T.C.

AKSARAY UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF ELECTRICAL-ELECTRONIC AND COMPUTER ENGINEERING

DIGITAL VOTING SYSTEM BASED ON BLOCKCHAIN TECHNOLOGY

MASTER OF SCIENCE THESIS

Mahmoud Bakir Ahmed AL-RAWY

SUPERVISOR

Prof. Dr. Atilla ELÇİ

AKSARAY, 2018

T.C.

AKSARAY UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF ELECTRICAL-ELECTRONIC AND COMPUTER ENGINEERING

DIGITAL VOTING SYSTEM BASED ON BLOCKCHAIN TECHNOLOGY

MASTER OF SCIENCE THESIS

Mahmoud Bakir Ahmed AL-RAWY

SUPERVISOR

Prof. Dr. Atilla ELÇİ

AKSARAY, 2018

Aksaray Üniversitesi Fen Bilimleri Enstitüsü'nün 162335803 numaralı Yüksek Lisans öğrencisi **MAHMOUD BAKIR AHMED ALRAWY** tarafından hazırlanan "DIGITAL VOTING SYSTEM BASED ON BLOCKCHAIN TECHNOLOGY" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ / OY ÇOKLUĞU ile **Elektrik-Elektronik ve Bilgisayar Mühendisliği** Anabilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr., Atilla ELÇİ

Üniversite Adı: Aksaray Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/~~onaylamıyorum~~.....

Üye: Prof. Dr., Ali YAZICI

Üniversite Adı : Atılım Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/~~onaylamıyorum~~.....

Üye: Dr. Öğr. Üyesi, Filiz SARI

Üniversite Adı : Aksaray Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum/~~onaylamıyorum~~.....

Tez Savunma Tarihi: ...28../12/2018

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

Mehmet Ali HINIS.

Doç. Dr.

Fen Bilimleri Enstitüsü Müdürü

DECLARATION

I hereby declare that this thesis, presented for the Master of Sciences degree, has been composed entirely by myself, been just the result of my work, and not been submitted for any other professional qualification or previous degree. The study was entirely conducted under the supervision of Prof. Dr. Atilla ELÇİ.

Signature

Mahmoud Bakir Ahmed AL-RAWY

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to my thesis advisor Prof. Dr. Atilla Elçi of the School of Natural and Applied Sciences, Electrical-Electronic and Computer Engineering at Aksaray University. Prof. Atilla Elçi was always supportive and helpful without prompting, offering essential information to make my thesis unique and novel. He is an inspiring professor who not only pointed me in the direction of the topic of this thesis, he also shared valuable ideas with me at various stages of the research without sparing useful remarks and suggestions, helping me tremendously in the process. I also would like to thank him for being my inspiration not to give up, and his words of encouragement helped me accomplish what I have done so far. I have never met another professor who loves teaching as much as he does.

Furthermore, thanks to my friends at Aksaray University. Dear friends, you have always believed, helped and encouraged me during my study period.

Finally, I must express my profound gratitude to my parents (BAKIR ALRAWY and ZAINAB TARIQ) and my wife ROZETA TUSHA for providing me with unfailing support and continuous encouragement throughout the years.

Thank you all, without you, I would not be able to achieve this accomplishment.

MAHMOUD BAKIR AHMED ALRAWY

AKSARAY, 2018

TABLE OF CONTENTS

Page

YÜKSEK LİSANS TEZİ

BLOKZİNCİR TEKNOLOJİSİ TEMELLİ DİJİTAL OYLAMA SİSTEMİ

Mahmoud Bakir Ahmed Al-RAWY

Aksaray Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Atilla ELÇİ

ÖZET

Son yıllarda, Blokzincir teknolojisi, büyük ölçüde, hayatın her yönünü etkilemiştir, ancak bugüne kadar, seçimler Blokzincir'in desteğini almadı ve kağıt temelli seçimler uygulanmaktadır. Blokzincir ve gelişmiş kriptografi yöntemleri gibi modern teknolojiyi kullanarak seçim senaryosunu yürütme zamanı gelmiştir. Estonya ve New South Wales (Avustralya'nın eyaleti), e-Oylama sistemlerini kullanıyordu; ancak analiz için üretilen sistemlere örnek bir yazılımın, kötü amaçlı yazılım, ağ saldırıları ve sunucu saldırıları gibi birçok saldırıya karşı zayıflıklarının olduğu keşfedildi. Blokzincir teknolojisinin, korsanlığa karşı sonsuz bir değişmezlik ve direnç sergilediği gerçeği, her bir veri parçasını değiştirilemez biçimde saklayarak seçim sonuçlarının güvenliğini sağlamak için kullanılmasının, değiştirilemeyen tarih ile kayıt veya işlemin mümkün olduğunu göstermektedir.

Bu tezde, vatandaşların seçim sürecini kolaylaştıran blokzincir bazlı güvenli bir çevrimiçi oylama sistemi öneriyor ve deneme için geliştiriyoruz. Çalışmamızın özü, geleneksel veri tabanından vazgeçerek ve bunun yerine kör imza ile iki Blokzincir kullanarak, oylayan/oy gizliliğini sağlıyor ve sonuçları değişikliklerden koruyoruz. Ayrıca Blokzincir, dağıtılmış ağı kullanarak ağ üzerindeki yükü azaltmaktadır. Son olarak, e-kimliğe bürünme ve oy satış sorunlarına çözümler önerilmektedir. Sayısal oylama sistemi tasarımı arkasındaki teknoloji, e-kimlik oluşturma, kimlik doğrulama, oylama ve oy kullanma gibi süreçler açısından açıklanmıştır.

Anahtar Kelimeler: Blokzincir, Oy gizliliği, e-Oylama, Açık anahtarlı şifreleme, Kör imza.

Aralık, 2018; 70 sayfa

M.Sc. THESIS

DIGITAL VOTING SYSTEM BASED ON BLOCKCHAIN TECHNOLOGY

Mahmoud Bakir Ahmed Al-RAWY

Aksaray University
Graduate School of Natural and Applied Sciences
Department of Electrical-Electronic and Computer Engineering

Supervisor: Prof. Dr. Atilla ELÇİ

ABSTRACT

In recent years, blockchain technology has, to a large extent, affected all aspects of life, however, until now, elections have not received their share of blockchain support and paper-based elections have been implemented. It is time to upgrade the election scenario using modern technology such as blockchain and advanced cryptography methods. Both Estonia and the New South Wales state of Australia have been using e-Voting systems, but an example software of their systems produced for analysis was discovered to have weaknesses against many kinds of attacks, such as malware, network attacks, and servers' attacks. The fact that the Blockchain technology has demonstrated infinite immutability and resistance against hacking illustrates that its use to secure election results from fraud by saving every single piece of data, record or transaction with unchangeable history is possible.

In this thesis, we propose and test implement a secure online voting system based on blockchain that facilitates citizens' election process. The essence of our work consists in abandoning the traditional database and compensating it with two private-blockchains instead of one with the blind signature to ensure voter/vote privacy and the results can be safeguarded from manipulation. Also, using Blockchain's distributed network reduces the load on the net. Finally, solutions to problems of impersonation and vote-selling are suggested. The technology behind the digital voting system design is explained in terms of the processes involved, such as ID creation, authentication, voting, and vote tallying.

Keywords: Blockchain, e-Voting, ballot privacy, Public-private key algorithm anonymization, Blind signature.

December, 2018; 70 pages

LIST OF FIGURES

Page

LIST OF TABLES

Page

LIST OF ABBREVIATIONS

BL-i	Identifications Blockchain
BL-v	Ballots Blockchain
EAR	Electoral Authority Representative(s)
NEN	National Electronic Number
NIDED	National Identification Card Expiration Date
NIDN	National Identification Number
Out	Recipient Address
Pout	Previous Out
TxID	Transaction ID
e-Voting	Electronic Voting
eID	Electronic Identification
i-Voting	Internet Voting

1. INTRODUCTION

Elections are serious and decisive events in every country and their fairness, simply put, determines the fate of those countries and their citizens. For hundreds of years, traditional elections, based on the principle of accepting only one ballot paper at a specific polling station, have been practiced. The cost budget of each ballot paper, trustee, preparation of a polling station, time spent, difficulties faced by disabled people and other high costs not to mention the repeatedly encountered problems such as fraud, manipulation of results and influencing voters are all factors to consider against traditional, paper-based elections. History is littered with examples of elections being manipulated to influence their outcome; scammers and rulers have developed means of manipulating votes to achieve personal agendas, which is considered a violation of the core principle of democracy.

With the advent of the ever-developed Internet, many researchers have devoted themselves to find the easiest, most economical and secure way to obtain a fair e-election by using an e-Voting system aimed to overcome all problems faced by the traditional elections. e-Voting system or Internet voting may be defined as an election system constructed on cryptographic techniques allowing voters to cast their ballots for their favourite candidates and transmitting their votes over the Internet from anywhere in the world while the voting and tally processes run entirely anonymously. However, due to the fact that e-Voting systems run over the internet, significant challenges such as voters' eligibility, ballot privacy, the completeness of the process, immutability of the results and fairness have been obstacles for decades.

All around the globe, countries have started executing their elections over Internet, which facilitated citizens' electoral process. So far, more than 14 countries have used online remote voting in national elections (Pran and Merloe, 2007). Yet, only two countries in the world (Estonia and Switzerland) have succeeded in holding national elections through the Internet. Some of the rest of the twelve countries just tried to use e-voting or used it for a while then suspended it. (Lars and Hole, 2012). In 2005, Estonia became the first country in the world to introduce an online election system

and implement an internet election. In their most recent elections, more than 30% of voters cast their ballots online (URL-4; Kizhakkedathil, 2016); Also, more than 280,000 votes were submitted through online voting in New South Wales (Brightwell et al., 2015), and 70,090 Norwegian votes were also submitted online in 2013 (Segaard et al., 2014).

Security analysis of these systems (Halderman and Teague, 2015; Springall et al., 2014) has shown that security vulnerabilities originating in voters' client devices, servers and in voter authentication process existed. The research for source e-voting practice was intensified as introduced below.

With the emergence of Bitcoin (Nakamoto, 2009), many researchers on Blockchain technology and smart contracts suggested that it is a suitable basis for e-Voting. Furthermore, there is an ability to enhance e-Voting systems efficiency and security with Blockchain technology's great properties such as the immutability, elimination of intermediaries, high availability and data integrity in cloud computing environments (Elçi, 2018; Gaetani et al., 2017), whereby Blockchain could have the prospect to render e-Voting more acceptable and reliable in societies (Glaser, 2017).

Blockchain was designed as a digitized, and decentralized public ledger for digital currency transactions originally in connection to Bitcoin, the first crypto-currency. Fundamentally, Blockchain technology is used to verify transactions within the crypto-currencies, although it has the potential to insert any token, as long as it is digitized, into a "chain" (Zheng et al., 2017). In recent years, Blockchain technology has attracted much attention due to its extreme resistance against hacking due to the fact that it is not an isolated technology but rather a mixture of cryptographies, mathematics, and algorithms that have been perfectly combined to complement one another using the distribution principle (distributed replication, decentralization). Therefore, there are many copies of the identical data spread over many nodes, but none of these copies are the "golden copy" (Nair and Sebastian, 2017). This technique gives us a fast, secure and transparent peer-to-peer transfer of any kind of transactions (Nakamoto, 2009) .

Just recently, in 2018, the presidential elections in Sierra Leone employed Blockchain as a base for the e-Voting system for the first time in voting history.

The most important advantages of using Blockchain in an e-Voting system structure are:

1) Anonymity.
2) Accuracy and security, particularly against DDoS Attacks (Distributed Denial of Service Attacks).
3) Strong integrity (immutability).

1.1. Major Election Concerns

We can highlight some of the main vulnerabilities e-Voting systems have usually faced as follows:

1) Ballot Privacy. Voters' choices must be anonymous (secret ballot). Non-observance of the secrecy of ballots leads to attempts to influence the voter either by intimidation or potential vote buying (Jonker et al., 2013). Ballot security is the biggest concern that threatens the integrity and fairness of the elections. There have been many breaches of the privacy of ballots in both digital and traditional elections. For instance, the secret ballot arrangements of the United Kingdom election have been criticized due to its possibilities to link a voter identity to his ballot paper after casting a vote. When every voter has an identification number and every ballot paper has its own number, it is easy to link the voter's number to their noted-down number on the counterfoil of the ballot paper. Stated more clearly, if anyone can gain access to the counterfoils or if the polling station staff plot with election trustees, the possibility emerges to determine how individual voters voted.
2) Eligibility. Impersonation and multiple vote casting are serious issues with traditional and digital voting systems. Deceptive voters used to register themselves multiple times or manipulate their ability to vote. In Australia, 217 ineligible votes were cast in the 1996 elections and in the 2010 elections, a family cast more than 150 votes by impersonating others. Also, the security analysis of the Estonian internet voting system demonstrated that attackers could plant

malware in the voters' client computers and read their National-ID card's PIN, then impersonate them to cast an unqualified vote to their desire (Nair and Sebastian, 2017; Springall et al., 2014). The lack of biometric devices may allow for identity theft. A biometric device would not be available in every home on election day as supplying such equipment for each house is very expensive and impractical. Therefore, a solution must address this issue.

