TDDD17 - Survey on E-Voting Systems and Protocols

Herman Nordin
Department of Computer and Information Science
Linköping University
Linköping, Sweden
herno643@student.liu.se

Techit Lerssongkram
Department of Computer and Information Science
Linköping University
Linköping, Sweden
tecle667@student.liu.se

***Abstract*—The purpose of this survey is to provide basic knowledge about e-voting systems. We discuss DRE and i-voting systems, which are the two most common types of e-voting systems and discuss some examples of those that have been implemented in the past.**

Security is one of the most important topics to discuss when talking about the possibility of e-voting systems, and in this paper, we answer the following three questions. First, what security aspects are important to remember when switching to e-voting? Second, what are the requirements for secure e-voting systems? and finally, what are some useful protocols for e-voting?

We conclude that e-voting systems are large systems with many parts, e.g., voting machines (hardware) provided at polling stations, data (votes) transmission and verification (using protocols) and counting centers. Preventing exploitation in these parts is the most important challenge in providing secure e-voting systems. Some common security requirements that are often addressed when talking about e-voting systems are correctness, privacy, eligibility, robustness, verifiability and usability. Secure e-voting systems aim to meet all these requirements. In recent years, blockchain has been addressed as a possible solution to provide more security for e-voting systems. Blockchain help with providing verification and two of those protocols are the Bitcoin protocol and Ethereum protocol.

***Index Terms*—Bitcoin, Blockchain, E-voting, Ethereum, I-voting**

I. Introduction

In a democracy, an election is a way for the people to choose who represents and leads the country. In traditional voting systems, voters are getting physical ballots where each candidate is individually tabulated and displayed. Voters must be physically present at the polling stations which is time consuming. The election process is expensive, needs much preparation and the tallying phases are bothersome. There are some attempts to solve these difficulties by using modern technologies and as a result electronic voting (e-voting) system is introduced.

E-voting system can provide mobility, increase participation, transparency, efficiency and accuracy [2]. There are several criteria that both the traditional and electronic voting systems must satisfy, these are anonymity, tamper-resistant and accessibility [15]. *Anonymity* aims to make sure that the voter can vote for whoever they want without fear for their safety. *Tamper-resistant* is to make sure that the votes are legitimate and that they are also counted correctly. There should be no possibility of voting twice or having someone else vote in your name, etc. Finally, *accessibility* is to ensure that everyone eligible understands how to vote and can participate in the voting. If this is not done, the legitimacy of the election can be questioned. All these criteria are important for a working voting system, regardless of if it is physical or not. Flaws can lead to incorrect election results. Some more security requirements for the e-voting system to prevent fraud and corruption are presented in section V in this paper.

The purpose of the national election is to choose people who will form a government and lead the country. Consequences of any successful attack are huge, and security is an essential part in e-voting systems. Compared to the traditional voting systems, e-voting systems are more vulnerable to attacks [13]. There is no guarantee that the entire system can be consider as secure from attacks when only one important part such as protocol is protected. Attacks can occur at any level (hardware, software and human) of the system. It is important to assure that every component in the system meets security aspects such as system stability, secrecy, integrity, availability, reliability, safety, and maintainability. For example, a comprehensive approach can be used to guarantee those security aspects [22].

A. Background

Currently most voting systems are physical although there are some examples of countries that have moved from physical voting to e-voting as early as in 2005 [19] [35] and some that have tried to implement such systems but stopped because of security concerns [30].

There are multiple ways of implementing e-voting. It can be done almost as regular voting but instead of filling out a physical ballot at a polling station there could be a computer in which you enter who you vote for as done in India [35], this is called direct-recording electronic voting machine. Or it could be done completely over the internet without having to leave your home as done in Estonia [19] which is called i-voting. These different implementations come with different obstacles to overcome, but they also have many similarities.

B. Problems at issue

The questions this survey seek answers to are the following:

1) What security aspects are important to consider when switching to e-voting?
2) What are requirements for secure e-voting systems?
3) What are useful protocols for e-voting?

II. Method

The method used to write this report was literature study. We divided this into three phases as follows.

A. Gathering Relevant Papers

To find material, various reliable libraries that store and share scientific reports and texts have been used. We mainly used IEEE Explore, Google Scholar and ResearchGate. YouTube videos and Google searches have also been used to inspire and find relevant information for our survey to better understand the topic of e-voting. It requires an account to access IEEE Explore and Linköping University provided such an account for our group.

Searches on these platforms were performed by searching for a keyword along with another word or phrase we were looking for, for example searches may look like *E-voting with blockchain*, *Analyst of e-voting system* or *Security in e-voting*. References in these found reports were also investigated further to find possible hidden information. Relevant paper titles, abstracts and DOIs were gathered in a document.

B. Scrutiny of interesting papers

After relevant papers had been gathered, they were filtered based on their quality, relevance, number of citations and freshness. The approved documents were used as references in this report. In this step we mainly focused on finding useful information about e-voting systems, security requirements and different protocols.

