Blockchain E-Voting Done Right: Privacy and Transparency with Public Blockchain

Conference Paper · October 2019

CITATIONS
0

READS
103

4 authors, including:

Kristian Kostal
Slovak University of Technology in Bratislava
8 PUBLICATIONS **15** CITATIONS
SEE PROFILE

Rastislav Bencel
Slovak University of Technology in Bratislava
8 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Michal Ries
Slovak University of Technology in Bratislava
50 PUBLICATIONS **639** CITATIONS
SEE PROFILE

Blockchain E-Voting Done Right: Privacy and Transparency with Public Blockchain

Kristián Košťál, Rastislav Bencel, Michal Ries, Ivan Kotuliak
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Bratislava, Slovakia
{kristian.kostal, rastislav.bencel, michal.ries, ivan.kotuliak}@stuba.sk

***Abstract*—Some forms of voting have been here ever since. Mostly used form all over the world are paper ballots. Electronic voting schemes are being popular only in the last decade and they are still unsolved. E-voting schemes bring problems mainly regarding security, credibility, transparency, reliability, and functionality. Estonia is the pioneer in this field and may be considered the state of the art. But there are only a few solutions using blockchain. Blockchain can deliver an answer to all of the mentioned problems and furthermore bring some advantages such as immutability and decentralization. The main problems of technologies utilizing blockchain for e-voting are their focus on only one field or lack of testing and comparison. In this paper, we present a blockchain-based e-voting platform, which can be used for any kind of voting. It is fully utilized by blockchain and all processes can be handled within it. After the start of the voting, the platform behaves as fully independent and decentralized without possibilities to affect the voting process. The data are fully transparent, but the identity of voters is secured by homomorphic encryption. We have tested and compared our solution in three different blockchains. The results show, that both public and private blockchains can be used with only a little difference in the speed. The key novelty of our solution is a fully decentralized management of e-voting platform through blockchain, transparency of the whole process and at the same time security and privacy of the voters thanks to homomorphic encryption.**

***Index Terms*—blockchain; e-voting; smart contract; Ethereum; Hyperledger Composer; elections; homomorphic encryption**

I. Introduction

The topic of e-voting systems is still at an early stage of development. We have chosen this domain not only for its recency but also because there are not many solutions that address problems of e-voting. Nowadays, popularity grows also in the development of e-Government. However, such a system is not feasible if basic services for citizens such as elections do not become electronic. "E-voting is one of the key public sectors that can be transformed by blockchain technology" [1]. Hand by hand with e-voting come also new challenges, which need to be addressed. One of them is e.g. securing the elections, which needs to be at least as safe as the classic voting systems with ballots. That is why we have decided to create safe elections in which voters do not have to worry about someone abusing the electoral system.

In recent years blockchain is often mentioned as an example of secure technology used in an online environment. Our e-voting system uses blockchain to manage all election processes. Its main advantage is that there is no need for confidence in the centralized authority that created the elections. This authority cannot affect the election results in our system. Another challenge in e-voting is the lack of transparency in the functioning of the system, leading to a lack of confidence in voters [2]. This problem is solved by blockchain in a way of total transparency that allows everyone to see the stored data and processes such as how are these data handled. In the field of security, this technology is more suitable in every way than the classic e-voting platform without blockchain.

The article is structured as follows. Section II shows a brief analysis of existing blockchain e-voting solutions. In Section III we present the design of our solution and describe all of its components. The evaluation is in Section IV and the results are discussed in Section V with further conclusions.

II. Related Work

Blockchain is a distributed database of records, often referred to as a ledger. The basic principle of blockchain is in the immutability of the records already written in blocks. Advanced cryptography is used to ensure the chaining of blocks, providing data integrity. Another feature is the type of network communication. Network nodes communicate with a client-client method. There is no third party to communicate between clients and therefore no trust in this person is required. The network participant's true identity is unknown [3]. In this section, we will analyze some of the existing solutions for e-voting systems based on blockchain technology.

A. Agora

The only successful project that was partially demonstrated in the state elections was created by the Swiss company Agora [4], which has the best technology and research in this area. The elections took place in the western part of Sierra Leone [5]. They were not completely decentralized and could not meet the eligibility requirements for elections via the Internet in the given area. The choices were made as regular options with paper ballots. Identity of citizens was verified by ID card and then they put their selection into the ballot box. These ballots were then manually inserted into a private blockchain Bulletin Board developed by Agora. A private blockchain differs from a public, such as Ethereum, in that, which nodes are allowed to validate transactions and blocks. In private,