3) Integrity. Ballot and result integrity must be guaranteed in voting systems; it must keep up the integrity of the voters' data and their ballots. Thus every single transaction must be recorded, verifiable and obtainable without being open to modification and manipulation by way of unauthorized access (Gibson, 2013).

1.2. Blockchain

Blockchain technology is a new way to keep transactions safely. The big difference between Blockchain technology and existing traditional methods (databases) is that Blockchain is not centrally maintained by any one authority, as shown in Figure 1.1. It works by the principle of decentralized distribution (Crosby et al., 2015).

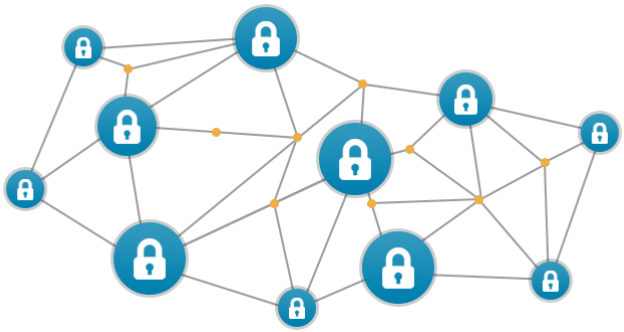

Figure 1.1. A Blockchain network.

The features rendering Blockchain so important are:

1) Decentralized: Blockchain uses the distribution principle (decentralization). Therefore, there is no golden copy, each group of nodes contains a copy of the same data (Glaser, 2017).
2) Open Source: The fact that Blockchain technology is open source makes this technology available for everyone in any domain to use.

3) Immutable: Blockchain elements (blocks) are unchangeable unless more than 51% of the nodes are attacked (Eyal and Sirer, 2018).
4) Anonymity: Anonymous transfer transactions between nodes. The only thing known is recipients and senders' addresses (Moser, 2017).

Blockchain technology is introduced in more detail in section (2.5) of Methodology chapter. Even in-depth treatise may be found in (Zibin et al., 2017).

1.3. Related Work

Several researchers have devoted themselves to designing a fair voting system with or without using Blockchain. David Chaum introduced the first electronic voting system in 1981, in which public key cryptography was used to maintain the anonymity of the voters. Furthermore, he used the Blind Signature (Chaum, 1981) to guarantee absolute disconnectivity between voters and their votes. Since then, many academicians have shown interest in e-Voting, especially with the evolution of cryptography (ElGamal, 1985; Evans and Paul, 2004; Ibrahim et al., 2003; Jan et al. 2001).

1.3.1. Centralized e-voting

DeMillo et al., (1982) proposed an e-Voting protocol which requires voters to participate and encrypt their ballot, before finally casting it.

Cohen and Fischer, (1985) proposed a protocol leading to a secure-ballot election where voters can cast their suitably encrypted votes electronically. After that, the government releases a tally result along with a proof of its correctness, which can be verified by all.

Fujioka et al., (1992) proposed a practical, secret e-Voting protocol suitable for large-scale elections, which can guarantee voters' privacy and the fairness of voting. It used the Blind Signature to blind ballots before sending it to the administrator.

Juang and C.-L., (1997) proposed a Blind Signature based voting scheme (Foo). The system requires that everyone must attend the election.

Ohkubo et al., (1999) promoted the Blind Signature based voting scheme by using a threshold encryption protocol and the Mix-Net communication channel which can protect the privacy of the voter. So voters did not need to participate in the tallying event, they can leave after voting.

Hirt and Sako, (2000) used homomorphic ElGamal encryption technique to design a private channel protocol with receipt-freeness, but his protocol was not suitable for large scale elections.

Baudron et al., (2001) proposed a new protocol which satisfies the receipt-freeness property by using Paillier cryptosystem and the zero-knowledge proof.

Juels, Catalano, and Jakobsson, (2010) proposed a scheme and introduced the new direction of e-Voting named coercion-resistance.

Schläpfer et al., (2011) promoted their project by adding a random integer and encrypting it. So the tally authority can discover fraudulent ballots through decryption of the encrypted random integer. Thus, the coercion-resistance can be further satisfied.

1.3.2. Blockchain-based e-voting

By the development of the decentralized ledger technology, many researchers proposed a way to cast votes and tally results on Blockchain. In 2015, Czepluch claimed that Blockchain could be used for the e-Voting after discussing the Blockchain application domains (Czepluch et al., 2015).

Also, Zhao and Chan suggest that a method to cast a vote using Bitcoin and ZK-SNARKs guarantees the privacy, verifiability, and irrevocability properties (Zhao and Chan, 2015).

Jason and Yuichi, (2017) proposed a scheme which uses the Blind Signature and Bitcoin cards. However, in some situations, there were privacy vulnerabilities. For example, if the administrator knows the voter's Bitcoin address, and so by linking the address and the ballot in the Blockchain he can see the voter identity.

Liu and Wang, (2017) exploited Blockchain properties such as transparency, decentralization, irreversibility and nonrepudiation to propose a decentralized e-Voting protocol without using a trusted third party. Furthermore, they provided many possible extensions and improvements that meet the requirements of some specific voting scenarios.

Bartolucci et al., (2018) introduced secret share-based voting on the Blockchain (SHARVOT) protocol. The main challenges to the protocol are anonymity, ballot privacy, and ballot irrevocability. Also transparency throughout the vote tallying process. The system provides for the exploitation of the immutability and transparency of the distributed ledgers (DLT) to implement a secure voting system for a fair election.

Wu (2017) proposed a voting system design based on Blockchain with the Ring Signature algorithm. He developed the system using PHP and JavaScript programming languages to prove the protocol feasibility. Eventually, this satisfied the requirements of the privacy of ballot, eligibility, individual verifiability, and coercion-resistance.

Ayed (2017) proposed a secure Blockchain-based e-Voting system concept. He took advantage of the open source Blockchain technology novel concept and leveraged it to introduce a secure Blockchain-based e-Voting system. A safe, reliable, and anonymous system that could be used in elections. Also, it could increase voter participation and increases citizens' trust in their governments.

CasaDo-VaRa and CoRCHaDo, (2018) proposed a new model of Blockchain for democratic voting how Blockchain could cast of voter fraud. This approach was designed to reduce or even prevent voting system flaws.

Furthermore, the distributed ledger (Blockchain) has been used to broadcast digital/smart contract voting to a polling station. After that, the polling station will send a smart contract to individual voters and register the vote on a side-chain. At the end of the voting process, the entire sidechain is committed to the main voting Blockchain. Using smart contract to store votes is an excellent way to prevent malicious or corrupt administrators from manipulating results.

Hardwick et al., (2018) proposed a potential new e-Voting protocol. The system uses the disruptive innovation which is the Blockchain as a transparent ballot box. Furthermore, the proposed scheme uses the decentralization and immutability properties of Blockchain to support a voting process, as well as makes it open, fair, and independently auditable. The system was designed to abide by fundamental e-Voting properties, also provides decentralization and allows for voters to change/update their vote (within the permissible voting period).

Hjalmarsson et al., (2018) elicited the requirements of building electronic voting systems and identifies the legal and technological limitations of using Blockchain as a service for realizing a distributed electronic voting systems. Then after evaluating some of the popular frameworks that offer Blockchain, they proposed a novel electronic voting system based on Blockchain addressing all limitations that they discovered. Also, they improved the security and decreased the cost of hosting a nationwide election.

1.3.3. Real-world e-voting applications

Let us highlight the factual practices in some of the expertly designed e-Voting systems:

Estonian i-Voting System: In 2005, Estonia became the first country in the world allowing its citizens to cast their votes online (i-Voting) apart from the traditional way, and more than 30% of voters cast their ballots online. Voters could cast a vote through any device connected to the internet by using their e-National identification cards (Kizhakkedathil, 2016). Using the ID Card is an easy and efficient way for voters to authenticate their identity, encrypt, and sign their ballots using SHA1/SHA2 (TRUEB, 2013). Each national ID card contains a public key (used for authentication) and a private key (used for creating signatures), see Figure 1.2. Eligible voters have to download the voting software, authenticate themselves using their national ID cards, where after the system displays a candidate list and they can vote to their desire. Votes cast will be encrypted with the election's public key and signed using the voter's private key, and then stored in the vote storage server. Voters could vote several times, with the exception that only the last vote would be considered valid. That is the Estonian solution to prevent vote buying. Besides, the Estonian government allowed its citizens to use their smartphones to cast a vote, but with a special SIM card for the authentication and signing purposes. In the 2013 election, about 9% of online votes cast their ballots through a system called Mobile-ID (Reid and Harrigan, 2013).

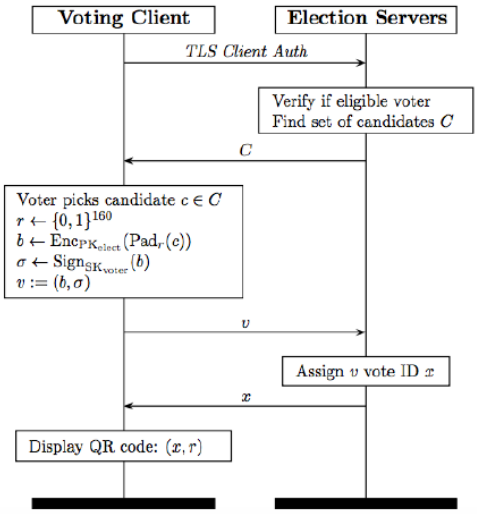

Figure 1.2. Vote casting structure in the Estonian i-voting system.

Norwegian i-Voting System: In 2011, Norway used an electronic voting system based on ElGamal encryption (Schnorr and Jakobsson, 2000), the system was designed for the country council elections for the first time. Then the system was developed by a Spanish electronic voting vendor company (Scytl) and its protocol was similar to the Estonian e-Voting protocol. Norway stopped using the system in 2014 due to security concerns (Modernisation, 2014). The major criticism that the Norwegian internet voting system faced was the fear of votes going public in case of a cyber-attack.

New South Wales i-Vote System**:** In 2015, in the New South Wales State election around 280,000 eligible voters cast their votes online using i-Vote system (Brightwell et al., 2015). I-Vote was developed by Scytl Company as well, though their system had a different protocol than the Norwegian e-Voting system. The voter has to register with authorities, after which they receive a voter ID and choose a six-digit PIN. Once voters log in to the system using their ID and PINs and cast a vote, they receive a 12-digit receipt number to confirm the process. Voters can also verify whether their votes have been successfully submitted or not. Also, after the election ends, voters can still use their 12-digit receipt number to check whether their votes were included in the tally presence or not, and why.

In 2000, e-Voting was used in the US elections (Wolchok et al., 2012). Even though it was experimental in some parts of Florida, it was a milestone in the development of electronic voting.

1.4. Motivation

Since the advent of the internet, researchers have been searching to find a secure way to transfer money, votes and other services that facilitate public life over the medium. However, the first nation to use online voting became Estonia. According to the latest statistics, more than eighty percent of the Estonians use internet for their daily activities. In 2015, increasingly more (above 30 %) of the voting participation of Estonia cast their ballots online (URL-4). Lately, 88% of Estonians used their ID card actively to access Estonia's various e-services databases. Such a large number of people using online services, as demonstrated by Estonians, illustrates that there is now a possibility to upgrade our methods from traditional, paper-based elections to digital elections. Especially nowadays with the emergence of the Blockchain, which provides infinite immutability to secure votes against manipulation and fraud.

1.5. The Problem Statement and Research Questions

For a long time, in fact since the 1838 parliamentary elections in Britain, researchers have been obsessed with e-Voting. The integrity of elections was and still is the biggest concern of critics, especially in regards to vote security. In 2005, Estonia proved that it is possible to accomplish the nearly impossible, especially after the noted participation of its citizens via the internet.

Many obstacles and questions were raised about the integrity and fairness of e-elections. The most common question is: should e-Voting be trusted. In fact, the traditional paper-based elections cannot be completely trusted either, and history is full of rigged, manipulated and counterfeited elections. So the question should not be whether the traditional paper-based polling is better than its electronic counterpart, but rather whether it is possible to achieve the best possible protection for the e-Voting systems against the various cyber-attacks that exist in the present day. Blockchain technique proved that our money (e.g., Bitcoin) is safe with it, so why

should it not be used to secure our votes/choices and guarantee the integrity of our democracies?

This research attempts to obtain satisfactory responses to the following questions:

1) Is it possible to have a voting system that satisfies all parties (Candidates, parties, election authority, and voters)?
2) How does the current research core category relate to previously presented literature?
3) Does combining Blockchain technology with the public key algorithm in an e-Voting system dispel common vulnerabilities like impersonation, ballot privacy disclosures, and manipulation?
4) How can the use of Blockchain instead of the traditional database ensure immutability of the results?

