human error or simply because the party which the voter voted for could be disliked by the individual which counted the vote. This transparency is non-existent because no ballot has information on who casted aforementioned vote. To introduce transparency in the process of an election would require a new law which would allow government officials to provide the services which allow such method of transparency.

3) Voter privacy: In every pen and paper election scheme, voters privacy is a key element. The law forbids any individual or entity to be able to know from a single vote, who gave aforementioned vote. If such information could be gathered for each vote, such information could then leak to the public which would allow for listing every single individual who voted

for a single party/candidate. To satisfy the privacy of each voter, no individual vote should be traceable back to the voter.

2. CONCLUSION

The idea of adapting digital voting systems to make the public electoral process cheaper, faster and easier, is a compelling one in modern society. Making the electoral process cheap and quick, normalizes it in the eyes of the voters, removes a certain power barrier between the voter and the elected official and puts a certain amount of pressure on the elected official. It also opens the door for a more direct form of democracy, allowing voters to express their will on individual bills and propositions. In this paper, we introduced a unique, blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost efficient election while guaranteeing voters privacy. We have outlined the systems architecture, the design, and a security analysis of the system. By comparison to previous work, we have shown that the blockchain technology offers a new possibility for democratic countries to advance from the pen and paper election scheme, to a more cost- and time-efficient election scheme, while increasing the security measures of the todays scheme and offer new possibilities of transparency. Using an Ethereum private blockchain, it is possible to send hundreds of transactions per second onto the blockchain, utilizing every aspect of the smart contract to ease the load on the blockchain. For countries of greater size, some measures must be taken to withhold greater throughput of transactions per second, for example the parent & child architecture[28] which reduces the number of transactions stored on the blockchain at a 1:100 ratio without compromising the networks security. Our election scheme allows individual voters to vote at a voting district of their choosing while guaranteeing that each individual voters vote is counted from the correct district, which could potentially increase voter turnout

3. REFERENCES

[1] Sos.ca.gov. (2007). Top-to-Bottom Review | California Secretary of State. Available at: http://www.sos.ca.gov/elections/voting-systems/ oversight/ top-bottom-review/.
[2] Nicholas Weaver. (2016). Secure the Vote Today. Available at:https:// www.lawfareblog.com/ securevote-today.
[3] TechCrunch, (2018). Liquid democracy uses blockchain to fix politics, and now you can vote for it [Online]. Available at: https://techcrunch.com/2018/02/24/liquid-democracy-uses-blockchain/
[4] Geth.ethereum.org. (2018). Go Ethereum. Available at: https://geth. ethereum.org/
[5] Vitalik Buterin. (2015). Ethereum White Paper. Available at: https:// github.com/ethereum/wiki/wiki/White-Paper.
[6] Nca.tandfonline.com. (2015). Pirates on the Liquid Shores of Liberal Democracy: Movement Frames of European Pirate Parties. [Online]. Available at: https://nca.tandfonline.com/doi/abs/10.1080/ 13183222.2015.1017264#.Wr0zCnVl8YR
[7] Feng Hao, P.Y.A. Ryan and Piotr Zielinski. (2008). Anonymous voting by two-round public discussion. Available at: http://homepages.cs.ncl.ac. uk/feng.hao/files/OpenVote_IET.pdf
[8] Feng Hao and Piotr Zielinski. A 2-Round Anonymous Veto Protocol Available at:

[9] The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. Available at: https://users.ece.cmu.edu/~{}adrian/ 731sp04/readings/dcnets.html.
[10] Patrick McCorry, Siamak F. Shahandashti and Feng Hao. (2017). A Smart Contract for Boardroom Voting with Maximum Voter Privacy Available at: https://eprint.iacr.org/2017/110.pdf.
[11] Ronald Cramer, Rosario Gennaro and Berry Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme Available at: http://www.win.tue.nl/~berry/papers/euro97.pdf
[12] Jonathan Alexander, Steven Landers and Ben Howerton (2018). Netvote: A Decentralized Voting Network Available at: https://netvote. io/wpcontent/uploads/2018/02/Netvote-White-Paper-v7.pdf
[13] Agora (2017). Agora: Bringing our voting systems into the 21st century Available at: https://agora.vote/Agora_Whitepaper_v0.1.pdf
[14] Kirill Nikitin, Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, Justin Cappos and Bryan Ford (2017). CHAINIAC: Proactive Software-Update Transparency via Collectively Signed Skipchains and Verified Builds
[15] Available at: https://www.usenix.
[16] org/system/files/conference/usenixsecurity17/sec17nikitin.pdf
[17] Alin Tomescu and Srinivas Devadas (2017). Catena: Efficient Nonequivocation via Bitcoin Available at: https://people.csail.mit.edu/ alinush/papers/catenasp2017.pdf
[18] Michael Del Castillo (2018). Sierra Leone Secretly Holds First Blockchain-Audited Presidential Vote Available at: https://www.coindesk.com/ sierra-leonesecretly-holds-first-blockchain-powered-presidentialvote/
[19] Ethereum Blog. (2018). On Public and Private Blockchains - Ethereum Blog. Available at: https://blog.ethereum.org/2015/08/07/ on-public-andprivate-blockchains/
[20] Bitfury.com. (2018). Digital Assets on Public Blockchains Available at: http://bitfury.com/content/5white-papers-research/bitfury-digital_ assets_on_public_blockchains-1.pdf
[21] Steve Ellis, Ari Juels and Sergey Nazarov. (2017). Chain Link: A Decentralized Oracle Network Available at: https://link.smartcontract.com/ whitepaper
[22] Konstantinos Chalkias, (2017). Demonstrate how Zero-Knowledge Proofs work without using maths
[23] Availableat: https://www.linkedin.com/pulse/ demonstrate-how-zero-knowledge-proofs-workwithout-using-chalkias
[24] Manuel Blum, Alfredo De Santis, Silvio Micali and
[25] Giuseppe Persiano (1988). Non-Interactive Zero-
[26] Knowledge Available at:https://people.csail.mit.edu/silvio/Selected%20Scientif ic%20Papers/
[27] Zero\%20Knowledge/Noninteractive_Zero-
[28] Knowkedge.pdf
[29] Amot Fiat and Adi Shamir. (1986). How to Prove
[30] Yourself: Practical Solutions to Identification and
[31] Signature Problems Available at: https://www.math.uni-
[32] frankfurt.de/~dmst/teaching/SS2012/Vorlesung/ Fiat.Shamir.pdf
[33] Mihir Bellare & Phillip Rogaway (1995). Random Oracles are Practical: A Paradigm for Designing Efficient Protocols Available at: https:// cseweb.ucsd.edu/~mihir/papers/ro.pdf

[34] Ethdocs.org. (2018). What is Ethereum? — Ethereum Homestead 0.1 documentation. [online] Available at: http://ethdocs.org/en/latest/ introduction/what-isethereum.html
[35] Auðkenni.is (2018). [Online] Available at: https://www.audkenni.is/en/
[36] Vincent Gramoli. (2018). On the Danger of Private
[37] Blockchains. [Online] Abailable at:https://www.zurich.ibm.com/dccl/papers/gramoli_ dccl.pdf
[38] Salanfe, "Setup your own private Proof-of-Authority Ethereum network with Geth", Hacker Noon, 2018. Available at: https://tinyurl.com/ y7g362kd.
[39] Jelurida, "Jelurida", 2017. Available at:https://www.jelurida.com/sites/ default/files/JeluridaWhitepaper.pdf