For the e-voting systems part, we tried to find common types of e-voting and some existing systems in the world. For the security requirements, we tried to find common security requirements that should or must be implemented in e-voting systems. Finally for the protocol part, we tried to find and compare different protocols by checking how they ensured the fairness of the voting process.

C. Writing Report

We decided on sections that we wanted to present in this report and wrote them based on the references from the previous section. We were constantly looking for more information for each section as we wrote. Also, the writing process were to be reported to the supervisor and some meetings were held when we needed guidance.

III. Physical voting systems

Physical voting systems require the voter to be physically present with a physical ballot at one of the designated polling stations. When all votes have been cast, they are sorted and counted. The result is then transmitted to the local office of the electoral body. During this tallying process it is checked for faulty votes and that the number of votes matches the number of voters. This process of voting can be very time consuming and there can be many factors that hinder a voter from getting to vote at a polling station. The process of counting all votes can also take a long time and the votes are easily miscounted.

To verify the identity, some form of identification must be brought along with a voter, e.g., passport or driver's license or id [31]. The identity is then checked to make sure that they have not voted before and that they are eligible to vote. To ensure the privacy of the voter, they get to fill out the ballot in a polling booth screened of from everyone else at the polling station [26].

IV. E-voting Systems

The idea of e-voting arose as early as in the 1950s [18] and the topic has continued to be studied since then. The interest in the area shined when David Chaum presented the very first e-voting system in 1981 [6]. E-voting system refers to any electronic platform provided by computers to cast ballots in an election. There are many different implementations of the e-voting system. Some are done completely remotely and some still need the voters to be present at a polling station. Most of the studies focused on two kinds of e-voting systems, the Direct-recording Electronic (DRE) Systems, and the Internet Voting (I-voting) Systems. In this part, we will present information about DRE and i-voting systems and some real systems than have been used in the world.

A. Direct-recording Electronic Voting Systems

The first type of e-voting system uses direct-recording electronic voting machines that allow voters to cast their votes without any physical ballot. Three main electronic processes delivered by DRE are ballot casting, tabulation and transmission. Such machines provide user interfaces that allow voters to vote by interacting with the system using touch screens or physical buttons. This kind of system have been adopted by a few big scale elections. For example, the 2000 US presidential election in Florida [15] and the 2009 India parliamentary election [35].

Because voting machines must physically be placed at voting locations, most problems come from participators within that environment. This kind of system hinges on the correctness, robustness, and security of the voting machines' software. Voters and malicious insiders can take advantage of the system to corrupt the election if security flaws exist. In addition, the machines must be able to send the result or votes to the counting center, which can be done in two possible ways. The results can be transmitted using a flash memory card or by using a network connection such as the internet or an isolated network [15]. Voting results in flash memory cards can be read, overwritten or dropped by insiders when they are physically transported to the counting center. For security reasons, voting machines encrypt both the voting data and the audit data before they are stored in the machines' memory. Strong encryption or a checksum algorithm is needed to prevent the data from hacking.

DRE systems need to receive a ballot definition file as an input. This file has essential information for the system, from appearance of the system to how communication in the system is done. If the system uses some old connection method such as Dial-Up, the ballot definition file will not be encrypted or use checksum when it is sent to the voting machines. If the voting station uses public internet connection, there is a possibility to sniff that connection. Attackers can investigate the IP address of the back-end server, and this opens up for man-in-the-middle attacks where the file can be read and modified by malefactors. This causes voting machines to receive wrong, incomplete or modified files. Another possible attack is Distributed Denial of Service (DDoS) attacks. For a large-scale election software may have to be downloaded through internet and a DDoS attacks can happen both at the software installation process and when voting data is transmitted to the counting center [15]. This can cause the software to not be installed correctly or the data to get lost in transfer.

Hardware-level vulnerabilities are also present in DRE systems. Security analysis of India's electronic voting machines shows possibilities to exploit the systems using vulnerabilities found at the hardware-level. Some common attack methods are using firmware and substitute look-alike hardware. Firmware aims to affect voting software. By attaching the firmware at the voting machine, it can provide a backdoor for the attacker used to interfere with the election. Substituting for a look-alike CPU for example, can lead to incorrectly counted votes which gives the wrong results of the election [35].

B. *Internet Voting Systems*

The second type of e-voting system allows voters to cast votes anywhere through computer devices with an internet connection, such as private personal computers (PC), smart phones, and tablets. Some countries that have adopted i-voting systems are for example Estonia [29] and Norway [1].

In September 2011, Norway used i-voting in municipal and county council elections. The country claimed that about 168,000 could cast their votes online. In these elections, voters authenticated themselves using an e-legitimation service called MinID to enter the voting system. The system's interface provided ballots to the voter. Once they voted, the votes were encrypted and sent to the central voting servers. SMS with 4-digit number was sent to the voter as a verification for their vote. The codes could be used to check if the system received the votes correctly [9].

Estonia's i-voting was introduced in the county council elections in 2005 [19]. The country continued to develop their system and in 2011 the Estonia parliamentary elections were done using the i-voting system [11]. Voting was done by using a client software where voters authenticated themselves using an electronic ID or mobile ID with the respective SIM card. Votes were encrypted by the client application and sent to the receiving server through the internet.