1.6. Novelty and Contributions of This Thesis

Researchers and developers have made many good attempts to introduce various voting system protocols using traditional databases and asymmetric encryption methods prior to the emergence of the Bitcoin. Researchers then used the Blockchain as a base for their systems for many reasons:

1) Prevent casting of double votes
2) Provide strong data immutability
3) And most importantly, provide the decentralization offered by Blockchain

However, there have always been many obstacles to these attempts. The fact is that the Blockchain is weak against Majority Attacks (Eyal and Sirer, 2018); as well as the type and nature of the motive for the transaction validator since the network is public; lastly some other vulnerabilities, such as Cybill Attacks (See 5.2.3), which pose a significant threat to public networks. In our research, we compared the public and private Blockchains. In order to overcome the weaknesses resulting from the public Blockchain, we suggested using two private Blockchains, one to store votes and the other to store the identities of those casting ballots. A new structure re-designed for vote identity transaction, though based on the same Bitcoin transactions cryptographic standards, was also suggested.

In general, after studying several previous attempts to build a fair voting system, we proposed a new scheme using the existing protocols (e.g., Zero Knowledge Proof Protocol), as well as robust encryption algorithms such as asymmetric encryption and Blind Signature algorithms. Lastly, we used the Blockchain technique after modifying it to suit the requirements of our current system. In this way, we introduced a fair e-Voting system based on the Blockchain technology structure.

1.7. Thesis Outline

This thesis is composed of six chapters:

1) Chapter 1: Introduction to the topic with the research problem, as well as a literature review of the e-Voting and Blockchain.
2) Chapter 2: Methodology of the thesis, detailing the properties and requirements of the e-Voting protocol and other techniques or algorithms used in the protocol.
3) Chapter 3: Most important chapter of the thesis, detailing proposed protocol indicating the whole design scheme of the implementation.
4) Chapter 4: Practical implementation of the protocol by creating a sample voting application.
5) Chapter 5: Evaluation of the proposed protocol and the threat model.
6) Chapter 6: Conclusion of the protocol and implementation. Limitations and recommendations.

2. METHODOLOGY

2.1. e-Voting

High security is essential to elections as democracy relies on the fairness of the electoral process. There have been a lot of attempts by cryptographers to come up with secure e-Voting. Many scientific researchers have worked strenuously to achieve security, privacy and integrity in e-Voting systems by improving cryptographic protocols (Cetinkaya and Cetinkaya, 2007;Cranor and Cytron, 1997; Fujioka et al., 1992). The primary interest is sufficient security in e-Voting systems, but, which properties must be justified so we can say that the system protocol is secure enough for implementation?

On one hand, the voting process must be private and the ballots anonymous. On the other hand, voters' identity must be authenticated to assure that only eligible voters are capable of casting votes. Therefore, e-Voting must be secure, confidential, verifiable and uniform. In the following list, we define the most crucial requirements of e-Voting.

1) Only eligible voters are capable of casting a vote, which in turn is included in the computation of the final tally.
2) Non-eligible voters are disenfranchised from voting.
3) The eligible voter must be able to only cast one countable ballot.
4) Privacy: voting must be private and the counted votes anonymous.
5) Possibility for Electoral Authority Representatives (EARs) to check whether all correctly cast votes are included in the final tally, means that auditors or EARs can verify the correctness of voting.
6) Election results must not be announced until the end of the election day. Any party other than EAR must not be capable of revealing the results of the election prematurely. Additionally, the system should guarantee that EAR must not expose the preliminary tally before the end of voting. Otherwise, the voting results could influence voters' decisions during the voting.
7) All valid ballots are tallied accurately, and the system outputs the final tally.

2.2. Asymmetric Encryption

In real-world voting systems, asymmetric cryptography is heavily used to de/encrypt or sign ballots. Based on algorithms like RSA (Rivest et al., 1978), the “classical” way is used to gain the advantage of this technique. Thereby, each voter and the election server maintain a key-pair. Asymmetric encryption uses two separate keys to encrypt/decrypt data as shown in Figure 1.2. Each key is connected to the other mathematically. The RSA algorithm is illustrated below.

1) Select two primes p and q.
2) Calculate the modulus $n = pq$,
3) Calculate the Euler’s totient, $\phi(n) = (p-1)(q-1)$.
4) Select $e \in [1, \phi(n) - 1]$.
5) Calculate the modular multiplicative inverse of $\phi(n)$ as d which ensures $ed = 1 \bmod \phi(n)$.
6) Define (e, n) as a public key and (p,q,d) as the private key
7) Encryption: Given plaintext m, gives the ciphertext c:

$$c = m^e \bmod n$$

8) Decryption: Given ciphertext c, gives the message m:

$$m = c^d \bmod n$$

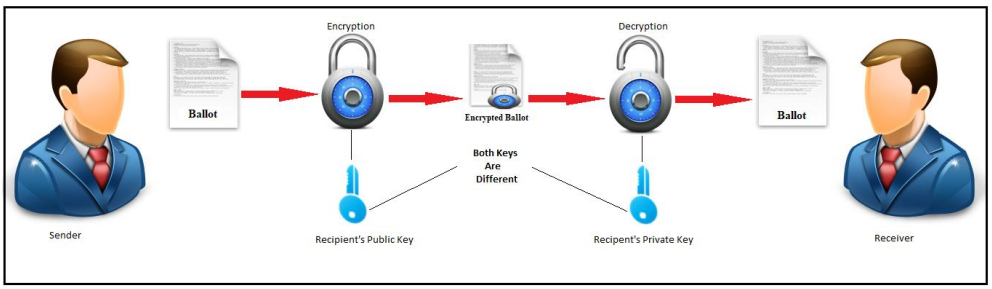

Figure 2.1. Public key ballot encryption example.

In general, in asymmetric encryption two keys can be described as follows:

1) Public key: The public key is used to encrypt data and it is openly available to the public.
2) Private Key: Also called a secret key, it is used for decryption of data that has been encrypted with the corresponding public key. The data code could only be

decrypted by the private key, as shown in Figure 2.1. In case of the loss of the private key, the encrypted data cannot be decrypted at all.

2.3. Blind Signature Cryptosystem

David Chaum introduced the blind signature as a form of the digital signature where message content is blinded before being signed (Chaum, 1981). Hence the resulting blinded message can be verified publicly. The blind signature is typically used in privacy-related protocols where the signer and message author are different parties.

The basic idea of the blind signature could be taken from blindly signed documents using carbon paper lined envelopes in the real world, where signing over the outer envelope leaves a copy signature on the inner document.

Considering ballot reliability and secrecy, voters might want to authorize their ballot and verify their vote tally, all without revealing their identity. This could be solved by using two sealed carbon paper lined inner and outer envelopes, into which the voter can place their ballot along with their return address, and send it to the trustees. Trustees, in turn, check rolls to make sure the voter is eligible to cast a vote, then take the inside envelope, sign it and send it back to the voter in a new signed outer envelope. By doing so, only eligible voters will be authorized to cast a vote in complete anonymity (Bellovin, 2009). The logic diagram is depicted in Figure 2.2.

The properties of blind signature are listed as follows:

1) Untrackable Voters
2) Anonymous, certified ballot
3) Inability to duplicate trustee's signature

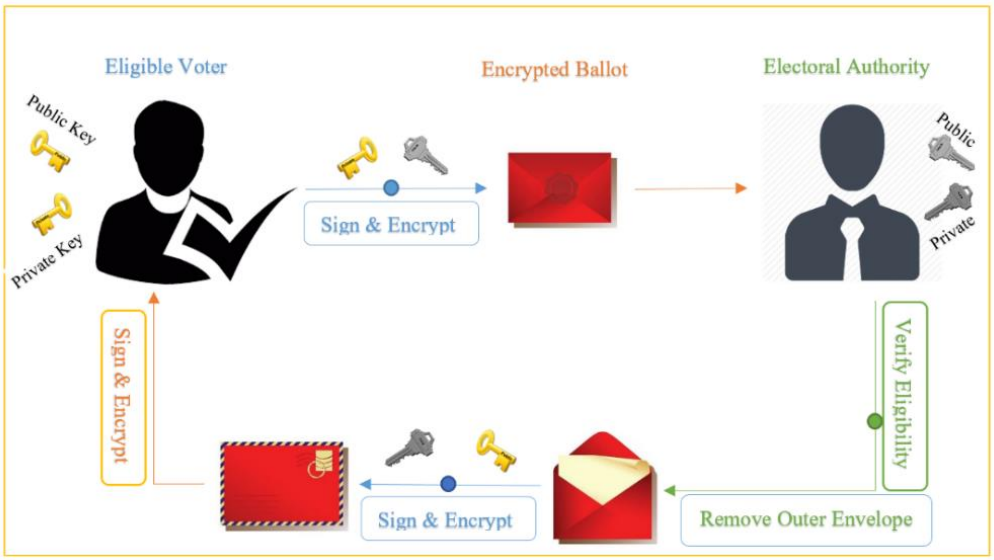

Figure 2.2. Blind signature structure.

Practically, the message is disguised to create an electronic fingerprint (signature), and the electronic signature is encrypted using the owner's private key, resulting in a digital signature attached to the document. Then, to verify the signature validity, the receiver uses the corresponding public key to decode the signature. If the signature decryption (camouflage) succeeds with no failure, this means that the sender has signed the document correctly. Since any change to this document, however small it may be, leads to a process failure during verification.

2.4. Digital Certificates

Digital certificates prove that the encrypted message is already encrypted by using the real public key sent by the original sender (Lopez, Oppliger, and Pernul, 2005). Also, these ensure that the message is only genuinely decrypted by the trusted original sender.

In this case, digital certificates use the digital certificate technology, which acts as the passport with which the original sender transmits data. The digital certificate generally consists of the following:

1) Public key
2) Public key expiry date

3) Information about the sender: such as the name of the sender, the number, the e-mail address, etc
4) Digital Signature

All of this information is incorporated into the public key and then sent to the receiving party to be accepted. The receiving party can confirm the public key and its characteristics such as the sender's name, etc, as shown in Figure 2.3.

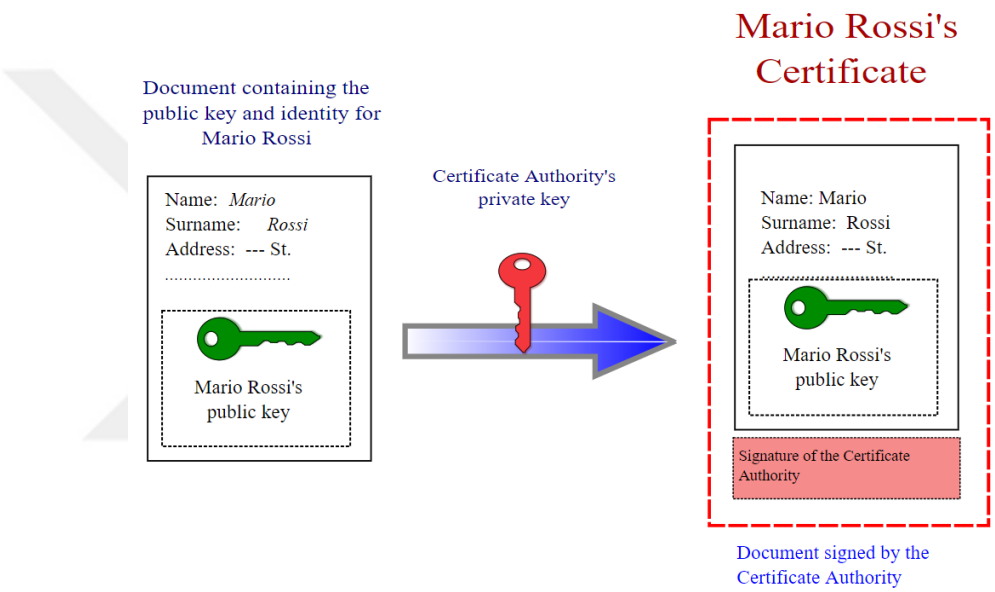

Figure 2.3. Digital certificates.

2.5. Blockchain

The term Blockchain refers to the process of producing successive blocks in the digital currency that is sequentially mined. Blockchain technology is just a new way to keep agreements.

In recent years, many improvements to Blockchain have been developed. There is a new version of the Blockchain which is the private type. Both the public and private Blockchains have many similarities such as:

1) Both are distributed by way of peer-to-peer networks, where each node maintains a ledger (replica) of digitally signed transactions.
2) Both guarantees ledger's immutability and resistance against many kinds of faulty

or malicious cyber-attacks.

Yet both also have different properties, as shown in Table 2.1 below:

Table 2.1. Differences between public and private blockchains.

Type	Public	Private
Access	Requires no permission, anyone can participate	Has restrictions on who is allowed to participate
Typical Implementation	As a public Blockchain application	Via a private Blockchain implementation
Blockchain Governance	New Business models	Equal weight to all participants
Number of users	Open	Limited

2.5.1. Public blockchain

The public Blockchain can be accessed across a decentralized peer-to-peer network (Nakamoto, 2009). Every node communicates with each other to exchange transactions or blocks. Hence, when a new node becomes connected to the network, it starts sending signals to show its location and presence in the network. Distribution is one of the best solutions that Blockchain provides.

