Towards Secure E-Voting Using Ethereum Blockchain

Ali Kaan Koç
Computer Engineering Dept.
Dokuz Eylul University,
Izmir/Turkey
kaan.koc@ceng.deu.edu.tr

Emre Yavuz
Computer Engineering Dept.
Dokuz Eylul University,
Izmir/Turkey
emre.yavuz@ceng.deu.edu.tr

Umut Can Çabuk
Electrical-Electronics Engineering Dept.
Erzincan University,
Erzincan/Turkey
ucabuk@erzincan.edu.tr

Gökhan Dalkılıç
Computer Engineering Dept.
Dokuz Eylul University,
Izmir/Turkey
dalkilic@cs.deu.edu.tr

***Abstract* — There is no doubt that the revolutionary concept of the blockchain, which is the underlying technology behind the famous cryptocurrency Bitcoin and its successors, is triggering the start of a new era in the Internet and the online services. While most people focus only at cryptocurrencies; in fact, many administrative operations, fintech procedures, and everyday services that can only be done offline and/or in person, can now safely be moved to the Internet as online services. What makes it a powerful tool for digitalizing everyday services is the introduction of smart contracts, as in the Ethereum platform. Smart contracts are meaningful pieces of codes, to be integrated in the blockchain and executed as scheduled in every step of blockchain updates. E-voting on the other hand, is another trending, yet critical, topic related to the online services. The blockchain with the smart contracts, emerges as a good candidate to use in developments of safer, cheaper, more secure, more transparent, and easier-to-use e-voting systems. Ethereum and its network is one of the most suitable ones, due to its consistency, widespread use, and provision of smart contracts logic. An e-voting system must be secure, as it should not allow duplicated votes and be fully transparent, while protecting the privacy of the attendees. In this work, we have implemented and tested a sample e-voting application as a smart contract for the Ethereum network using the Ethereum wallets and the Solidity language. Android platform is also considered to allow voting for people who do not have an Ethereum wallet. After an election is held, eventually, the Ethereum blockchain will hold the records of ballots and votes. Users can submit their votes via an Android device or directly from their Ethereum wallets, and these transaction requests are handled with the consensus of every single Ethereum node. This consensus creates a transparent environment for e-voting. In addition to a broad discussion about reliability and efficiency of the blockchain-based e-voting systems, our application and its test results are presented in this paper, too.**

Keywords — blockchain; ethereum; smart-contracts, e-voting.

I. Introduction

Blockchain technology that shines like a star after the entrance and widespread acceptance of Bitcoin [1], the very first cryptocurrency in peoples' everyday life, has become a trending topic in today's software world. At the beginning, Blockchain was only used for monetary transactions and trade, but studies have started to suggest that it can be used in many more areas over time, because there is a high degree of transparency in this system. For example, in Bitcoin, since the wallets are in a distributed structure, the total amount of coins and instant transaction volume in the world can be followed momentarily and clearly. There is no need for a central authority to approve or complete the operations on this P2P-based system.

Because of that, not only the money transfers but also all kinds of structural information can be kept in this distributed chain, and with the help of some cryptological methods, the system can be maintained securely. Like people's assets, marriage certificates, bank account books, medical information, etc., a lot of information can be recorded with this system with relevant modifications [2]. Ethereum coin (Ether), another cryptocurrency with multipurpose development environments, which emerged a few years after Bitcoin, distinguishes the blockchain in a real sense, revealing that this technology can produce software that can hold information that is structured as described above. The software programs enforced by smart contracts [3] (explained later) are written into the blockchain and are immutable. They cannot be (illegally) removed nor manipulated once written. Hence, they can work properly, autonomously and transparently forever, without any external stimuli [4].