Both Estonian and Norwegian i-voting received criticism about security concerns [3]. Estonian i-voting system was not designed to meet the security requirements in a secure way. In real world elections, i-voting systems must hold certain properties, see security requirement section V-A, in a secure way. One example of the properties is verifiability. Many studies show that this property can be achieved by e.g., implementing end-to-end (E2E) verifiability using cryptographic techniques. Voters can verify that their votes came to the right person without revealing the votes.

The Estonian i-voting system did not use the E2E verifiability property but trusted their voters and the election staff behaved truthfully and that the system worked correctly. A review of Estonia's official election videos from 2013 showed that their system's processes did not complete operational security which made the system open for exploits [29]. The published source code of the Estonian i-voting system was both incomplete and insufficient. These problems raised the question about the system's verifiability and possibility of cyber-attacks.

V. Security in E-voting

Security critics where the main reason Norway suspended its i-voting project in 2013. People were afraid that their votes would be revealed in a cyber-attack [21].

Secure e-voting systems seek to meet all the security requirements to gain trust from the voters. Several issues arise when implementing such systems and one problem is that some of the requirements compete against each other. An example is verifiability and receipt-freeness [33].

A. *Security Requirements*

E-voting systems are big systems containing both hardware and software that provide most of the voting steps in an election. If any part of the system is exploited, it can have a huge effect on the election result [33]. Due to security questions, a number of people do not believe that the e-voting system can be trusted [10]. There is no standard set of requirements or system properties for secure e-voting systems. Different papers consider and prioritize different requirements. In this part we will present most found requirements for e-voting system.

- **Correctness**: This requirement means that all votes must be counted correctly. This requirement is divided into two parts, completeness and soundness. Completeness means that votes cast correctly by authorized persons shall be counted. Soundness means that all votes sent from unauthorized persons shall not be counted [33].
- **Privacy**: Only the voters themselves know who they voted for. The voting system should provide anonymity for voters, which means that the system does not reveal any voters' opinions anywhere [33], [28].
- **Eligibility**: Only people who have the right to vote can access the system. Unauthorized people, such as attackers, who want to influence the election results or people who do not have the right to vote shall not have access to the election process. [33], [28].
- **Robustness**: This means that the system must be able to function properly even if a number of incorrect behaviors

are performed by the voters. For example, the system should not allow voters to cast their vote twice (double-voting) [33].

- **Verifiability**: It must be possible for involved parties to review the systems and verify that all the features are working properly. Some examples are that voters can check that their votes have been counted [28].
- **Usability**: This means that the system must meet technical requirements of efficiency, effectiveness and satisfaction. Efficiency means the help a user needs to use the system correctly. Effectiveness means accuracy with which a user completes the tasks and satisfaction means the user's willingness to use the system [33].

VI. Protocols in E-voting

Protocols in e-voting are created and used to get a structure for how the voting process is completed. The steps in this process are there to make sure that the election is conducted in a legitimate way such that no voter fraud, election fraud or any other corruption exists. It is said that receipt-freeness is necessary for a protocol to be viable in a real election [17]. This is to prevent the threat of coercers buying votes. If a voter does not receive a receipt after voting, there is no way for a third party to confirm that that they voted for whom to get the vote except if they observe them cast their vote.

One way of creating an e-voting system is using a blockchain. There are different blockchain implementations used for e-voting. In this paper we discuss will discuss the Bitcoin [7] and Ethereum [16] protocols.

A. Blockchain

Blockchain was first mentioned by Satoshi Nakamoto in 2008 [20]. Blockchain is a chain of blocks where all transactions are stored and proposed to be used as a peer-to-peer payment system. Each block in the chain contains a hash of the block before itself, the current timestamp, a Merkle tree root hash which contains the hash value of all the transactions in the block and a block version to indicate which set of block validation rules to follow [32] [7]. An example of a blockchain can be seen in figure 1.

Blockchains can work in decentralized environments with the help of its cryptography and distributed consensus. Other features of blockchains are anonymity, persistency and auditability. Although blockchain has all these components there are still some technical limitations. Scalability is the first limitation [32]. Furthermore, the privacy of a transaction is not always kept secret. IP addresses can be tracked even though a transaction is done using only a user's public key and private key [4].

B. Bitcoin Protocol

Bitcoin is one implementation of the blockchain technology. Users can send bitcoin to other users using the addresses corresponding to the sender's and receiver's wallets. To make sure that only valid transactions are recorded the transaction is verified by the miners in the network [32] [7].

The addresses in bitcoin consist of a 160-bit long hash of an elliptic curve digital signature algorithm key pair. The public key which is distributed to other users is derived from the private key and must undergo several steps of encryption (SHA-256, RIPEMD-160 and Base58-encoding) to be used for transfers [7] [4].

To register a transaction, it is broadcasted to the bitcoin network and collected by the blocks in it. The transaction consists of the addresses the transaction is between, the amount of Bitcoin that is transferred and a digital signature to confirm that the sender is the owner of the Bitcoin to be transferred [7].