these operations can be only performed by trusted nodes that have been previously recognized as trusted. Other nodes, ordinary network users, are part of the network and can see all the data if allowed to. In the mentioned election method, there was no need to address the problems of anonymity of voters. Paper ballots were counted by officials of the State Election Commission. The number of paper votes was compared with the number of votes on the blockchain. Thus, the final counting of the votes did not proceed through blockchain. However, the solutions that Agora delivers have a perspective in this regard. Agora offers solutions for elections to governments and institutions. Agora uses its token to make choices. Governments must pay for each citizen a pre-determined amount for which a sufficient number of tokens for election is allocated to the voter. The electoral system is a multi-layer architecture using a blockchain, called Bulletin Board, which is based on Skipchain architecture. Bulletin Board data are cryptographically bounded to Bitcoin blockchain [6] via the Cotena layer that provides immutability and decentralization of stored data. Bulletin Board uses nodes recognized as authorities (cothority). These confirm transactions where each such node in the network has a full copy of all transactions. The Cotena layer is a change-resistant mechanism built on the Bitcoin blockchain. This layer provides the integrity of data layer records to a decentralized system. Cotena has been designed to take advantage of Bitcoin's blockchain security and has introduced a design that has low memory requirements and low fees for using Bitcoin transactions. Skipchain allows software clients to navigate through large numbers of blocks both forward and backward. In doing so, it provides proof of transaction validity without the need for a full copy of the records contained in the blockchain.

B. Netvote

Netvote is a decentralized application based on blockchain technology for elections and works on the Ethereum network [7]. Netvote provides an environment with decentralized applications (dApp) for network users. DApp designed for admin allows to create elections, set election rules, set voter registration rules, create ballot boxes and setup voting. DApp for voters allows them to sign up for elections and vote for the selected candidate. After closing the elections, you can view the results using the appropriate application. The application allows the administrator to choose one of three types of voting. The first type is open elections where everyone who has an account in the Ethereum network can vote. The second type is private elections where only authenticated voters can vote. The last option allows only voters who have the required amount of correct tokens issued specifically for the elections. In Netvote, each election consists of multiple smart contracts deployed on the Ethereum network. These smart contracts are created by an administrator through his dApp. Every electoral district is a smart contract. The smart contract includes a list of ballots also represented by separate smart contracts. The voter registers in a given electoral district and all actions subsequently performed through his dApp are processed in that district. Netvote, as we mentioned, includes the option of private elections where only authenticated voters can participate in. The solution that Netvote brings is called the Vote Gateway and serves to verify voters' identity. The voter sends his signed ballot through his dApp. If the voter is registered, the Vote Gateway selects the voter's private key from the vault where this unique private key is stored. Subsequently, the SHA3 cryptographic function is executed along with the ballot and the private key to create the anonymous identifier of the voter. A ballot mapped to an anonymous identity is sent to the electoral district via a transaction [8]. Netvote is a good solution for both state elections and institutions. The drawback is the need for a lot of smart contracts and transactions which highers the costs of deployment and also brings scalability issues. Netvote uses Ethereum's public blockchain. The architecture of Netvote needs to be refactored. Maybe using private blockchain elections where the number of transactions processed per second increases significantly is also interesting.

C. OV-net

The OV-net (Open voting network) [9] is a 2-round decentralized electoral protocol implemented on Ethereum blockchain. This protocol has several advantages. One of them is counting the votes that the protocol does itself without the necessary authority. Privacy is maximized. The only time a voter's choice might be revealed is if all the other nodes in the network are fraudulent. Each user can check others for compliance with the protocol. The protocol consists of five parts: 1. Setup - The election's administrator is responsible for uploading a valid voters' list to the smart contract when it is started. 2. Signing up - Voters will send their electoral key and use Zero-knowledge proof (ZKP) to confirm the electoral key. Ethereum confirms the correctness of the ZKP and stores the electoral key. 3. Voting - Voters will send an encrypted vote. This vote can be either 1 (yes) or 0 (no). Ethereum verifies that the vote is only one of the options 1 or 0, and then stores it. 4. Voting Count - Ethereum counts the votes when all of them have been sent. OV-net is a well-managed protocol that supports multiple necessary functions required for e-voting. However, it has several disadvantages. The only voting options are yes or no. The whole implementation is based on the principle of two options for voting for the proper functioning of the ZKP. In our application, we need to make choices for multiple candidates with their names. OV-net is also not eligible for a large number of voters because of its specific implementation. Another disadvantage is the need to vote for each voter. If one voter did not vote, then the elections could not be evaluated or counted.

Several schemes for e-voting have been published in recent years, but most do not have documentation and they lack information about the internal operation of the service. One of the services is Follow my Vote. This system does not answer multiple questions about how it works in a blockchain network. Another service is BitCongress, which was designed to work with multiple protocols like Bitcoin and Mastercoin. Finally, both applications were not put into operation.