As already mentioned, with its unique distributed and secure concept, the blockchain technology may address many issues other than digital trade. It might be a very suitable solution for e-voting projects. E-voting is being

978-1-5386-3449-3/18/$31.00 ©2018 IEEE

studied extensively, and many implementations are tested and even used for a while. However, very few implementations are reliable enough and are still in use. Of course, there are many successful examples of online polls and questionnaires, yet we cannot claim the same for online elections for governments and businesses. That's mainly because, official elections are essential elements of the democracy and democratic administrations, which are the most preferred administrative methodology in the modern world. More, what is most valued in democratic societies is a robust electoral process that provides transparency and privacy. Today, a lot of decisions are being made by people (and members in organizations). Means of such voting systems are used in a lot of fields ranging from the law and act referendums to the TV shows.

While most government elections and many organizational elections are held physically using sealed paper ballots, other polls and questionnaires are usually made on the Internet or SMS channels, notarized accounts are counted and publicly announced. But, legacy paper-to-box voting systems create some questions; How reliable are the notaries at hand? How can we be sure that the votes people gave are not changed before they are counted on the system? How can we verify the transparency of the system? How can we prevent the tricks that reduce people's trust in the polls? How expensive is to hold an election in one vote center with 1000 voters, including material, logistics and salary costs? What about 1000 vote centers and 1,000,000 voters? And repeating all the setup for each election, considering there are a few each year? These and other similar problems have gradually entered a growth trend.

II. Motivation and Related Works

Our main motivation in this project is to provide a secure voting environment and show that a reliable e-voting scheme is possible using blockchain. Because, when e-voting is available for everyone who has a computer, or a mobile phone, every single administrative decision can be made by people and members; or at least people's opinion will be more public and more accessible by politicians and managers. This will eventually lead humanity to the true direct democracy [5]. It's important for us since elections can easily be corrupted or manipulated especially in small towns, and even in bigger cities located in corrupt countries. Plus, large-scale traditional elections are very expensive in the long term, especially if there are hundreds of geographically distributed vote centers and millions of voters [6]. Also, the voters (mainly for members of organizations) might be on vacation, on a business trip or far away for any other reason, which will make impossible for that particular voter to attend the election and may lower the overall attendance. E-voting will be able solve these problems, if implemented carefully.

The concept of e-voting is significantly older than blockchain. So that, all known examples so far used means of centralized computation and storage models. Estonia is a very good example, since the government of Estonia is one of the first to implement a fully online and comprehensive e-voting solution [7]. The concept of e-voting was started to be debated in the country in 2001 and officially started by the national authorities in the summer of 2003 [8]. Their system is still in use, with many improvements and modifications on the original scheme. As reported, it is currently very robust and reliable. They use smart digital ID cards and personal card readers (distributed by the government) for person-wise authentication [9]. For citizens to attend the elections by listing the candidates and casting a vote, there is a special web portal as well as an equivalent desktop app. So that, anyone having a computer and Internet connection and also his/her ID card, can easily vote remotely. People can also digitally create petitions and proposals for acts and laws at the parliament's website (http://rahvaalgatus.ee). These petitions can be digitally signed using the smart ID card by any citizen who wants to support the proposal. If proposals achieve a certain number of signatures, they are discussed in the parliament. That's another good example showing how technology can strengthen the democracy. Though being considerably successful and reaching nearly 30% penetration rate during recent elections, the Estonian model has some drawbacks, too. The centralized solution, by its nature, creates a single-point-of-failure and is open to hacking/hijacking attempts. In example, Distributed Denial of Service (DDoS) attacks can harm the software, servers or databases used. The administrators of such a system may act malicious and steal, if cannot manipulate, some valuable information during an election. The scalability of this system is another question. Since Estonia has a relatively small population, it is hard to estimate if such a system would work flawlessly in, say, China. The constant need for the ID card and the reader device is not nice, too, due to the extra cost of producing, distributing, and carrying (for voters) them.

Switzerland is another one of the few countries participating in the electronic voting trend. In Switzerland, known for its widespread democracy, every citizen who completes the age of 18 can take an active or passive role in the elections, which may be held in many different topics for many different decisions. They have also begun an official work on a voting system called remote voting [10].

There are other similar commercial or experimental works found on the Internet that aim to address that issue such as https://followmyvote.com/. There, voters are declaring their votes anonymously and they count anonymous votes and apply their mathematical formula, because they know there can be fake anonymous votes and they also know that not everyone in election declared who they vote, that's why they put a margin to the percentage of results. However, it does not show true transparent results. Although being a promising attempt, it is currently far from being a solid solution.