C. Bitcoin e-voting protocol

One proposed e-voting system using the Bitcoin protocol proposed by J.P. Cruz and Y. Kaji is using it in combination with a Blind Signature Protocol [7]. This process is divided into five steps with three different participating types. The participant types are voters, an administrator and a counter. The administrator's task is to start the voting process, distribute signatures, verify that the voter is eligible to vote, to publish who received signatures and their commitments and distribute prepaid Bitcoin cards. The counters task is to count all the votes and verify that they are valid and announce the result [7].

The process starts of by the administrator publishing the empty voting ballots. The voter, physically present, then selects a vote and creates a commitment with the vote and a random key. From this commitment, a blinded message [34] is then generated using a blinding factor and the public key of the administrator. If the voter is eligible to vote, the administrator generates a signature with their signature scheme and the voters commitment. All voters' identities and their responding blinded messages are published when all voters have received a signature from the administrator. If voters want to make sure that the signature they received from the administrator is valid, they can unblind the signature with the random blinding factor mentioned before. If this signature matches the signature scheme of the administrator it is considered as valid, and the voter can continue [7]. The administrator also distributes the prepaid Bitcoin cards during this process.

When the voter has received their prepaid Bitcoin card, they generate their pair of private key and bitcoin address that they will use when voting. To ensure that the privacy of the voter is kept the voter will have to transfer the Bitcoin from the prepaid card to their generated Bitcoin address from where they will transfer the Bitcoin when voting.

By having the voter publishing the used Bitcoin addresses that are used in the prepaid Bitcoin cards another step of security and reliability is introduced by having only these addresses as valid addresses used in the voting process.

The following steps are done remotely. To publish the address used for voting the voter needs to transfer an arbitrary amount of Bitcoin (it can be any amount that is given to the voter before) to the administrator. The transaction should contain the previous commitment and the unblinded signature

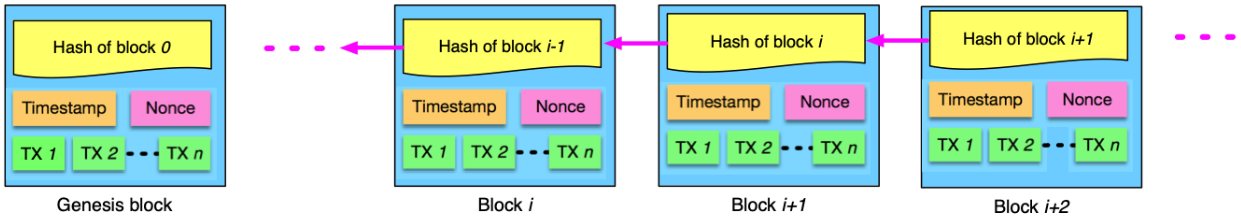

Fig. 1. An example of a blockchain. Source: [32]

received from the administrator. The transaction will be added to the blockchain when the miners have confirmed that it is legitimate. This transaction can be verified by the administrator when published. If the administrator decides that it is a legit transaction, they should then publish the address of the voter, the commitment and the unblinded signature. This data will be used by the counter in later steps. Once everything is published the number of published data should be the same as the number of published identities [7].

Since the votes are published by the administrator the counter can easily collect the votes from the published information. To then count the votes the counter needs to receive the keys that the voters used to commit the votes in the beginning of the process. These keys are received when the voters send their keys, through a Bitcoin transaction, to the counter who then opens all commitments and counts the votes. The result should then be published with a bitcoin address corresponding to each vote.

D. Ethereum Protocol

Ethereum is another blockchain implementation that can be used as a e-voting protocol. Ethereum comes with its own programming language that makes it possible to implement smart contracts. This makes it possible to implement your own rules to satisfy ownership and transaction formats [8] which gives a wide range of use cases [16]. The contracts are written in a programming language called Solidity which is a combination of JavaScript and C++ [16]. These contracts are validated every 15 seconds by other users in the network where it requires a minimum of two other users to verify a contract for it to be valid. Contracts cannot be removed from the blockchain after they have been initialized. A trigger in a smart contract can be triggered by any user with the right address to the smart contract and enough Ethereum to perform the transaction [14].

E. Ethereum E-voting Protocol

One suggested implementation of the Ethereum protocol for e-voting takes advantage of its built-in smart contract feature [14]. This implementation is divided into five components presented in fig 2.

The first component is the web application that helps the administrators create and manage voting events. In this web application the administrator can fill in the questions and possible answers for the voter. This is then sent to the event management server.

The second component is the event management server. This server has the task of creating and deploying smart contracts to the network. The smart contracts should contain the questions and possible answers. This component contains an Ethereum wallet address, and a full node to access the network. These are included to enable the creation of the smart contracts. A list of all existing contract addresses is also present in this component.

The third component is the smart contract itself. This is divided into two different types of contracts, one for registration and one for voting. There exists one registration contract for all voting contracts. This contract registers and authenticates all voters. The task of the voting contract is to handle the questions and votes. This is deployed by the event management server several times with different questions and answers.