Generally, each node contains a Blockchain, which in turn each consists of interconnected blocks. As illustrated in Figure 2.4, the Blockchain is formed sequentially by the blocks and each one of these blocks has the final hash value of the previous block hash.

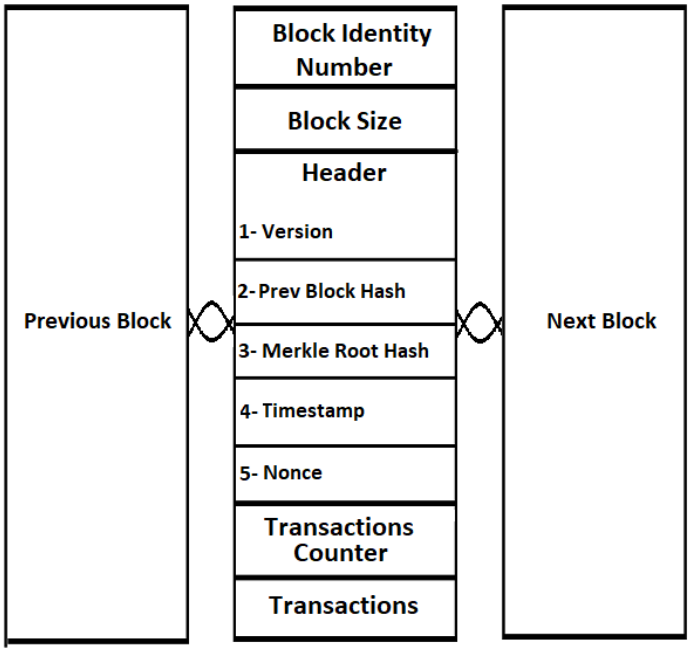

Figure 2.4. Blockchain design.

Furthermore a block, which is Blockchain's most basic element, consists of:

1) Block Identity Number: Consists of 4 bytes, used to identify block's ownership. For instance, gender identities which identify all men as male.
2) Block Size: Consists of 4 bytes, used to show the block's size including all transactions.
3) Block Header: Consists of 80 bytes, considered the most important element in the block. The header contains:
    a) Version: Consists of 4 bytes, shows the current chain version. If block elements changed, a new version would be released and referenced as well.
    b) Prev-Block Hash: Consists of 32 bytes, the hash value of the previous block element. Including the previous block's hash in the new one ensures the blocks' validation all the way back to the 1st block ever in the chain. Chain validation happens every time Blockchain changes. Especially, by referring to the previous blocks by their hash value, the chain will be protected against modifications. For instance, if someone changes a block by modifying any data, they must re-hash the modified block, which will have a new hash.

Therefore, they also must re-hash all blocks again, all the way up to the top of the chain. Thus, it is computationally almost impossible to achieve.

c) Merkle Root: Consists of 32 bytes, cryptographic proof of which transactions are in the block and which order they are in. It provides a convenient piece of information to be included in the block header, which is hashed and then added in the next block header. Without the Merkle root in the block header, blocks will not have any cryptographic proof of which transactions are included nor any proof that their contents have not been tampered.

Supposing a block consists of eight transactions, (a) to (h), so according to the transactions count, a four-tier pyramid will be considered. Pyramid's base contains the hash of each of the eight transactions. Every two hash from the base tier will be combined separately and encrypted again to produce an element of the third tier. The third tier will contain four hash combinations. The previous procedure will be followed to create each one of the second tier elements. Lastly, the top hash (Merkle root) will be a combination of the two elements of the second tier, as seen in Figure 2.5.

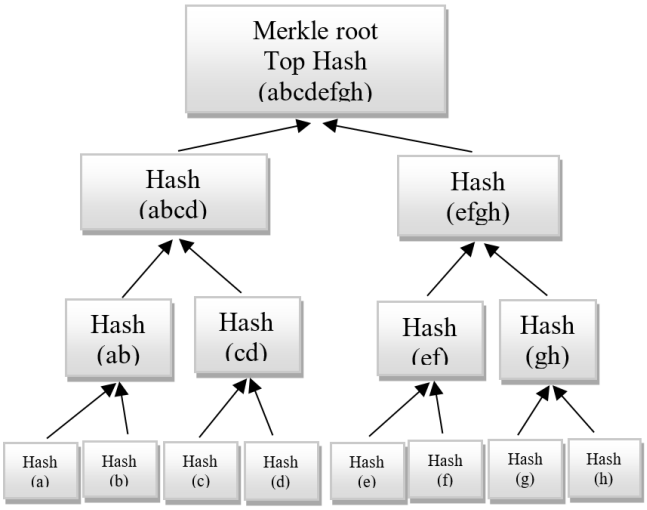

Figure 2.5. Merkle root tree.

Using Merkle tree is preferable over a hash chain or a hash of concatenated

transactions because it allows a much quicker and simpler test of whether a particular transaction is included in the set or not.

d) Timestamp: Consists of 4 bytes, shows the current timestamp.

e) Nonce (Proof of work): Consists of 4 bytes. The nonce is the puzzle that the miner is solving in every mining process. Looking more closely, the miner takes a random number, starting from zero as a value for the nonces and appends it to the block hash and then hashes it all together to obtain the final output. If the final hash is higher than the Bits target, the miner increases the nonce value by one and repeats the hashing process until he gets a smaller value than the Bits target. In fact, it is not a real puzzle; it is computation-heavy yet solvable problem.

4) Transaction Counter: Consists of 4 bytes, shows how many transactions the block holds.
5) Transaction: Contains money transactions, votes or any kind of digitized transactions.

The beauty and sophistication of Blockchain technology is in its interconnection and resistance against penetration. As mentioned earlier, each block of the chain contains the top hash value of the previous block. Any change in a block will change the value of the final hash of the block itself, thus manipulation is easily detected, because the chain becomes invalid. Then, the original version will be restored by a copy from other nodes.

2.5.2. Private blockchain

The private Blockchain, also known as a permissioned Blockchain, works similarly to the public Blockchain, except for the restrictions of the participators (URL-6). Private Blockchains embody the following properties:

1) Controlled access: Authorities can only allow authenticated nodes to join their network. Also, new users and their identity could be validated by the Authority. This means that the authority can only give access to specific information to eligible users after having verified them, as shown in Figure 2.6.
2) Performance at large scale: It is not uncommon for large businesses to process

hundreds of thousands of transactions per second (TPS). Thus, permissioned Blockchains must scale so that they can provide performance accordingly.

3) Resilience: They must have built-in redundancy, automated monitoring, and require minimal human intervention.
4) Confidentiality: Permissioned Blockchains should encrypt data to assure maximum security. To achieve that, it can follow the cryptographic standards of the public Blockchains to secure transaction messages and ensure transaction authenticity.
5) Consensus: Permissioned Blockchain transactions are involved, for instance, sending assets from one address to another.

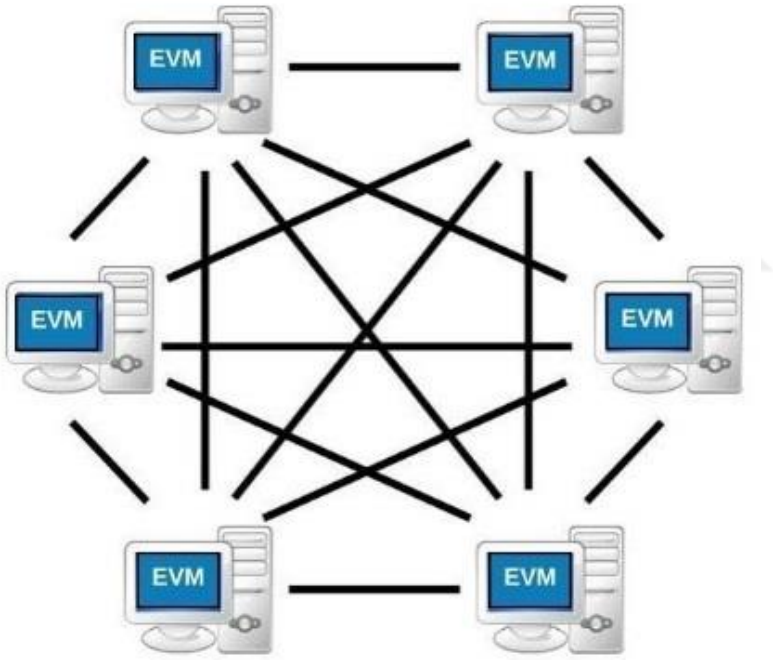

Figure 2.6. Private blockchain network.

3. PROTOCOL

3.1. Outline

As seen in Figure 3.1. Our e-Voting system protocol consists of six sequential phases as shown in Figure 3.1., where the very first stages are executed automatically in the backend along with the last two. As for the middle stages, they are performed by voters.

1. Publishing Phase: EAR could set up the project some days before the election in order to allow voters to enter and establish their eIDs.
2. Login Phase: Eligible voters will be able to log in to the system using their full name, National ID Card number, and National ID Card's expiration date, to create their eIDs.
3. eID Establishing Phase: Once voters enter the system, they must generate asymmetric key pairs to establish unique eID. The eID will be used for identifying voters eligible to cast votes.
4. Re-login Phase: During the election event, voters must use their eIDs to identify themselves to the system in order to cast a vote, where the registration will be closed.
5. Voting Phase: After voters re-login to the system using their private keys to prove their identity, they would be able to cast a ballot. Their eIDs will be stored in the first Blockchain and the second one will contain the votes.
6. Tallying Phase: Tallying stage is strictly under EAR supervision, people must not know the results before the end of the event.

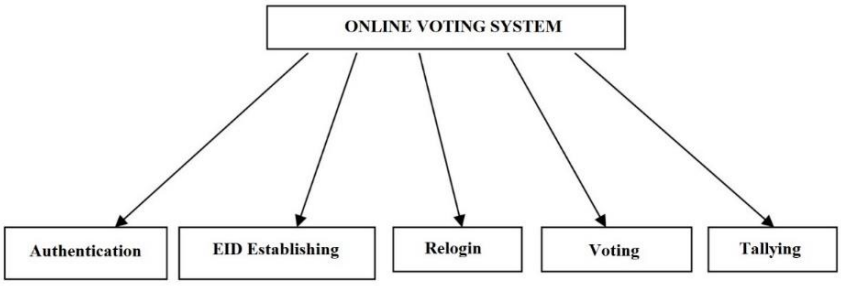

Figure 3.1. Ballot casting phases.

3.2. Assumptions

To promote privacy, the protocol is under the following assumptions:

1) The EAR will not correspond under any pressure or temptation.
2) The hashing algorithm SHA-256 and 1024-bit RSA keys are secure.
3) Voter will be able to follow the stages. Despite its difficulties, even if there was a complexity, to enroll the voting process.
4) Nowadays, almost every individual has their own cell phone, which gives us the possibility to use it to secure the voters' data from getting exposed.
5) Voters will follow user protection instructions to ensure they are not exposed to any kind of cyber-attack.

3.3. e-Voting Scheme

3.3.1. Preparation phase

It will be possible to keep the election system open to the public a number of days before the election in order to allow voters to enter and establish their eID. This way, the EARs will have enough time to correct voter data if there will be any mistakes. Meanwhile, voters will become familiar with the election system and check whether someone else has used their data to create an eID. If so, they can inform the EAR to receive the necessary assistance. Even after establishing the eID, attackers can hack the eID of an eligible voter and change the phone number to redirect the privileges to their favor and these will be exposed. The moment voters try to login to the system for the second time, their eID will be corrupted because their data has changed. As a result, they can immediately report the issue to EAR. As the voter completes establishing their eID successfully, they can log out of the system, and re-login during the polling day to vote. As a general rule, eID will remain valid forever once obtained.

3.3.2. Login phase

We assume that the country where a digital voting based election will be held has its own e-government where every single citizen is registered and has their private e-number. Therefore, it is possible to make the very first voter's authentication stage by asking the full voter name, national ID card number (NIDN) and national electronic

number (NEN). After that, voters must identify themselves via a verification code received on their mobile number which they previously registered with the EAR. Then all the requested data will be encrypted together. Hence, computing of the resulting hash is performed by the following equation:

$$\text{Hash} = (\text{Enc}(\text{FullName} + \text{NIDN} + \text{NEN})) \quad (3.1)$$

The computed hash is used in the browser cookie to identify a voter. Web applications which use cookies are known to be vulnerable to XSS (Cross-site Scripting) Attacks (URL-7), so unauthorized accesses might violate voter privacy. Using HTTPS protocol is insufficient as an attacker can request a page with HTTP, and as a result the same cookie will be sent without any protection. In order to overcome this issue, system pages must only be sent to HTTPS calls before starting the session. Furthermore, the browser should be configured to not allow access JavaScript into the session cookie.