As an online polling example, rather than an e-voting system, http://www.strawpoll.me/ is a popular and free service. It's a simple website that allows everyone to create questionnaires and allows answering others' polls with votes. It shows how powerful can be e-voting, because everyone easily accesses the election and uses his/her votes and declares his/her choice. People can share private hyperlinks to any created poll (as long as they know the link) and people who have the link can vote and one browser can only use one vote. The security here, in terms of voter authentication, duplicate votes and non-repudiation of votes, is very weak. http://www.strawpoll.me/ trusts people about they will not violate the election process while benefiting ease of access and using features of e-voting. Hence, it cannot be used in real cases such as choosing the chairman of a department, etc.

Another example of e-voting process is implemented at https://electionrunner.com/, they have a mobile application and a web platform that people can create and share elections with other users. People can define who will vote in this election and how long it will last and then they share this election to authenticated subscribers of electronic runner. However, one still has to trust the central authority in Electronic Runner Inc. It is still one step away from being a 100% transparent and efficient e-voting platform.

A very comprehensive research paper proposes a solid methodology for a blockchain-based e-voting system [11]. The authors also considered countermeasures for voter privacy and vote anonymity, using an intermediate unit between the voter (wallet) and the candidate (wallets) as well as using two different coin types for these intermediate coin (vote) transfers. Here, the coins (votes) sent by the voters are collected by the intermediate unit and converted to another currency using that currency's wallet. Then the intermediate unit sends the new coins to their original destinations (candidates). Although it is a very informative source, it does not contain much information regarding the implementational aspects apart from use of Bitcoin and Zerocoin as the currencies, nor provide a broad discussion about it.

Our primary goal is to focus on implementational works and build our solution in a smaller scale to make our university election process online such as: department chairs, university rector, or student councils elections. We would like to do it in a way that everyone can check and keep track of the election process and election process will be completely online so that everyone may attend voting easily in university's elections. Our primary contribution to the online elections concept is integrating them with the Ethereum blockchain platform. At the time of writing, there are only a few academic works covering the Ethereum blockchain as an e-voting solution. In [10], authors have proposed a comprehensive and so-called secure protocol using the Ethereum blockchain, but their protocol includes complex mathematical operations and thus, requires vast computational power, so is not Internet of Things (IoT)-friendly. We built Ethereum smart contacts that allow to check and count the votes when the time of the election is over. Our contract has functions to set the time and duration of an election, such as: 120 minutes or 3000 minutes. Also, we can include any Ethereum account to the elections. By using the accounts' hash values, people's identity cannot be revealed. Personal authentication is considered a different sub-problem and left out of the scope of this study, as well as legal regulations.

III. Implementation and Discussion

In our study, Ethereum environment is preferred as the development platform and the blockchain network. That is because, while Bitcoin is only intended to validate coinage transactions, Ethereum network provides a broader range of use cases, with the power of smart contracts. Many applications, that may normally require a web server, can be run through these smart contracts, without using a server. Thus, it is very hard, if not impossible, to manipulate or harm the source codes of the intended software.

In the Ethereum network, all operations are (at least supposed to be) in real time, and all the blocks are written in the ultimate chain in exchange for some Ethers (the currency of the Ethereum network). These are given as prize to the miners, who execute these writing and validation operations, which are costly in terms of computation time and power. We have defined our smart contracts as we briefly mentioned above. These contracts are written in Solidity programming language, which is a combination of C++ and JavaScript. Smart contracts are executed by the peers of the Ethereum network in every 15 seconds, and they should be validated at least by 2 other users to be activated. After that, functions of contracts can be executed, and contracts can be shared with other candidates.

To be able to hold completely online elections, we need to solve the following problems. We need transparency, authentication and provability in the voting platform. We need to assure that the people who attend the elections are real people and use correct credentials that we know in electronic environments, and we should be able to prove that any time, also we need our elections are 100% transparent as desired. So, we need to gather and check signed and timestamped data of the elections. Because, nobody should be able to change the votes after they are casted. Also, we need individuality in elections, so that nobody can vote for someone else.