The SMS gateway is the fourth component and helps with user authentication. This is done by sending an SMS to the authenticated MSISDNs with a PIN when they have requested authentication through the mobile application.

The last part is the mobile application. This is used by users to register and authenticate themselves and cast their vote. It has the ability to visualize the results of the voting in real-time. The application also has a connection to the Ethereum network to be able to fetch all the data. To be able to do this it has its own Ethereum light client integrated which is considerably smaller in size compared to the full Ethereum node. This Ethereum light client implements a Merkle tree structure to keep track of all transactions which helps in saving space on users' mobile devices.

1) Registration and Voting: To be able to vote the voter first needs to complete the registration process. This is done through mobile authentication [14]. First the user needs to download the application to their mobile device with an existing sim-card and phone number. When the application is launched the application will retrieve the MSISDN from the sim-card. A wallet generated by the application needs to be filled with enough Ethereum for the process to continue. When the voter has transferred the right amount of Ethereum to their wallet a *register*-function will be called within the contract with that voters MSISDN. This function verifies if the MSISDN has been registered before. If it has not been registered before, a HTTP-request is sent to a *true random number generator*-server which will generate a PIN code for

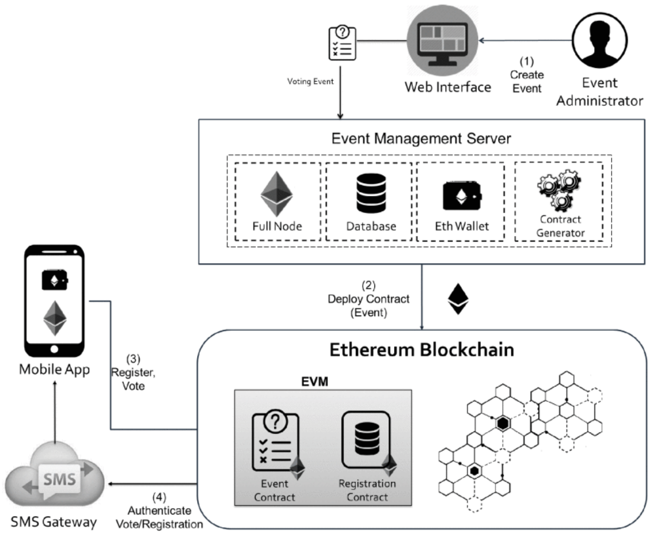

Fig. 2. An example of a blockchain. Source: [14]

that MSISDN. The request is sent using Oraclize (now named Provable) which offers a secure connection between a smart contract and an external web API [24]. After a PIN has been generated it is sent through a SMS gateway back to the voter who enters it into the application to validate itself. If the voter tries to verify someone else's PIN, it will be denied because of a validation that confirms that the sender of the PIN is the same person that received it.

The voting event is created by an organizer through the web client. When questions, possible answers and payment has been entered the contract is deployed to the Ethereum network. The address of the contract is stored in a database in the event management server.

The voter answers the question of who to vote for in the application which uses the function *VoteFor(option)*, this function verifies that the voter is eligible by checking that the user is registered in the registration contract, that they have not voted before and that the voting event has not ended. If the voter passes all those checks the vote is cast and a message is returned saying that the voting was successful. If it was not successful a message saying why is returned [14].

VII. Testing

Testing is a much-needed step when moving to e-voting. Without testing we cannot be sure that the system works, and the general public is also much less likely to trust an untested and unproven system. One way of testing a system in a larger scale is by first introducing it in parallel to the usual voting system. This way the voters can be sure that their votes are counted as they usually are. At the same time, the result can be double checked at the end of the voting to make sure that the result of the e-voting and regular voting is the same.

There are many different existing security models such as *ISSAF*, *OSSTM* and *GNST*.

ISSAF or The Information Systems Security Assessment Framework is a penetration testing method [23]. This method is used for testing networks, systems and application controls. It is divided into three main methods: planning and preparation, assessment and reporting. The first step involves setting up testing environments. This can be what test tools to use, deadlines, requirements and final report structure. The second step involves information gathering, network mapping, vulnerability identification, penetration testing, gaining access and privilege escalation, compromise remote users' sites, keeping access and covering tracks. This is the main part of the testing method and where the penetration testing occurs. The last step is reporting the findings and remove potential modifications that occurred during testing [25].

OSSTM or The Open-Source Security Testing Methodology Manual is another standard for security testing. This method states that all threats must be considered possible even though they are not probable. The possible ways to access the target is divided into three parts, communication security, physical security and spectrum security. These three parts are then further divided into the following subcategories: human, physical, wireless communication, data networks and telecommunication. This method is complex and has many steps before the final report of findings can be written [12].

GNST or The Guideline on Network Security Testing has four steps, first is planning where the most interesting targets of a system are decided by system analysis. Second is discovery, this is where the system is analysed for vulnerabilities by the tester. The third is attack where the found vulnerabilities are analyzed to test whether they were exploitable or not. And the last step is reporting [27].

These methods can be used to analyze the various parts of a e-voting system. The parts that need to be analyzed and satisfied are the following.