3.3.3. eID generation phase

As voters will identify themselves in the login step, they must have first created an electronic identification (Reid and Harrigan), consisting of an ID and digital signature. "eID" is an electronic identification solution for citizens (Lyon, 2009). The idea behind creating a digital ID card is to ascertain voter identity and to be verified by Electoral Authority's Representative (EAR, more on this below) while using it as an identity in the voting processes. Creating a digital ID card is essential to generate the public-private key and use these to maintain ballot/voter privacy.

After voters login to the system, a public-private key set will automatically be established. The public key will be stored in the system, open to the public. The second key, the private one, will only be known and used by the voter. Indeed, it must not fall into the hands of others, because someone might use it to impersonate the voter or use it for signature purposes. After that, some information about the person will be requested to be matched with the stored data in the system's database, in order to verify the voters' credibility. As illustrated in Figure 3.2. The required information consists of the following fields:

1) Voter’s National Identification Number (NIDN) and its expiration date (NIDED).
2) Three images of the voter, captured right by the system using a computer's webcam. The first one of the voter himself. The second image of the first side of the voter's identification card and the last picture of the other side of the voter's own identification card.
3) Voter’s phone number.

However, voter identification card number is requested for the second time for confirmation purposes. In order to not depend on the availability of biometric devices (Fingerprint, Eyeprint), it is necessary to invoke another solution, such as photographing, encryption and storage of these required images in the database. These images will prove that the voter has created their own electronic identity.

Voter’s phone numbers will be utilized in the process of creating the time-based, one-time password. It will be used to verify voter identity while logging in to the system as well as every process after that, including casting a vote. Finally, all the requested fields are combined and encrypted with the voter’s public key to obtain a verifiable eID hash.

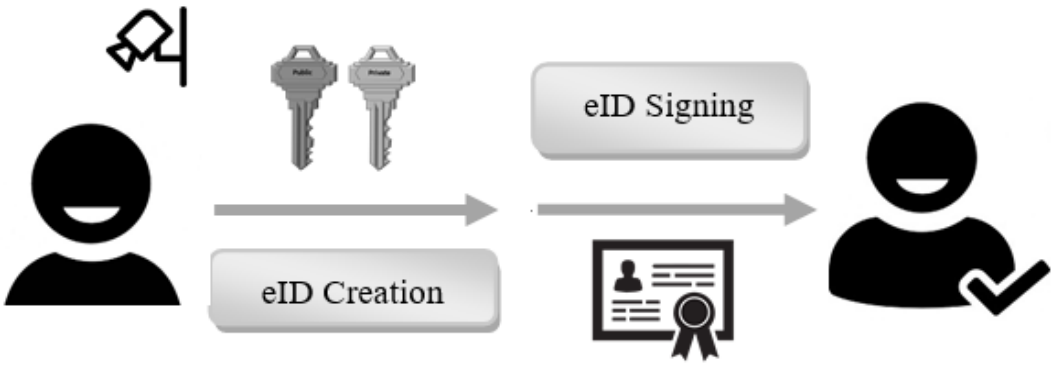

Figure 3.2. User authentication.

$$\text{eID} = \left\{ \begin{matrix} \text{FullName} + \text{NIDN} + \text{NIDED} + \text{Images} \\ + \text{PhoneNumber} \end{matrix} \right\}_{\text{Voter's Public Key}} \quad (3.2)$$

Voters will create a unique signature by using their private key, allowing others to check the validity of the signature using their public key for the purpose of verification. The signature will be used by the voter to sign the ballot to prove ownership of the private key without requiring to reveal it.

$$\text{Signature} = [\text{eID}]_{\text{Voter's Private Key}} \tag{3.3}$$

So far, creating the eID and electronic signature instead of relying on self-created ID of the database management system is to increase the e-Voting system's security level.

Obligating the voters to capture photos of themselves and their ID cards is part of the system's processes. These images allow the EAR to verify whether the voters have established their eID or there are impersonations, especially the impersonation of deceased voters. These will be converted to bits format, encrypted and stored in the database. Also, it will be part of the eID hash, so it will not be possible to decrypt, change or manipulate images.

3.3.4. Re-login phase

Serious consideration must be given to protecting voter data from being used by others; time-based one-time password algorithm (M'Raihi et al., 2011) may be used to preclude this issue. Time synchronization is essential in time-based passwords, a secret key and a timestamp will be defined and everything will be synchronized via a standard protocol such as network time protocol. Once the password token is created, it can be used within a limited duration to ensure that the voter is using their data in the login process, or the ongoing operation is within their knowledge because they are the one who receives the verification code on their cell phone.

Let us describe the steps of this phase in order as follows:

1) Voter opens the system links
2) Voter logs in to the system using the private key to effect authentication
3) Voter will receive an expirable verification code on the cell phone to verify

oneself again after using the private key

Once voter logs in to the system for the second time in order to vote, the system will check whether or not the electronic identity has been successfully established. In case everything was done in proper order, voter will receive a time-based verification code (PIN) on the phone. The PIN could last for two minutes. Also, voter must confirm own identity using the private key.

3.3.5. Voting phase

Basically, in any election employing secret ballot/open counting, the following points must be considered:

1) Vote by secret ballot: a vote's originator must be unknown. More clearly, a voter's ballot shall not be violated by anyone anyhow, which means it must be concealed entirely whether directed to a candidate or null.
2) The voter must be marked in the electoral registry as having voted in order to prevent duplicate voting or cheating and to allow only legitimate voters to cast ballots.
3) Open counting: the outcome of the election should be determined by an open-verifiable counting of the votes.

Unlike the Bitcoin (Nakamoto, 2009), which uses a single public Blockchain, the proposed digital voting system will use two private(/permissioned) Blockchains. Let us call the first Blockchain as BL-i, the voters' identities, and the second one as BL-v, the encrypted ballots. The use of permissioned Blockchain limits the parties who can read the information, also restricting the nodes and separating these in different locations around the country.

3.3.5.1 Generation of addresses

Every transaction requires candidate's new address in order to keep the vote-voter anonymity every single time voters cast votes (Dunphy and Petitcolas, 2018). The generated address will be specified by the recipient to receive the vote credit.

Address generation is done by:

1) Generating a corresponding public key from the ballot's private key by using the elliptic-curve factorization method (ECM) (Constantinescu, 2010),
2) Encrypting the result with the SHA-256 hash algorithm to gain a 265-bit long hash for this particular public key.
3) Reducing the hash from 256 to 160 bits by encrypting it again with the RIPEMD-160 hash algorithm (Preneel et al., 1997). The generated public key hash is the actual hash which will be specified by the recipient to receive the vote credit.
4) Adding one-byte prefix to the front of the public key hash to specify its version.
5) Generating a checksum by hashing the result twice with the SHA-256 algorithm,
6) Obtaining a 256-bit hash result from the second SHA-256 including the checksum as the first four bits of the particular hash.
7) Appending the four bits checksum to the end of the public key hash. Thus, address hash will start with the version and finish with a checksum.
8) Converting the resultant public-key hash from a string type to Base58Check encoding.

See the address generation flowchart in Figure 3.3.

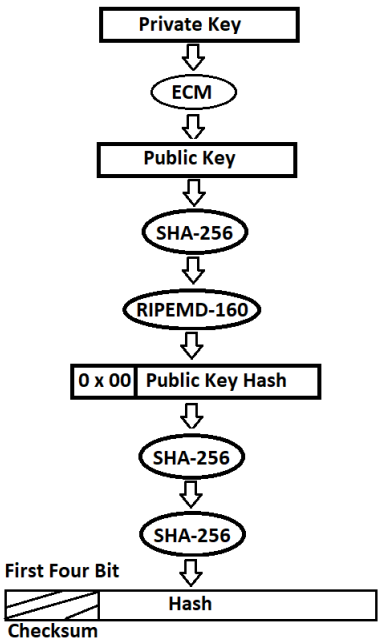

Figure 3.3. Address generation flowchart.

$$\text{Public key hash} = [\text{Version} + \#\#\#\#\#\#\#\#\#\#\#\#\# + \text{Checksum}] \quad (3.4)$$

Therefore, sending a checksum with an address permits verification that the address was not manipulated. Also, by making the communications between voter and candidate go through a secure channel using a Zero-Knowledge Protocol (Quisquater et al., 1989), as illustrated in Figure 3.4, prevents any possibility of a Men-In-The-Middle Attack.

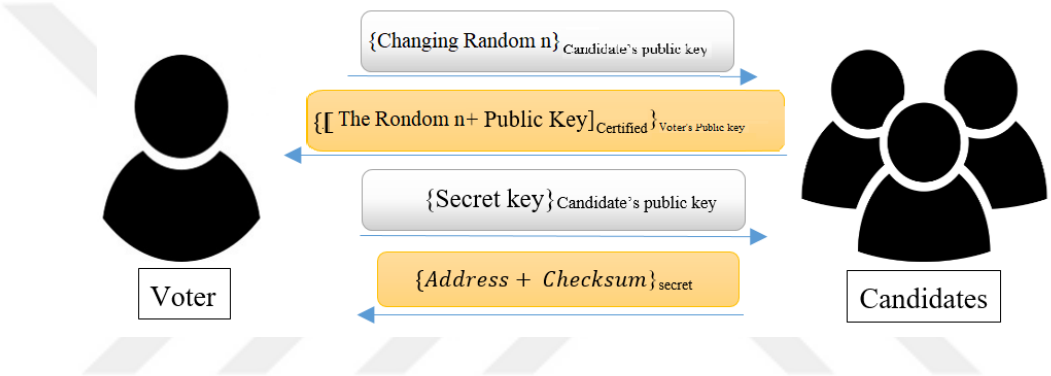

Figure 3.4. Securely obtaining address diagram.

3.3.5.2 Generation of transactions

Each voter has the right to cast only one countable vote. Once the voting starts, voters have to sign their ballots from the EAR anonymously using the blind signature. The blind signature scheme is described as follows:

EAR owns a signing function S'EA. The corresponding publically known inverse SEA, which satisfies SEA(S'EA(s)) = s, but gives no clue about S'EA. To obtain EAR's signature of the transaction t without revealing it, Alice will depend on a computing function CVoter and its inverse C'Voter, both of which belong to him/her only, and satisfy the condition that C'Voter(S'EA(CVoter(s))) = S'EA(s) while CVoter and S'EA give no clue about s. The signing scheme is presented as follows.

1) Voter sends $C_{Voter}(t)$ to EA.
2) Bob receives $C_{Voter}(t)$ check the eligibility and signs it using S'_{EA} to obtain $S'EA(C_{Voter}(t))$. Then he sends S'_{EA} $(C_{Voter}(t))$ back to the voter.

3) The Voter uses C'_{Voter} to obtain S'_{EA} (t) according to $C'_{Voter}(S'_{EA}(C_{Voter}(t))) = S'EA(t)$.

The steps above demonstrate how voters could obtain S'EA (TxID), the signature of the transaction signed by EAR, without revealing the transaction. After a voter signs the ballot and submits it, the voter will specify a candidate/party and cast the vote. Every voter has one value credit (Signed Ballot) to be spent once by giving it to a specific candidate.

Once a ballot is cast, the addresses will be generated, and the transaction is created containing the following:

1) TxID. Formed by encrypting the transaction data twice with the SHA-256 algorithm.
2) Version. Transaction version.
3) Size.
4) Prev-Out. Presents whether the voter has voted before or not.
5) Out. Refers to a candidate's address.
6) SEA. Contains the S'EA(t).
7) See Table 3.1.

Table 3.1. BL-v transaction.

TxID	f4184fc596403b9d638783cf57adfe4c75c605f6356fbc91338530e9831e9e16
Version	2
Size	8 bytes
Timestamp	DateTime
P_{Out}	Previous Out
Out	{ Recipient Address } $_{\text{Recipient Public Key}}$
Signature	[Transaction Signature] $_{EAR}$

The BL-v transaction is created by including Transaction version, Transaction size,

Prev-out, and the Out. The Out contains the recipient wallet address encrypted with his public key. However, the recipient will prove that he is the owner of the private key; only then, he gets the vote. Whereas the TxID established by including all the transaction stuff together, encrypted with SHA-256 twice as follow:

$$\text{TxID} = \text{SHA256.Enc}\left(\text{SHA256.Hash}\left(\text{Version} + \text{Size} + \text{eID} + \text{POut} + \text{Out}\left(\text{Recipient Address}\right)\right)\right) \quad (3.5)$$

As for the BL-i transaction it will include (Transaction version, Transaction size, Pout, and the Out), as shown in Table 3.2. The Out contains the recipient wallet address encrypted with voter's public key to concealing it.

Table 3.2. BL-i transaction.