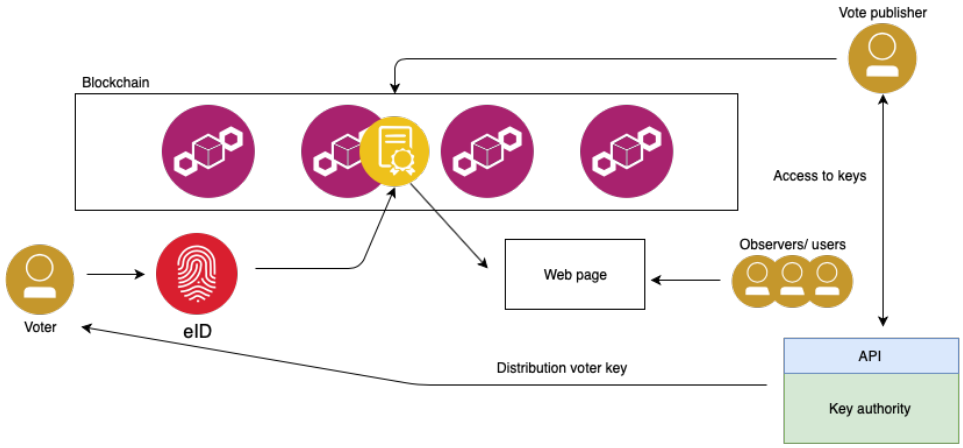

Fig. 1. High-Level architecture

III. Design

The proposed blockchain voting system considers all requirements for voting and is designed generally for any elections e.g. president, student parliament, etc. The system allows more round elections and preferably uses a public blockchain. The public blockchain can be replaced by other types of blockchain but the stored data (votes) have to be easily verified by any user. The user represents any observer who is interested in the blockchain voting.

In our proposed system we identify three main roles: vote publisher; key authority; and voter. These three roles can represent an organization, a company, or a user. The roles vote publisher and key authority can be grouped to one role due to that they can be the same organization or person. The voter attends the elections depending on vote configuration. The configuration of the votes is performed by the vote publisher and is included in the smart contract. The vote publisher has to know all cipher keys before publishing the smart contract. The close collaboration between the vote publisher and the key authority is required. The key authority creates and distributes all cipher keys to a voter and vote publisher. The distributing channel has to be secured and should not be vulnerable to any 3rd party.

The high-level voting system architecture is shown in Figure 1. There are illustrated roles, components, and interconnections between them. The architecture contains the following components: blockchain (required), eID (required), results interface (required), key authority API (optional). A specialized component is a smart contract which is part of the blockchain and is responsible for vote processing and evaluation. The programming language of the smart contract depends on the type of blockchain.

Key authority API is an optional module which can be part of the key authority organization. This API serves vote publisher and voter to obtain public key for homomorphic encryption and a key to access the votes. These keys can be distributed manually or in a different way. It is the reason why the component is only optional in the architecture and is not required.

A. Blockchain

The blockchain component represents the whole architecture for storing data and performs voting operations. The blockchain can be built by technologies of a public blockchain, e.g. Ethereum or a private blockchain, e.g. Hyperledger. The advantages of the public blockchain are that it provides all information about transactions and blocks to all users, which is the reason to give it greater confidence compared to the private blockchain. This confidence is in the context of a regular user, which is not advanced in technologies and wants to see all information. The private blockchain can provide the same level of confidence, but an organization has to prove it by showing data. The proposed architecture does not restrict what blockchain should be used. Both types of blockchain can provide the same level of confidence. The platform which is used is the decision of blockchain organization.

B. Smart Contract

Security of the voting is based on blockchain and processing is performed by a smart contract, which is a part of the blockchain. The smart contract is published to the blockchain network after a configuration. The configuration contains times, candidates or other properties. The candidate does not have to be a person, it represents anything, which is the object of the elections. The published smart contract cannot be modified or changed, which contributes to the transparency of the voting. The smart contract contains an access list of users who are eligible for voting. The access list has to follow a key distribution, which is realized by the key authority.

The data are stored in a blockchain and are encrypted by homomorphic encryption [10]. The smart contract has to contain the public key of the homomorphic encryption to make the encrypted results of the elections. The private key is stored separately and is used to see results after the voting end. This key can be distributed to multiple parties which are responsible for voting evaluation. Nobody knows the ongoing

results, because they are visible only after the end. To improve security we used Zero-knowledge proof [11] to verify that the votes contain correct values, e.g. voter can give only one vote and cannot give two or more.