These issues can be solved by using the blockchain peer-to-peer technology. We can define the required self-executable smart contracts in the blockchain. It is as same as writing code, we define rules, objects, data models, and thus contracts can start to execute. After smart contracts are initialized, they cannot be discarded from the blockchain, and people can look back whether results of execution of smart contracts are true or not. In Ethereum network, there is no need for a central authority to provide the proof-of-work. All peers can calculate the results of the contracts without any interference. The Ethereum network is able to provide self-tallying [10].

At any rate, use of the original Ethereum network for testing of experimental software related to the development of new smart contracts is costly (since requires spending some Ethers) and unnecessarily occupies vast memory in the system. Hence, private Ethereum networks are created and made available to the developers to allow them to test their software without congesting the original network. One of them is the Rinkeby network, which we also used in our project (https://www.rinkeby.io/). At the time of writing, it has more than 1.5 million blocks and gives 20 (replica)

Ethers to its users to spend during tests. It should be noted that, such test networks may have other implicit or explicit rules or limitations. For example, the Rinkeby network forces its users to download all of the existing blocks in the network and to contribute to the pool of the total computational power. Such a rule does not exist in the original network. In order to use a test network, users should download a legit Ethereum wallet from the Ethereum website and change the connected network to the chosen test network, using the settings menu.

In the code given in Fig. 1 shows the variable definitions. The “Voter” is defined as a struct in the Solidity programming language. We defined Voter and collected Voters in an array. Voters have some properties, and may have much more depending on the use case scenarios. The variable “isVoted” is a flag indicating that whether the voter has casted his vote already, or not yet. “vote” variable, likewise, stores the choice of the voter among all the candidates, called proposals in a broader sense (defined as the Proposal struct). ID is an Ethereum wallet address associated with a voter account in the Ethereum network. For instance, a voter may have the following values for their properties:

ID: 0xF40877673bbea2C068c6721A2E43DF5AE8771C23
isVoted: true
vote: 2

```
address chairPerson;
struct Voter {
        bool isVoted;
        bool hasRightToVote;
        uint8 vote;
        address ID;
    }
struct Proposal {
        uint voteCount;
    }
```

Fig. 1. Code block to define the structs and variables.

The address-typed variable “chairperson” stores the wallet address of the responsible person, who is the main administrator of the voting process. Though he/she cannot manipulate an ongoing (or finished) voting, he/she has the right to initialize the voting process and the Voter objects, which will be assigned to real voters. How this has been implemented is given in the Fig. 2. Other responsibilities of the administrator may be related to non-technical issues, if any.

In Fig. 2, giveRightToVote function can be seen. Owner of the contract, who was declared once during the construction of the contract is held in the chairperson variable. This function can only be executed by the owner of the contract. This property can be checked with a basic if statement. Then, we give the voting right to an eligible Voter’s (wallet) address. Example as follows:

giveRightToVote(0xDF69B68b00A3a4e6F907eD353467bAC068aF0717);

The person, who has that Ethereum address, imposed by the chairperson, has the right to vote within this contract.

```
function giveRightToVote(address toVoter) public {
        if (msg.sender != chairPerson ||
voters[toVoter].isVoted){
            return;

        }
        else{
            voters[toVoter].hasRightToVote = true;
        }

    }
```

Fig. 2. Code block of the function that initialize voters.

We wrote the vote() function, given in the Fig. 3, that can be executed by every voter, whenever they want to attend the voting (until the deadline). Voters just send the id of the proposal, which they want to vote, as a parameter and their votes are hence recorded. This function firstly detects who currently is trying to execute that function of the contract. More, if the person has the right to vote, and casted his/her vote, thereafter the person is marked as already isVoted, and the vote count of the candidate (proposal) of his choice is incremented by one or by another number based on his/her voter weight.

```
function vote(uint8 toProposal) public {
 Voter storage sender = voters[msg.sender];
 if (sender.isVoted || toProposal >=
proposals.length && !sender.hasRightToVote)
return;
 sender.isVoted = true;
 sender.vote = toProposal;
 proposals[toProposal].voteCount += 1;
}
```