- **Voter validation.** The voter should be able to register and be authenticated.
- **Ballot validation.** The voter must be able to choose the right ballot for their voting purpose.
- **Voter privacy.** The voters should not be associated with their respective ballots.
- **Integrity.** There should be no change in ballots during the election.
- **Voting availability.** All eligible voters should be able to vote.
- **Voting reliability.** All votes must count towards the intended recipients result.
- **Election transparency.** It must be possible to audit the result of the election.

It is not until all these parts are satisfied that the system can be trusted.

VIII. Related Works

Earlier studies that have examined e-voting systems present common security issues caused by human error, software and hardware issues, and data transfer communications. These security issues presented in the earlier studies such as [33], [10] come from the same source [15]. The most discussed e-voting systems in these studies is Estonia's voting system. Analysis and discussion about the security requirements, possibility of e-voting systems and various techniques that can be used to address the security issue are often the main topic in these studies.

IX. Discussion

The previous implementations of e-voting systems such as Norwegian and Estonian proved to be insufficiently secure due to lack of security. Common security issues were published as early as 2004 [15] and the systems implemented after that time tried to solve these problems. More specifically, the systems implemented between 2007 and 2013 tried to solve these problems with various technologies that existed at the time, such as encryption and cryptography. These techniques increased the difficulty of attacking but there were still vulnerabilities.

A. Bitcoin E-voting Protocol

In the proposed Bitcoin protocol for e-voting the voter needs to understand how Bitcoin works and how to create wallets and transfer coins. This gives them security over what sensitive information they share with other parties but at the same time require them to do and know more about the process themselves. It can be made easier by not having the voter create their wallet, transfer the coins etc. manually but instead doing it through a well-designed interface as in the proposed system using the Ethereum protocol.

Correctness is provided if the protocol is followed correctly and if all parties included are honest. It also satisfies the completeness property if everything is done correctly, and all votes should therefore be counted. The soundness property is also satisfied if the voter does not share their private keys with other ineligible voters.

If a voter does not share who they voted for their privacy is kept. The voter only confirms its identity when registering to vote. The administrator in has no knowledge of who the voter votes for since a blind signature is used on the vote. The IP-address from where the Bitcoin is sent can be traced in some cases, but if the voter takes the proper steps to hiding it, e.g., using a wallet only for voting and using the Tor browser it can be protected [7].

Eligibility by not having any unauthenticated voters vote is ensured mainly by the administrator whose task it is to authenticate them. If the signature use by the administrator is impossible to reproduce by a voter themselves, they cannot fake authentication and therefore they need to be authenticated by the administrator instead.

Robustness is also satisfied if the protocol is followed properly. There are a few different scenarios where the voting could go wrong that are prevented using this protocol. The first stage is the voting. If a user tries to send invalid votes to the administrator, it will be impossible to tell at that stage and the voting will continue. Instead, when the counting of the votes is taking place, the counter will notice that the vote is invalid, and it will not be counted. If the voter sends a bad key to the counter in the counting stage the problem is not because of the system but because of the voter and the vote will not be counted because of it. It is not possible for the administrator to publish fake votes since the list of identification and votes will not match the list of addresses, votes, and signatures. They cannot publish votes for a person that did not vote either unless they have access to their private keys, Bitcoin address etc. and if they have that access it is not because of the system. The counter cannot exclude a vote in their counting since it is publicly available in the blockchain, neither can the administrator exclude adding Bitcoin addresses to the blockchain.

Since everything is publicly available on the blockchain it is easily verifiable by the voter that their vote should be included in the counting.

That the usability criterion is satisfied is difficult to tell from a technical analysis of a system. For it to be correct a rigorous usability test needs to be conducted since the usability and willingness of the voter to use the system is one of the most important parts [33]. To contribute to the usability aspect, it is necessary to teach users about blockchain, Bitcoin and the voting process beforehand.

B. Ethereum E-voting Protocol

The proposed E-voting protocol using Ethereum is designed with usability in mind, with its mobile application it is easier to vote for many. On the other hand, not everyone has a private smartphone, some share one with a partner and some does not have one at all [5]. Since the smart contract of Ethereum handles all verification, registration, voting and counting the aspect of corrupt employees is removed.

Privacy is accounted for since who a voter has voted for is never saved, only if a voter is eligible or not and if they have voted or not. This way the privacy is kept if the voters and votes are separated.

Eligibility in the current system is only checked based on the MSISDN which can be a problem if multiple persons use the same phone. Therefor to satisfy the eligibility constraint further authentication is needed. This is mainly when used it national elections and elections where fraud is a common thing.

Since the system is completely done on the Ethereum smart contracts it is not possible to miscount the votes. It is not possible to insert illegitimate votes either since the smart contract always check if the voter is authenticated and if they have voted before. The robustness criterion is therefore satisfied.

Since everything is done on the Ethereum network it is easily verifiable for a voter that the vote was counted. This satisfies the verifiability constraint.

Usability still needs a lot of testing to be satisfied. As with the Bitcoin e-voting protocol, it is difficult to know how a user who will only use the voting system a few times every four years will react to its design etc.