TxID	Transaction ID
Version	0.2
Size	4 bytes
eID	Voter's eID
Timestamp	DateTime
P_{Out}	Previous Out
Out	{Recipient Address} Voter's Public Key
Signature	[Transaction Signature] Voter's Private Key

3.3.5.3 Block generation

During this stage, voters establish and submit their transactions to the network. For a transaction to be accepted, it is must be valid and included in a block. Also, it must have the EAR's signature included in the transaction. Voters' identity transaction will be stored in the BL-i Blockchain and the ballot transaction stored in the BL-v Blockchain. As mentioned previously, voters can vote multiple times as long as only one of the cast votes is registered and counted. So voters will cast transactions containing their eID and the recipients' addresses encrypted and signed by their own public key. These are then posted to the BL-i. Once the BL-i blocks are successfully created, the recipients' addresses will be decrypted by voters and encrypted again with the recipients' public keys to create BL-v blocks. Eventually, the transactions will be posted with the one value credits to the BL-v. Let us summarize this stage as follows:

1) A voter's first cast vote will include no P_{Out} of the BL-i transaction and false value in P_{Out} of the BL-v transaction. After that, if there is an overwriting vote, it will consist of the previous TxID in the BL-i and the previous out value in BL-v.
2) BL-v does not have any clue leading to the voter.
3) The EAR must sign BL-v.
4) BL-v blocks will be created after creating the BL-i to check the previous cast votes.

3.3.6. Tallying phase

Vote tallying will be one of the responsibilities of the EAR for that constituency. EARs are (virtual agent) persons or institutions authorized by the network and directed by the state to perform audit and certification of the votes/voters. An EAR could be assigned to every polling station or even one per district.

Candidates/parties will get a summary ballot tallying just the votes received; furthermore, it will be entirely controlled by the EARs. This is how the tallying of votes will happen automatically. The EAR will verify either Blockchains, also counting or listing all voters, thus validating who cast their votes and who did not.

The EAR must certify and validify votes before they are cast, while an unsigned ballot will be disregarded. As a result, we had a secure system and distributed unchangeable data history of the election. The system protocol is shown in Figure 3.5.

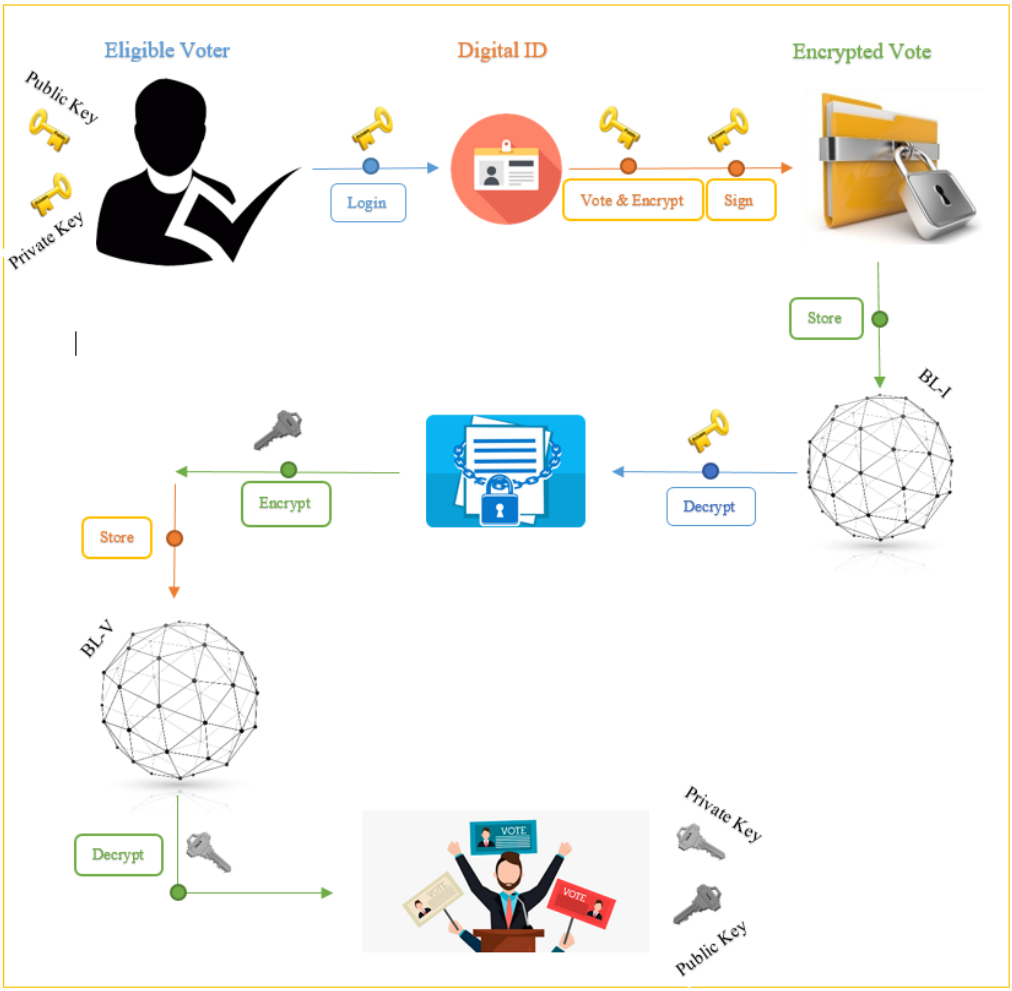

Figure 3.5. Voting process structure.

4. IMPLEMENTATION

This e-Voting system is designed over a double private Blockchains. The system satisfies the properties of voter/vote privacy, eligibility, verifiability, etc. When implemented, the system must consider the basic models and the features of the e-Voting. In general, this system allows voters to register in the system, vote and verify whether their vote has been counted or not.

This system should have the following models:

1) Voter Login Model is the logic of logging in.
2) Voter, eID Generation Model, is the logic of establishing an eID.
3) Voter Re-Login Model is the logic of re-logging.
4) EAR Model is a model for EAR including its APIs.
5) Voting Model is a model for the voter to cast a vote.
6) Vote Verifying Model where a voter can verify his vote tally.
7) Tallying Model.

4.1. Protocol's Models and Implementation

This protocol has been developed by using MVC ASP.NET programming language (Freeman, 2014). The MVC is used to route the page and secure the system against the following vulnerabilities:

1) Cross-Site Scripting (XSS), (URL-7).
2) Cross-Site Request Forgery (CSRF), (URL-3).
3) Authentication.

Also, the .NET framework is used to enhance application security as web applications developed using ASP.NET have windows confirmation and configuration. Managed code and CLR offer safeguard properties such as role-based security and code access security.

4.1.1. Login model

In the proposed protocol, we divided the parties into voters, candidates, and EARs. The system will consist of the main entry page, shown in Figure 4.1., in which both voters and candidates will enter to establish their identity and cast votes.

Registrar's information such as full name, national ID card number, and private e-government code will be requested. Then, in case of any mistaken entry, the system will produce a warning. Furthermore, any random registration attempts will be verified and prevented using reCAPTCHA (Von et al., 2008). reCAPTCHA protects the system from spam and abuse, using an advanced risk analysis engine and adaptive CAPTCHAs to keep automated software from engaging in abusive activities on a site. reCAPTCHA allows valid users to pass through easily.

```
UserAccounts user = dBContext.UserAccounts.Where(x => x.UserName == entity.Username
&& x.Password == entity.Password && x.ID_Number ==
entity.ID_Number).FirstOrDefault();
                if (user.UserAccountId != 0)
                {
                    FormsAuthentication.SetAuthCookie(entity.Username, false);

                    Session["UserID"] = user.UserAccountId.ToString();
                    Session["UserName"] = user.UserName.ToString();

                    HttpCookie httpCookie = new HttpCookie("user",
user.UserAccountId.ToString());
                    httpCookie.Expires.AddMinutes(5);
                    HttpContext.Response.SetCookie(httpCookie);

                    Persons persons = dBContext.Persons.First(x => x.FullName ==
entity.Username && x.ID_Number == entity.ID_Number);
                    if (persons.Valid == false)
                    {
                        return RedirectToAction("Index", "IDCard");
                    }
                    else
                    {
                        var results = SMS.Send(new SMS.SMSRequest
                        {
                            from = "Number", to = "Number",
                            text = "CONFIRMATION CODE IS: " + String(4)
                        });

                        return RedirectToAction("Index", "Verify");
                    }

                }
                else
                {
                    ViewBag.LoginValidate = "Sorry UserName or Password Invalid";
                    return View();
                }
```

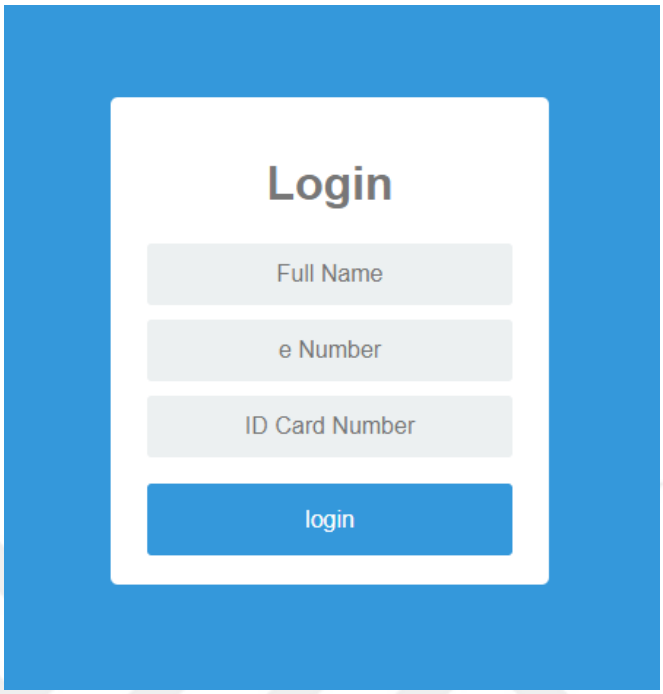

Figure 4.1. Login model.

After performing a secure login, the system will check whether voters have created their eIDs or not. If voters have created their eID, they will receive an expirable verification code on their cell phone, which they have registered during the eID creation process to identify themselves, and then are redirected to the eID verification model, the authentication protocol is shown in figure 4.2. In case they have not established an eID they will be redirected to the eID establishing model.

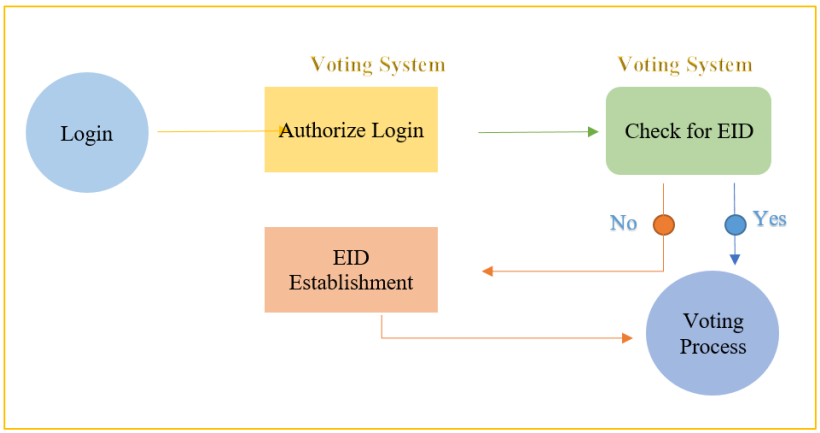

Figure 4.2. Login structure.

4.1.2. eID generation model

In case users have not created their eID, they will be redirected from the login page to the eID creation page shown in Figure 4.3. The process is listed in order as follows:

1) Generate RSA public-private keys
2) Enter their National ID card Number, ID card Expiration date and Phone Number
3) Voters take three photos; one of themselves and one of each side of their National ID cards.
4) Personally chosen passwords for protecting the eID from being used by others.
5) Encrypts all above-requested information with his public key,
6) Sign the resultant hash with voter's private key.
7) After all, voters' eID must be verified by the EAR.

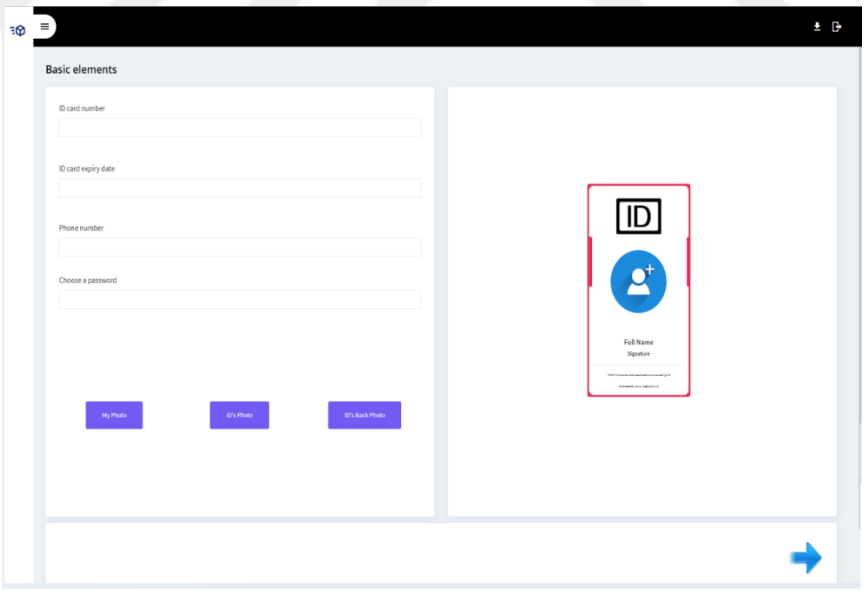

Figure 4.3. eID Creation Model.