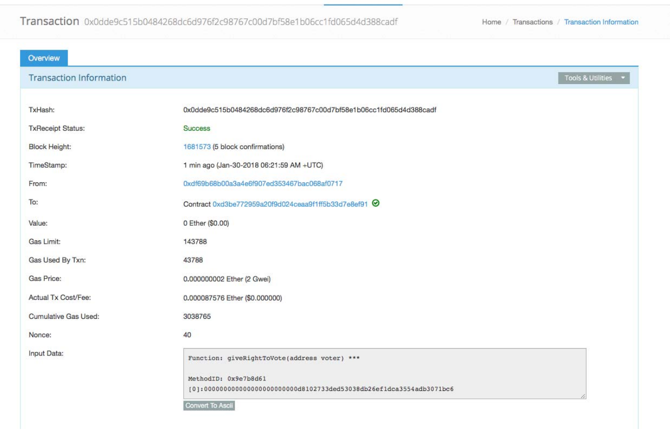
Fig. 3. Code block defining the vote casting process.

winningProposal() function, presented in Fig. 4, returns the id of the winning candidate in the winningProposal variable. It doesn’t finish the voting process itself, but it returns the winning proposal every time it is executed. This function checks every proposal, counts the votes and then returns the one, who is the winner of the whole voting process as of the execution time, since it doesn’t end the election.

```
function winningProposal() public constant returns
(uint256 _winningProposal) {
 uint256 winningVoteCount = 2;
 _winningProposal=0;
 for (uint8 prop = 0; prop < proposals.length;
prop++)
 if (proposals[prop].voteCount > winningVoteCount)
 {
  winningVoteCount = proposals[prop].voteCount;
  _winningProposal = prop;
 }
}
```

Fig. 4. Code block for returning the results of the voting.

Fig. 5 and 6 show detailed records of the entries (blocks) regarding the vote creation and casting operations written in the blockchain. This information is publicly available to everyone tracking the network.



Transaction 0x0dde9c515b0484268dc6d976f2c98767c00d7bf58e1b06cc1fd065d4d388cadf

TxHash: 0x0dde9c515b0484268dc6d976f2c98767c00d7bf58e1b06cc1fd065d4d388cadf

TxReceipt Status: Success

Block Height: 1681573 (5 block confirmations)

TimeStamp: 1 min ago (Jan-30-2018 06:21:59 AM +UTC)

From: 0xdf69b68b00a3a4e6f907ed353467bac068af0717

To: Contract 0xd3be772959a20f9d024ceaa9f1ff5b33d7e8ef91 ✓

Value: 0 Ether ($0.00)

Gas Limit: 143788

Gas Used By Txn: 43788

Gas Price: 0.000000002 Ether (2 Gwei)

Actual Tx Cost/Fee: 0.000087576 Ether ($0.000000)

Cumulative Gas Used: 3038765

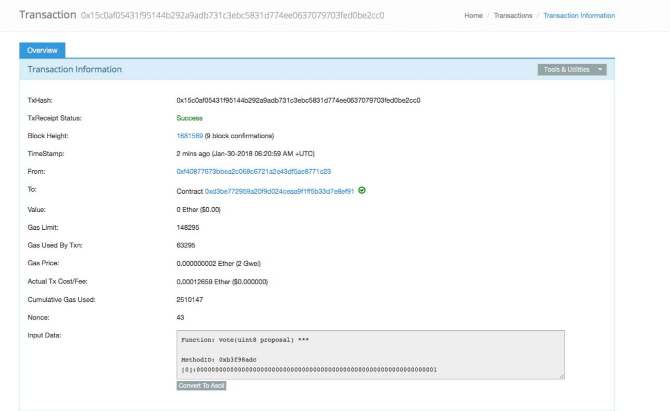
Nonce: 40

Input Data:

```
Function: giveRightToVote(address voter) ***

MethodID: 0x9e7b8d61
[0]:000000000000000000000000d8102733ded53038db26ef1dca3554adb3071bc6
```

Fig. 5. Screenshot of a voting creation entry in the chain.