C. eID

This component provides access to blockchain to do the voting. In the architecture, we consider eID, which works as access to the blockchain network. The ID card contains public and private keys, which are released by the key authority to the user. The public key represents the standard public address of wallet in the blockchain. The private key is known only by the voter.

D. Results interface

This component represents an interface with results. The interface has to have access to blockchain and also gives information to observers and users. The data contains vote results and there should be access to see transactions on the blockchain as it should have all transactions visible. The results are there in graphic form for easy understanding and only final results are visible. Live results are not available due to the usage of homomorphic encryption.

IV. Evaluation

After the design phase, we have decided to implement the desired service. To make sure that our application is working, we have created two test scenarios. The main goal was to test the overall functionality, security, and speed of the proposed solution. Our testing environment consisted of a MacBook Pro with macOS Mojave (10.14.3), Chrome web browser with enabled JavaScript and installed MetaMask plugin. MetaMask provides a simple interface for communication with the Ethereum network. For testing purposes, we have created our local Ethereum network using Ganache. When all automatic truffle tests passed, we deployed the smart contract to Ropsten test network.

The first test scenario covers basic unit tests, which checked the smart contract and its behavior for mistakes. It was implemented with 1000 virtual test users, who did normal voting as real people would do. Furthermore, there were test cases for double voting, unauthorized voting, looking at the results after the end of elections, and second-round voting for candidates, who progressed from the first round if it was unsuccessful, i.e. none of the candidates got 50% or more.

The second test scenario was for measuring the time needed for voting by a real person. The scenario consisted of opening the web application for voting, voting for the preferred candidate, adding the vote into blockchain and registering the vote. The test was done by implementing smart contract into three different blockchains: Ganache as local Ethereum network, Hyperledger Composer as a private blockchain and Ropsten as a live Ethereum test network. In the background, there was a running script for automatic voting of 15 persons each second. The reason for running this script was that we tried to simulate real conditions, which can happen in real elections. In every blockchain, the test scenario was repeated five times and the test results can be seen in Table I. Note that the difference in times of Ropsten network is affected by average block time in it, which is around 12 seconds.

TABLE I
Comparison of voting time in different blockchains

Test nr.	**Time in seconds**		
	Ganache	***Hyperledger Composer***	***Ropsten***
1	6.34	6.12	17.34
2	6.25	5.97	17.93
3	6.40	6.04	18.05
4	6.39	6.15	17.98
5	6.22	5.96	17.47

V. Conclusions

Although we can see slight differences in network times, they are so negligible that public blockchain has more advantages in such an electoral system due to its openness of data and that anyone can watch them in the real time. A private blockchain is a bit faster, but it reduces the credibility of the whole system by being partially centralized because it only runs where the authority wants it. The table shows that the average times to add one person's voice are: Ganache 6.32 s (median 6.34 s), Hyperledger Composer 6.05 s (median 6.04 s), and Ethereum Ropsten 17.75 s (median 17.93 s). These times are influenced by the used consensus algorithm and also by the block time.

Acknowledgment

This research was supported by the Ministry of Education, Science, Research and Sport of the Slovak Republic, Incentives for Research and Development, grant agreement number: 2018/14427:1-26C0. It is also a part of APVV-15-0731 project. The authors would like to thank for financial contribution from the STU Grant scheme for Support of Young Researchers.

References

[1] N. Kshetri and J. Voas, "Blockchain-Enabled E-Voting," *IEEE Software*, vol. 35, pp. 95–99, jul 2018.

[2] M. Pawlak, J. Guziur, and A. Poniszewska-Marańda, "Voting Process with Blockchain Technology: Auditable Blockchain Voting System," in *Lecture Notes on Data Engineering and Communications Technologies*, pp. 233–244, Springer, Cham, 2019.

[3] B. Singhal, G. Dhameja, and P. S. Panda, "How Blockchain Works," in *Beginning Blockchain*, pp. 31–148, Berkeley, CA: Apress, 2018.

[4] Agora, "Agora Whitepaper," 2018.

[5] R. Perper, "Sierra Leone is the first country to use blockchain during an election - Business Insider," 2018.

[6] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," tech. rep., 2008.

[7] G. Wood et al., "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[8] S. Landers, "Netvote: A Decentralized Voting Platform - Netvote Project - Medium," 2018.

[9] P. McCorry, S. F. Shahandashti, and F. Hao, "A Smart Contract for Boardroom Voting with Maximum Voter Privacy," in *Lecture Notes in Computer Science*, ch. FCDS, pp. 357–375, Springer, Cham, 2017.

[10] Z. Brakerski and V. Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard) LWE," *SIAM Journal on Computing*, vol. 43, pp. 831–871, jan 2014.

[11] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *Journal of Cryptology*, vol. 7, no. 1, pp. 1–32, 1994.