Creation of the eID and electronic signature instead of the database's self-created IDs to increase the system's security level.

```
[HttpPost]
 public ActionResult KeyGenerator(string id, string exDate, string
password,string PhoneNumber)
        {
            try
            {

                DateTime dateTime = Convert.ToDateTime(exDate);
                BlockChainDBContext blockChainDBContext = new
BlockChainDBContext();
                VotersDBContext dBContext = new VotersDBContext();
                PersonalCertificate personalCertificate = new
PersonalCertificate(id);

                if (dBContext.Persons.Any(x => x.ID_Number == id &&
x.Expiry_Date == dateTime))
                {
                    Persons persons = dBContext.Persons.First(x => x.ID_Number
== id && x.Expiry_Date == dateTime);
                    if (!string.IsNullOrEmpty(Session["Pic"] as string) &&
!string.IsNullOrEmpty(Session["IDPic"] as string) &&
!string.IsNullOrEmpty(Session["IDBPic"] as string))
                    {
                        persons.Valid = true;
                        persons.Phone_Number = PhoneNumber;
                        persons.Pic = Session["Pic"].ToString();
                        persons.IDPic = Session["IDPic"].ToString();
                        persons.IDBPic = Session["IDBPic"].ToString();
                        persons.PersonIdHash =
personalCertificate.Encrypt(Encoding.UTF8.GetBytes(persons.PersonId + persons.
FullName + persons.BirthDate + persons.GenderId + persons.Mobile +
persons.Email + persons.ID_Number + persons.Expiry_Date + persons.IDPic +
persons.IDPic + persons.IDBPic + "//" + password));
                        persons.Signature =
personalCertificate.Sign(persons.PersonIdHash);
                        dBContext.Entry(persons).State =
System.Data.Entity.EntityState.Modified;
                        dBContext.SaveChanges();

                        HttpCookie cookie = new HttpCookie("PersonalID");
                        cookie.Value =
Convert.ToBase64String(persons.PersonIdHash);
                        Response.Cookies.Add(cookie);

                        return RedirectToAction("Index", "DownloadIDCard");
                    }

                }

            }
            catch (Exception)
            {

                throw;
            }

            return View();
        }
```

4.1.3. re-Login model

Once voters log in to the system, they must identify themselves again using the national ID card number eID, eID's password and the verification code they received on their phone, as shown in Figure 4.4.

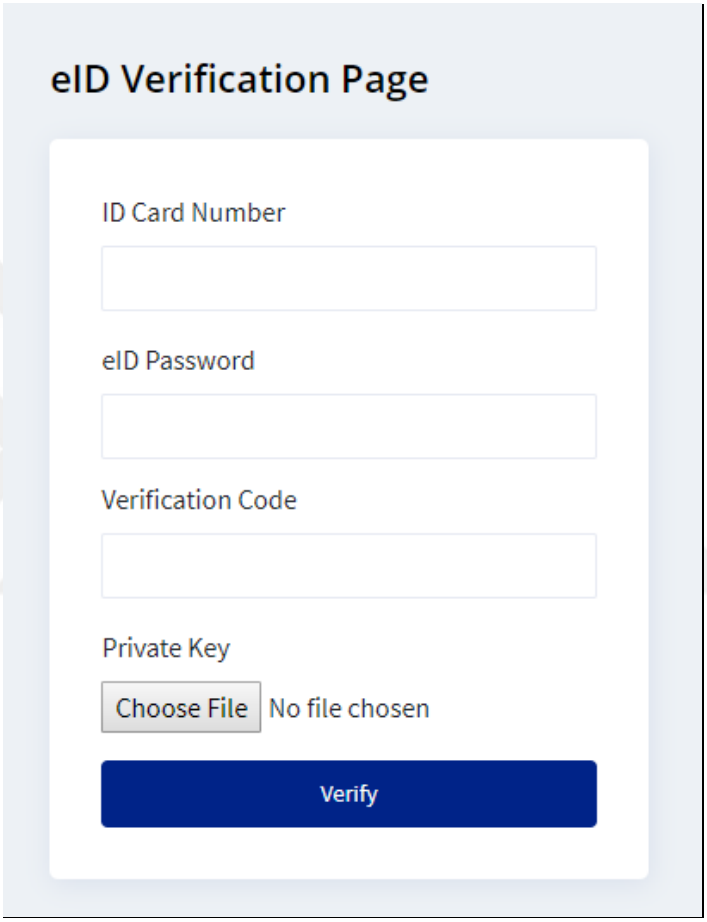

Figure 4.4. eID verification model.

If the eID's password and the verification code are correct, users will be redirected to the voting model and ready to cast votes.

4.1.3. Voting model

Only eligible voters can cast votes after establishing ballots and verifying themselves. The voting process listed in order as follows:

1) Voters will specify candidates/parties and ask for new addresses to cast votes, as illustrated in figure 4.5.

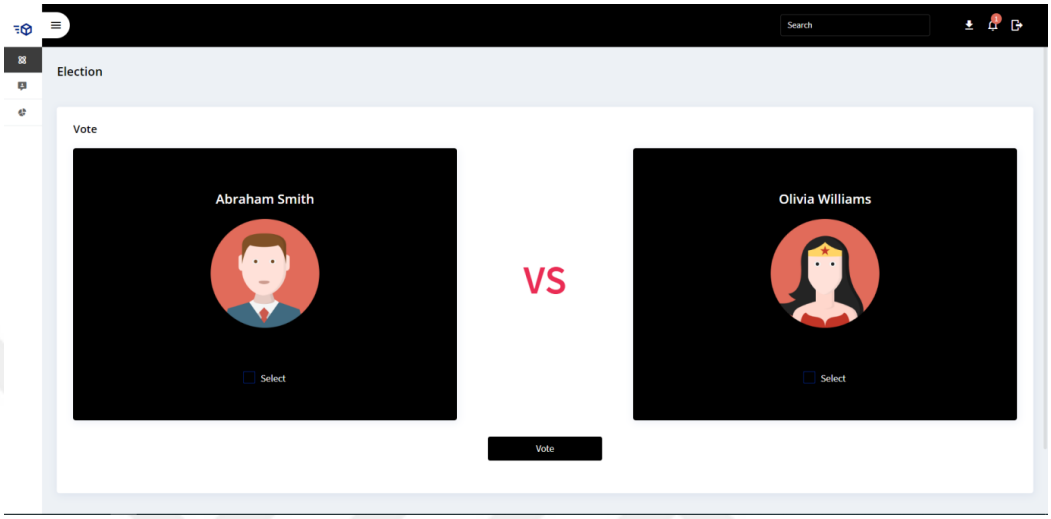

Figure 4.5. Voting model.

2) New address will be generated as requested in order to receive votes.

```
private Byte[] Address(Byte[] args)
        {
            byte[] PubKey = args;
            Console.WriteLine("Public Key:" + ByteToHex(PubKey)); //Step 1
            byte[] PubKeySha = Sha256(PubKey);
            Console.WriteLine("Sha Public Key:" + ByteToHex(PubKeySha)); //Step 2
            byte[] PubKeyShaRIPE = RipeMD160(PubKeySha);
            Console.WriteLine("Ripe Sha Public Key:" + ByteToHex(PubKeyShaRIPE));
// Step 3
            byte[] PreHashWNetwork = AppendBitcoinNetwork(PubKeyShaRIPE, 0); //
Step4
            byte[] PublicHash = Sha256(PreHashWNetwork);
            Console.WriteLine("Public Hash:" + ByteToHex(PublicHash)); // Step5
            byte[] PublicHashHash = Sha256(PublicHash);
            Console.WriteLine("Public HashHash:" + ByteToHex(PublicHashHash));
//Step 6
            Console.WriteLine("Checksum:" + ByteToHex(PublicHashHash).Substring(0,
4));  //Step 7
            byte[] Address = ConcatAddress(PreHashWNetwork, PublicHashHash);

            return (Address);
        }

public static string ByteToHex(byte[] array) […]

public static byte[] Sha256(byte[] array) […]

public static byte[] RipeMD160(byte[] array) […]

public static byte[] AppendBitcoinNetwork(byte[] RipeHash, byte Network) […]
public static string Base58Encode(byte[] array) […]
public static byte[] ConcatAddress(byte[] RipeHash, byte[] Checksum) […]
```

3) BL-i transactions will be created including voters' eID and null Pout if it is the first cast vote.

```
#region ID_Transaction
                _IDLedger.EID = PersonalHash;
                _IDLedger.Size = 8;
                _IDLedger.Version = 2;
                _IDLedger.Timestamp = DateTime.Now;
                if (blockChainDBContext.ID_Ledger.Any(x => x.EID == PersonalHash))
                {
                    _IDLedger.POut =
blockChainDBContext.ID_Ledger.OrderByDescending(x => x.Timestamp).FirstOrDefault(x
=> x.EID == PersonalHash).Out;
                }
                else
                {
                    _IDLedger.POut = null;
                }
                byte[] PK = cNDTFirst.publicKey.Modulus;
                var Addr = Address(PK);
                voteWallet.Can_ID = id;
                voteWallet.Can_Address = Addr;
                blockChainDBContext.VoteWallet.Add(voteWallet);
                _IDLedger.Out = personalCertificate.Encrypt(Addr);


                if (_VoteLedger.POut != null)
                {
                    _IDLedger.TxID =
Encoding.UTF8.GetBytes((sha256_hash(sha256_hash(Encoding.UTF8.GetString(_IDLedger.
POut) + Encoding.UTF8.GetString(_IDLedger.Out) + _IDLedger.Version.ToString() +
_IDLedger.Size + _IDLedger.EID.ToString()))));
                }
                else
                {
                    _IDLedger.TxID =
Encoding.UTF8.GetBytes((sha256_hash(sha256_hash(Encoding.UTF8.GetString(_IDLedger.
Out) + _IDLedger.Version.ToString() + _IDLedger.Size +
_IDLedger.EID.ToString()))));
                }
                _IDLedger.Signature = personalCertificate.Sign(_IDLedger.TxID);
#endregion
```

4) Voters will create BL-v transactions after creating BL-i transactions.
5) Voters will sign the transactions BL-i by themselves; then they will sign the BL-v transactions from the EARs in order to make the transactions valid and countable.

```
#region Vote_Transaction
                _VoteLedger.Size = 8;
                _VoteLedger.Version = 2;
                _VoteLedger.Timestamp = DateTime.Now;

                if (blockChainDBContext.ID_Ledger.Any(x => x.EID == PersonalHash))
                {
                    _VoteLedger.POut =
personalCertificate.Decrypt(blockChainDBContext.ID_Ledger.OrderByDescending(x =>
x.Timestamp).FirstOrDefault(x => x.EID == PersonalHash).Out);
                }
                else
                {
                    _VoteLedger.POut = null;
                }
                _VoteLedger.Out = Addr;
                if (_VoteLedger.POut != null)
                {
                    _VoteLedger.TxVId =
Encoding.UTF8.GetBytes(((sha256_hash(Encoding.UTF8.GetString(_VoteLedger.POut) +
Encoding.UTF8.GetString(_VoteLedger.Out) + _VoteLedger.Version.ToString() +
_VoteLedger.Size))));
                }
                else
                {
                    _VoteLedger.TxVId =
Encoding.UTF8.GetBytes(((sha256_hash(Encoding.UTF8.GetString(_VoteLedger.Out) +
_VoteLedger.Version.ToString() + _VoteLedger.Size))));
                }
                Message_Sign Message_Sign = new Message_Sign(_VoteLedger.TxVId);
                var B_TxVId = Message_Sign.BlindMessage();
                var SB_TxVId = Message_Sign.SignBlindedMessage(B_TxVId);
                var UnBlinded_TxVId = Message_Sign.UnblindMessage(SB_TxVId);
                _VoteLedger.Signature = UnBlinded_TxVId.ToByteArray();

#endregion
```

6) After submitting both BL-i and BL-v transactions, Blockchain blocks will be created including the previous block hash and the submitted transactions.

```
private BLv Create_Block(byte[] Tx)
{
    VotersDBContext db = new VotersDBContext();
    VotersDBContext dBContext = new VotersDBContext();
    BLv _BLv = new BLv();
    Persons persons = dBContext.Persons.First(x => x.PersonIdHash == PersonalHash)
    DateTime timestamp = DateTime.Now;
    int Nonce = 1;
    int FinalNonce = 0;
    IList<char> auth_hash = new List<char>();
    string prevHash = GetLatestBLv_Block().ToString();
    for (int i = 0; i < Nonce; i++)
    {
        auth_hash = sha256_hash(Tx + prevHash + "2" + timestamp + +i).ToCharArray(
        Nonce = Nonce + 1;
        IList<char> df = auth_hash.Take(4).ToList();
        foreach (var item in df.ToList())
        {
            if (item.ToString() == "0")
            {
                df.Remove(item);
            }
        }
        if (df.Count() == 0)
        {
            FinalNonce = i;
            Nonce = Nonce - 1;
        }
    }

    string StringHash = "";
    foreach (var item in auth_hash)
    {
        StringHash += item.ToString();
    }
    _BLv.BLv_Prev = Encoding.UTF8.GetBytes(GetLatestBLv_Block());
    _BLv.BLv_Hash = Encoding.UTF8.GetBytes(StringHash);
    _BLv.BLv_Nonce = FinalNonce;
    _BLv.BLv_BlockVersion = 2;
    _BLv.Timestamp = timestamp;
    _BLv.Tx = Tx;
    return _BLv;
}

private string GetLatestBLv_Block()
{
    BlockChainDBContext dBContext = new BlockChainDBContext();
    if (dBContext.BLv.ToList().Count != 0)
    {
        BLv ledger = dBContext.BLv.ToList().OrderByDescending(x => x.BLv_ID).First
        return Encoding.UTF8.GetString(ledger.BLv_Hash);
    }
    else
    {
        return "";
    }
}
```

4.1.4. Tally model

As illustrated in Figure 4.6, every candidate has a summary ballot. Only transactions with a signature from the EAR and not including Pout will be counted. The EARs could verify every single vote within the BL-v Blockchain. Since the system runs over a private network, in case the EAR decided to publish or open the private vote network to the public every one could verify the votes and the eligibility of it.