For this system to be used in bigger elections it needs more verification of the voters. The addition of other verification than just the MSISDN of a SIM-card is needed to verify the identity of a voter in a national election for example as the MSISDN is not enough to tie a person to an identity. This makes this proposed implementation of the e-voting more suitable for smaller elections where less identification is needed.

X. Conclusion

In this study, we introduced two most common types of e-voting systems (Direct Recording Electronic (DRE) Systems and Internet Voting (i-voting) Systems) that have recently been implemented on a large scale. We presented the e-voting system in India as an example of the DRE system and Norwegian and Estonia as examples of the i-voting system. All systems shown in this study contain vulnerabilities that allow exploitation and some of these systems have been discontinued due to security criticisms. Moving to e-voting system requires all the parts (hardware, software, data transfer and contacting center) in the system to by protect from any exploitation. This study addresses common security requirements for e-voting that should be met and some difficulties in implementing such a system.

We study various protocols presented in recent years based on Blockchain technology. Both Bitcoin and Ethereum show the potential to improve the security of e-voting systems. The Bitcoin e-voting protocol discussed in this report ensures all security requirements are met if the protocols are followed properly. The proposed Ethereum e-voting protocol satisfies the security requirements. Although the lacking part of the system for it to be viable in bigger elections, such as a national election, is the verification. The current verification relies on the voters MSISDN which is not reliable to tie a voter to an identity. To improve this the verification needs to rely on a more secure way of identification through the internet such as e-identification.

The main problem implementing these protocols and moving from the current voting system is teaching people how to vote and have them trust the system. One way of increasing the trust for such system would be teaching voters more about blockchain and the underlying technology of the systems as well as creating a well-designed and easy to understand interface. Rigorous testing is also much needed to make sure that the system meets all requirements. There are many suitable testing methods that gives the tester a structure of how and what to test, some of these are ISSAF, OSSTM and GNST.

Acknowledgment

We would like to thank Linköping University for providing access to the IEEE Xplore library. We would also like to thank our supervisor Dr. Ioannis (Yannis) Avgouleas at Linköping University for the materials, feedbacks and suggestions.

References

[1] Gjøsteen K. (2012). The norwegian internet voting protocol. in: Kiayias a., lipmaa h. (eds). In Robert Krimmer and Rüdiger Grimm, editors, *International Conference on E-Voting and Identity*, pages 1–18, Bonn, 2008. Gesellschaft für Informatik e. V.

[2] R. Anane, R. Freeland, and G. Theodoropoulos. e-voting requirements and implementation. In *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007)*, pages 382–392, 2007. doi:10.1109/CEC-EEE.2007.42.

[3] Ahmed Ben Ayed. A conceptual secure blockchain-based electronic voting system, 05 2017. doi:10.5121/ijnsa.2017.9301.

[4] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. Deanonymisation of clients in bitcoin p2p network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS ’14, page 15–29, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2660267.2660379.

[5] PEW RESEARCH CENTER. Demographics of mobile device ownership and adoption in the united states, 2021-04-07. Accessed: 2021-05-03. URL: https://www.pewresearch.org/internet/fact-sheet/mobile/.

[6] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–90, February 1981. doi:10.1145/358549.358563.

[7] Jason Paul Cruz and Yuichi Kaji. E-voting system based on the bitcoin protocol and blind signatures. In *IPSJ Transactions on Mathematical Modeling and Its Applications*, 03 2017.

[8] Ethereum.org. Ethereum whitepaper, a next-generation smart contract and decentralized application platform, 2021-02-09. Accessed: 2021-04-26. URL: https://ethereum.org/en/whitepaper/.

[9] Ida Sofie Gebhardt Stenerud and Christian Bull. When reality comes knocking norwegian experiences with verifiable electronic voting. In Manuel J. Kripp, Melanie Volkamer, and Rüdiger Grimm, editors, *5th International Conference on Electronic Voting 2012 (EVOTE2012)*, pages 21–33, Bonn, 2012. Gesellschaft für Informatik e.V.

[10] J.P. Gibson, R. Krimmer, and V. et al. Teague. A review of e-voting: the past, present and future. *Annals of Telecommunications*, 2016. URL: https://doi.org/10.1007/s12243-016-0525-8.

[11] Sven Heiberg, Peeter Laud, and Jan Willemson. The application of i-voting for estonian parliamentary elections of 2011, 09 2011. doi:10.1007/978-3-642-32747-6_13.

[12] Pete Herzog. Open-source security testing methodology manual. *Institute for Security and Open Methodologies (ISECOM)*, 2003.

[13] H. Hussien and H. Aboelnaga. Design of a secured e-voting system. In *2013 International Conference on Computer Applications Technology (ICCAT)*, pages 1–5, 2013. doi:10.1109/ICCAT.2013.6521985.

[14] David Khoury, Elie F. Kfoury, Ali Kassem, and Hamza Harb. Decentralized voting platform based on ethereum blockchain. In *2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, pages 1–6, 2018. doi:10.1109/IMCET.2018.8603050.