Transaction 0x15c0af05431f95144b292a9adb731c3ebc5831d774ee0637079703fed0be2cc0

TxHash: 0x15c0af05431f95144b292a9adb731c3ebc5831d774ee0637079703fed0be2cc0

TxReceipt Status: Success

Block Height: 1681569 (9 block confirmations)

TimeStamp: 2 mins ago (Jan-30-2018 06:20:59 AM +UTC)

From: 0xf40877673bbea2c068c6721a2e43df5ae8771c23

To: Contract 0xd3be772959a20f9d024ceaa9f1ff5b33d7e8ef91 ✓

Value: 0 Ether ($0.00)

Gas Limit: 148295

Gas Used By Txn: 63295

Gas Price: 0.000000002 Ether (2 Gwei)

Actual Tx Cost/Fee: 0.00012659 Ether ($0.000000)

Cumulative Gas Used: 2510147

Nonce: 43

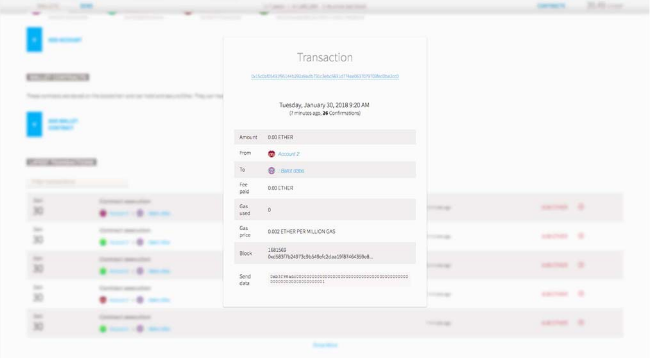
Input Data:

```
Function: vote(uint8 proposal) ***

MethodID: 0xb3f98adc
[0]:0000000000000000000000000000000000000000000000000000000000000001
```

Fig. 6. Screenshot of a vote casting entry in the chain.

Fig. 7 and 8, show a receipt of a casted vote and a query for the results of the voting, respectively. The information found in these screenshots given in Fig. 7 and 8, on the other hand, are only available to the voters. Because, these records are (and can only be) directly obtained from a voter's wallet account.



Transaction

Tuesday, January 30, 2018 9:20 AM
(7 minutes ago, 26 Confirmations)

Amount 0.00 ETHER

From Account 2

To : Ballot d3be

Fee paid 0.00 ETHER

Gas used 0

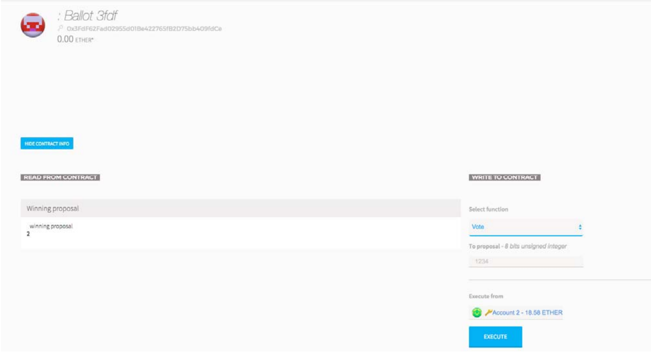
Gas price 0.002 ETHER PER MILLION GAS

Block 1681569
0xd583f7b24973c9b549efc2daa19f87464359e8...

Send data 0xb3f98adc000000000000000000000000000000000000000000000000000000000000001

Fig. 7. Screenshot of a voter's wallet panel after voting.

In Table I, 5 test ballots are presented. We calculated the time spent in each vote for all voters. Voter 1 is the one who creates the test-election, and has the right to vote initially, but other voters do not have this right to vote, therefore we should give them the voting permit. It usually adds one block creation time to their voting time. As can be seen in the table, generally voter 1 is faster than others during casting their votes. All transactions can be run asynchronously, so voter 3 doesn't have to wait others. The cause of the time variation in experiment is the block creation and workload in the Rinkeby network. But, it never exceeds one minute.