[15] T. Kohno, A. Stubblefield, A. D. Rubin, and D. S. Wallach. Analysis of an electronic voting system. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, pages 27–40, 2004. doi:10.1109/SECPRI.2004.1301313.

[16] Ali Koç, Emre Yavuz, Umut Çabuk, and Gökhan Dalkılıç. Towards secure e-voting using ethereum blockchain. In *International Symposium on Digital Forensic and Security (ISDFS)*, 03 2018. doi:10.1109/ISDFS.2018.8355340.

[17] Byoungcheon Lee, Colin Boyd, Ed Dawson, Kwangjo Kim, Jeongmo Yang, and Seungjae Yoo. Providing receipt-freeness in mixnet-based voting protocols. In Jong-In Lim and Dong-Hoon Lee, editors, *Information Security and Cryptology - ICISC 2003*, pages 245–258, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[18] Leontine Loeber. E-voting in the netherlands; from general acceptance to general doubt in two years. In Robert Krimmer and Rüdiger Grimm, editors, *Electronic Voting 2008 (EVOTE08). 3rd International Conference on Electronic Voting 2008, Co-organized by Council of Europe, Gesellschaft für Informatik and EVoting.CC*, pages 21–30, Bonn, 2008. Gesellschaft für Informatik e. V.

[19] Epp Maaten. Towards remote e-voting: Estonian case. In Alexander Prosser and Robert Krimmer, editors, *Electronic voting in Europe - Technology, law, politics and society, workshop of the ESF TED programme together with GI and OCG*, pages 83–90, Bonn, 2004. Gesellschaft für Informatik e.V.

[20] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Satoshi Nakamoto Institute, 2008.

[21] Ministry of Local Government and Modernisation. Internet voting pilot to be discontinued., 2014-06-15. URL: https://www.regjeringen.no/en/aktuelt/Internet-voting-pilot-to-be-discontinued/id764300/.

[22] B. Ondrisek. E-voting system security optimization. In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–8, 2009. doi:10.1109/HICSS.2009.173.

[23] owasp.org. Penetration testing methodologies, 2021-04-29. URL: https://owasp.org/www-project-web-security-testing-guide/v41/3-The_OWASP_Testing_Framework/1-Penetration_Testing_Methodologies.

[24] provable.xyz. The provable blockchain oracle for modern dapps, 2021-04-29. URL: http://docs.provable.xyz/#background.

[25] Marco Ramilli and Marco Prandini. An integrated application of security testing methodologies to e-voting systems. In Efthimios Tambouris, Ann Macintosh, and Olivier Glassey, editors, *Electronic Participation*, pages 225–236, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[26] Riksdagen. Så här röstar man, Accessed 2021-04-19. URL: https://www.riksdagen.se/sv-ll/sprak/lattlast/val-till-riksdagen/sa-har-rostar-man/.

[27] Karen Scarfone, Murugiah Souppaya, Amanda Cody, and Angela Orebaugh. Technical guide to information security testing and assessment. *NIST Special Publication*, 800(115):2–25, 2008.

[28] Freya Sheer Hardwick, Apostolos Gioulis, Raja Naeem Akram, and Konstantinos Markantonakis. E-voting with blockchain: An e-voting protocol with decentralisation and voter privacy. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1561–1567, 2018. doi:10.1109/Cybermatics_2018.2018.00262.

[29] Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine, and J. Alex Halderman. Security analysis of the estonian internet voting system. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 703–715, New York, NY, USA, 2014. Association for Computing Machinery. doi:10.1145/2660267.2660315.

[30] Prof. Dr. Alexander H. Trechsel. Recent internet voting experiences in europe: A short assessment of the situation. In *POTENTIAL AND CHALLENGES OF E-VOTING IN THE EUROPEAN UNION*, pages 27–29, 2016. URL: https://www.europarl.europa.eu/RegData/etudes/STUD/2016/556948/IPOL_STU%282016%29556948_EN.pdf.

[31] Valmyndigheten. Rösta i din vallokal på valdagen, 2021-02-24. Accessed 2021-04-19. URL: https://www.val.se/att-rosta/var-rostar-jag/rosta-pa-valdagen-i-din-vallokal.html.

[32] Huaimin Wang, Zibin Zheng, Shaoan Xie, Hong-Ning Dai, and Xiangping Chen. Blockchain challenges and opportunities: a survey. *International Journal of Web and Grid Services*, 14:352 – 375, 10 2018. doi:10.1504/IJWGS.2018.10016848.

[33] King-Hang Wang, S. Mondal, K. Chan, and Xuecai Xie. A review of contemporary e-voting: Requirements, technology, systems and usability. *Data Science and Pattern Recognition, Ubiquitous International*, 1(1), 2017.

[34] Wikipedia. Blind signature, 2021-01-18. Accessed: 2021-05-03. URL: https://en.wikipedia.org/wiki/Blind_signature.

[35] Scott Wolchok, Eric Wustrow, J. Alex Halderman, Hari K. Prasad, Arun Kankipati, Sai Krishna Sakhamuri, Vasavya Yagati, and Rop Gonggrijp. Security analysis of india's electronic voting machines. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS '10, page 1–14, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1866307.1866309.