: Ballot 3fdf
0x3FdF62Fad02955d01Be422765fB2D75bb409fdCe
0.00 ETHER*

Winning proposal
winning proposal
2

Select function: Vote

To proposal - 8 bits unsigned integer: 1234

Execute from: Account 2 - 18.58 ETHER

Fig. 8. Screenshot of a voter's query on the results.

TABLE I. CONTRACT CREATION AND VOTING TIMES

	Contract Creation	Voter-1 Transaction	Voter-2 Transaction	Voter-3 Transaction
Voting -1	38s	33s	47s	49s
Voting - 2	32s	32s	45s	45s
Voting - 3	42s	39s	56s	56s
Voting - 4	47s	36s	54s	54s
Voting - 5	1m 1s	32s	28s	28s

In this project, our scope is limited for small-scale polls and elections. A larger voting with millions of voters may have different problems to address. The Ethereum network's scalability is still unknown and needs further research, that's why we cannot suggest use of these contracts for nation-wide elections, at least for now. Our contracts are executed in the Ethereum blockchain, so wherever Ethereum wallet can be run (location, platform, device, etc.), our voting application can be used, too. Right now, the Ethereum wallet is supported in Linux, OS X, and Windows platforms. Also, a person who will vote should have a small amount of Ethereum coins, to be able to execute the voting application and to cast a vote. A fundamental problem of blockchain-based e-voting systems is to provide anonymity for voters without compromising the transparency of the general voting process. In detail, all the transactions (money transfers, votes etc.) are essentially written to the blocks of the blockchain as plaintext. So that, a vote from wallet address A to wallet address B can be seen by anyone who has access to the chain. Which is, of course, a big disadvantage. And, it is not possible to use such a system for official/critical elections. Providing this anonymity is also a major challenge in the current state-of-the-art works. Hao et al. in their work, proposed a solution based on the Diffie-Hellman process, which also implies the use of public/private key pairs and random numbers, so that a "two-round" referendum can supposedly be held with some ballot privacy [13]. But, this work does not consider blockchain technology and it may be hard to adopt for different scenarios, though authors of [13] have done it for an Ethereum smart-contracts implementation, in their follow-up work. In [11], authors proposed a two-step voting process using Bitcoin and Zerocoin transfers, respectively. The exchange of the coins in the middle of the transaction process (by a central unit/wallet) provides a rough anonymity. Due to mentioned concerns, our project (initially) is just for small-sized and less critical kinds of elections, the anonymity study is left as a future work. Moreover, there is also a mobile side of this project. Today's fast growing mobile systems make great opportunities and allow us to integrate new technologies to daily life. This whole blockchain voting process can be done and be managed via mobile applications. Today's TV-channels try to implement mobile voting technologies and evolve their systems from SMS-based voting to mobile application-based voting.

IV. Conclusion

By building this proposed smart contract of ours, we have succeeded in moving e-voting to the blockchain platform and we addressed some of the fundamental issues that legacy e-voting systems have, by using the power of the Ethereum network and the blockchain structure. As a result of our trials, the concept of blockchain and the security methodology which it uses, namely immutable hash chains, has become adaptable to polls and elections. This achievement may even pave the way for other blockchain applications that have impact on every aspect of human life. At this point, Ethereum and the smart contracts, which made one of the most revolutionary breakthroughs since the blockchain itself, helped to overturn the limited perception of blockchain as a cryptocurrency (coin), and turned it into a broader solution-base for many Internet-related issues of the modern world, and may enable the global use of blockchain.

E-voting is still a controversial topic within both political and scientific circles. Despite the existence of a few very good examples, most of which are still in use; many more attempts were either failed to provide the security and privacy features of a traditional election or have serious usability and scalability issues [7]. On the contrary, blockchain-based e-voting solutions, including the one we have implemented using the smart contracts and the Ethereum network, address (or may address with relevant modifications) almost all of the security concerns, like privacy of voters, integrity, verification and non-repudiation of votes, and transparency of counting. Yet, there are also some properties that cannot be addressed solely using the blockchain, for example authentication of voters (on the personal level, not on the account level) requires additional mechanisms to be integrated, such as use of biometric factors [12].

The prominence of distributed systems stands out especially when considering the mitigation of the risk that storing the registrations at a central location (office). This can always somehow allow officials to have the opportunity to physically access to the vote records, which could lead to corruptions and cheatings by the authorities. Additionally, in today's connected world, with the concept of the Internet of Things (IoT), expectedly, many non-computer devices will gain access to the Internet. While we are still working on a mobile phone application as a supportive extension to our work to widen the usability; It is important to note that, apart from phones and tablets; air conditioning devices, cars, chairs, clothes, refrigerators, televisions, and many other everyday objects are/will be able to directly reach to the internet. In terms of blockchain, it won't be difficult to build such distributed systems when there is such a large network and a reserve processing power. Moreover, if all these devices work together as a grid to shorten the validation period of transactions in a blockchain, we will be able to do most of our online transactions securely, reliably, and effectively, not only in theory but also in practice.

Acknowledgment

This study is a part of a broader research related to e-voting systems, conducted by the Security Research Group at Dokuz Eylul University Computer Engineering Department, and has been supported by Ericsson Araştırma Geliştirme ve Bilişim Hizmetleri AŞ.

References

[1] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system", [Online]. Available: https://bitcoin.org/bitcoin.pdf .

[2] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger", Ethereum Project Yellow Paper, vol. 151, pp. 1-32, 2014.

[3] C.D. Clack, V.A. Bakshi, and L. Braine, "Smart contract templates: foundations, design landscape and research directions", Mar 2017, arXiv:1608.00771.

[4] E. Maaten, "Towards remote e-voting: Estonian case", Electronic Voting in Europe-Technology, Law, Politics and Society, vol. 47, pp. 83-100, 2004.

[5] U.C. Çabuk, A. Çavdar, and E. Demir, "E-Demokrasi: Yeni Nesil Doğrudan Demokrasi ve Türkiye'deki Uygulanabilirliği", [Online] Available: https://www.researchgate.net/profile/Umut_Cabuk/publication/308796230_E-Democracy_The_Next_Generation_Direct_Democracy_and_Applicability_in_Turkey/links/5818a6d408aee7cdc685b40b/E-Democracy-The-Next-Generation-Direct-Democracy-and-Applicability-in-Turkey.pdf.

[6] "Final report: study on eGovernment and the reduction of administrative burden (SMART 2012/0061)", 2014, [Online]. Available: https://ec.europa.eu/digital-single-market/en/news/final-report-study-egovernment-and-reduction-administrative-burden-smart-20120061

[7] F. Hao and P.Y.A. Ryan, Real-World Electronic Voting: Design, Analysis and Deployment, CRC Press, pp. 143-170, 2017.

[8] N. Braun, S. F. Chancellery, and B. West. "E-Voting: Switzerland's projects and their legal framework–In a European context", Electronic Voting in Europe: Technology, Law, Politics and Society. Gesellschaft für Informatik, Bonn, pp.43-52, 2004.

[9] Estonian National Electoral Committee "E-voting System", 2010. [Online]. Available: https://www.valimised.ee/sites/default/files/uploads/eng/General_Description_E-Voting_2010.pdf.

[10] P. McCorry, S.F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy", International Conference on Financial Cryptography and Data Security. Springer, Cham, pp. 357-375, 2017.

[11] Y. Takabatake, D. Kotani, and Y. Okabe, "An anonymous distributed electronic voting system using Zerocoin", IEICE Technical Report, pp. 127-131, 2016.

[12] U.C. Çabuk, T. Şenocak, E. Demir, and A. Çavdar, "A Proposal on initial remote user enrollment for IVR-based voice authentication systems", Int. J. of Advanced Research in Computer and Communication Engineering, vol 6, pp.118-123, July 2017.

[13] F. Hao, P.Y.A. Ryan and P. Zielinski, "Anonymous voting by two-round public discussion", IET Information Security, vol. 4, pp. 62-67, June 2010.