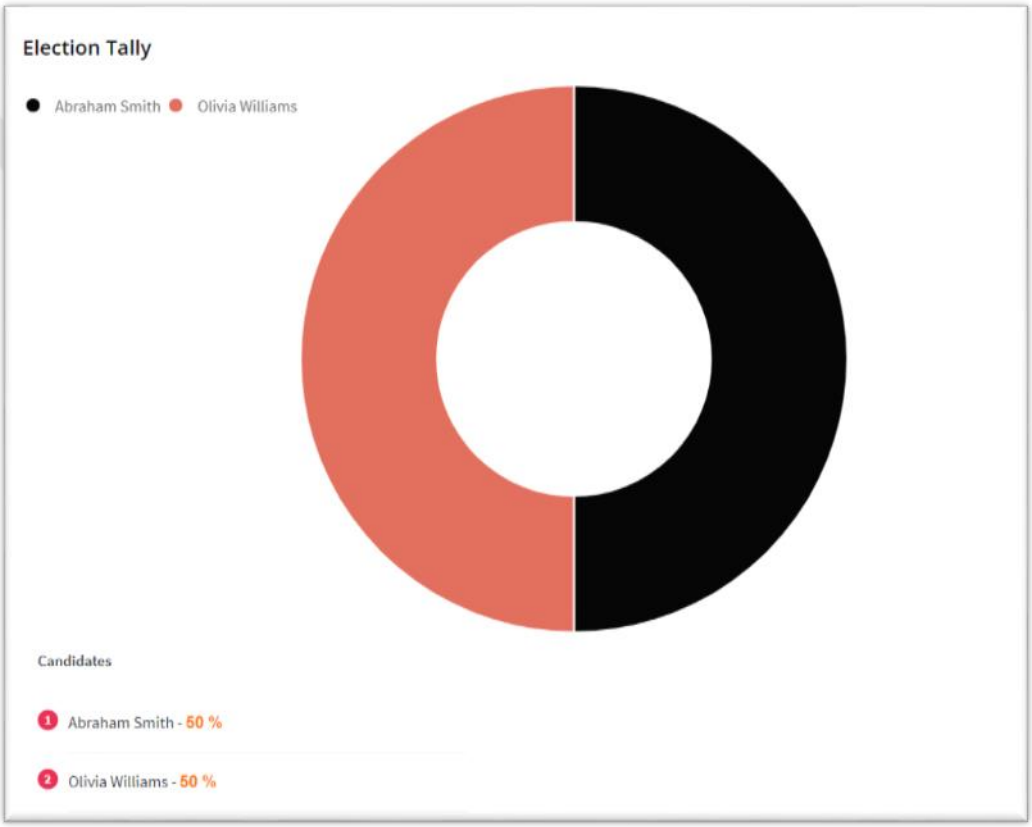

Figure 4.6. Tally model.

5. EVALUATION AND THREAT MODEL

Our proposed current system is structured to defeat many of the vulnerabilities existing in real-world voting systems. Our system meets the basic requirements of fairness and security by knowing the fundamental pillars of building a secure e-Voting system. The implementation of the private Blockchain is in a basic way, and not in the approved manner because the professional implementation exceeds the scope of this master's thesis and requires more resources. No doubt we will highlight and discuss points of potential adversity.

5.1. Expected features

The sufficient properties of the proposed scheme and implementation are listed as follows:

5.1.1. Eligibility

As a first level of determining voter eligibility, the proposed system uses a database containing only eligible voters in the system infrastructure as done in Estonia e-Voting system or to some extent in Germany. The database is controlled by the Electoral Authorities, as such, ineligible voters trying to authenticate themselves to the system in order to establish an eID will not be able to pass the very first stage of the system and based on this, they will be excluded. In case of the main state database, it is extremely reliable; it could be connected to the system instead of our database which contains all the information of the individuals. Thus, age will be determined according to the birthdates on file and excluded any ineligible citizens by the system API which queries the state of the fellow’s eligibility. So the API will return "true" if the person is eligible, and all else will return a "false" value.

The use of eIDs is an essential solution to this system. eIDs are easy to be verifiable and authenticated by EARs. Whereas, in case of any impersonation or disclosing of individual authentication data, the EARs can cancel or suspend the eID. Thus, impersonators will not be able to authenticate themselves or cast ballots.

Furthermore, we will be using the blind signature to authenticate voters who have passed the first eligibility check by providing them with the eligibility signature in

order to allow them to cast an acceptable and countable vote.

5.1.2. Coercion-resistance

Compulsion is one of the most significant problems of both traditional paper-based and electronic elections. Essentially, voters must cast their ballots entirely freely and independently without any external influence. Compulsion not only could be done by force, meanwhile buying votes can be a form of coercion, such as exploiting the poverty of some people by offering amounts of money cannot be rejected to the tyrannical needs. However, the proposed system allows re-voting without displaying any information to voters about their previously cast ballots; this is an excellent strategy which may evade coercion. Voters can cast their votes in the very first minute after connecting to the election, after that, every vote cast will be ineffective. Thus the vote buyer will not have any guarantee that the voter had voted or not in earlier time or not, which could limit the vote-buying process. The previously proposed systems procedures or a real-world system (i.e., Estonian's system procedure) allows re-voting with the exception that the final ballot will be counted, but this does not limit the coercion process because the opportunity will be available for the corrupted candidates to compel people during the electoral period. On the other hand, a high-level attacker will have time to hack into voters' accounts or devices and overwrite their votes. For instance, attackers could create malware which could read the individual's PIN to allow hackers to overwrite votes in the Estonian e-Voting system (Springall et al., 2014).

5.1.3. Availability

To the best of our knowledge, so far, all-digital voting systems used in the real-world are concentrated in one data center except for Sierra Leone. Sierra Leone became the first country in history to use Blockchain as a base for its voting system.

Our protocol runs over a network in a distributed nature where it transfers information securely and the operation executed in a decentralized manner across a network of nodes. Blockchain technology’s distributed nature provides a tremendous resistance to Distributed Denial of Service (DDoS) attacks and other network abuses

(URL-2). Thus, sufficient availability is gained by the redundancy of the servers. Since each node holds a replica, there will be no need for full reliance on a single node. So, if one node goes down, the network will continue its work as normal. Hence, the risk associated with a DDOS attack is drastically reduced.

As a summary, distributing the servers into multiple data centers could achieve high availability and prevent several kinds of related attacks.

5.1.4. Ballot anonymity and election secrecy

In order to obtain a possibility for authentication of the ballots by the Electoral Authorities in the tally processes while preserving the privacy of the voter, we used the blind signature in our protocol. Thus each vote will be certified by the electoral authorities in order to be accepted for the tally. Also, the protocol allows the electoral authorities the chance to check the credentials of the voter to ensure that they are eligible to cast a vote without violating the privacy of the ballots. The unlinkable blind signature provides the EARs the ability to see the contents of any ballot they sign while simultaneously not being able to link the blinded ballots back to open ballots received during the tally process.

On the other hand, the use of double private Blockchain to separate the vote from the voter identity achieves full anonymity for the ballot. In general, the votes will be encrypted by voter and then blindly signed, after which they are encrypted using the candidate's public key. This ensures the anonymity of the vote and the voter during the whole election process. Also, the Blockchain address is generated randomly, so voters and the address are unlinkable for the system or even an outside observer.

5.1.5. Integrity

Blockchain has intentionally been created to provide infinite immutability, i.e., in theory at least nobody can alter Blockchain's 'distributed ledger' of all obligated blocks. Integrity is one of the most excited properties which guarantee the integrity of the whole process. Since the ballots are encrypted with a blind signature in this scheme, ballots will not be modifiable all the way to the block in each Blockchain without the private keys.

Besides, accepting the first vote cast will thus not allow overwriting by voters themselves or by attacker who have gained an undeserved privilege to access voters' accounts in any way.

5.1.6. Uniqueness

A voter can establish no more than one eID and vote multiple times, whereas only first vote cast will be counted in the tally. Voters will be free to vote as many times as they want in order to manipulate vote buyers by creating unreliability between buyers and voters. Accordingly, in the tallying phase if the previous transaction Out (POut) has a value that transaction will be ignored.

5.1.7. Robustness

Since citizens could cooperate with corrupted candidates, their credibility had to be destroyed to force them to vote in the right way (see 5.1.2). Due to the address anonymity and the vote anonymity that is created by using the second Blockchain, it will not be evident that the vote vender manipulates the corrupted candidate. Also, it is impossible to alter the final voting result after result tally.

5.1.8. Fairness

As explained before, if results are announced before the end of the election, voters could be influenced by these results. They would reconsider their choice because of the published results or they may make a decision based on the majority vote and not on complete contentment. Voters must not be manipulated through preliminary results or any opinion poll (Blais et al., 2001), they must vote for any party/candidate they wholeheartedly believe in. Publishing opinion polls during the elections is a strong strategy to influence voters' choices. In 2014, during the European Parliament elections, the European Commission warned countries not to publish any early results until all states had tallied the ballots. They made such a warning in order to slough all the pressure from voters, to allow them to vote without any influence (URL-8).

Our protocol follows the standard practices which forbid the early publishing of any results. This protocol waits for the end of the election event, after which it shuffles

the ballots and then decrypts them.

5.2. Threat Model

Maintaining sufficient security of the e-Voting paradigm requires overcoming many difficult problems in cybersecurity, primarily with existing technology. The smallest lapse can affect the integrity of election results and the democratic process. Undesirable experiences in various places around the world (New South Wales, Estonia, India, etc.) in real-world applications have come up in online voting examples, demonstrating many security issues. Many kinds of attackers (i.e., foreign countries, deceptive voters, funded criminals, etc.) displayed realistic threats, many of these threats must be taken as example while designing or implementing online voting. Let us initially identify some potential attacks and any exposures that might appear, subsequently highlight the way to avoid and overcome them.

5.2.1. DDoS attacks

Distributed Denial of Service attack is a type of malicious attack by an attacker or a group of attackers to put a computer, computers, or network resources out of service for a limited period or permanently. Attackers typically flood requests faster than a server can respond or send requests specifically designed to consume the resources of the target devices, so the server can no longer respond to even legitimate requests. Besides, every node could be implemented with a Byzantine Fault Tolerance algorithm (Reid and Harrigan, 2013), that helps to detect the failed nodes in the network. So the network maintainer will immediately locate any failure occurring in any node.

5.2.2. Authentication vulnerability

In our design, individuals are identified / authenticated by the system via their electronic IDs and the corresponding verification code sent to their cell phones. The time-based, one-time password is used to ensure voters themselves are logging in to the system or at least it is being done with their knowledge, using their cell phone. In case attackers are able to obtain individuals' data and gain access to their personal cell phone, they would be able to cast a ballot as they prefer. Overcoming this

vulnerability requires implementation of a biometric scan algorithm.

5.2.3. Sybil attack

Sybil attack is a dangerous threat where a node in a network claims multiple identities while it is unknown to the network, that the nodes are controlled by the same adversarial entity. For instance, an adversary can spawn up multiple computers, virtual machines, and IP addresses. They can then establish various accounts with different usernames and pretend that all of these exist in different regions. In centralized systems, for instance, a centralized entity may try to avoid Sybil attacks by forcing an individual IP to not create more than a specific number of user accounts in a given time interval.

Since our proposed system is running in a permissioned network, unauthorized people are not allowed to create an account. Permissioned Blockchains provide significant resistance against today's security problems by using robust cryptography features and limited access to the ledger, without affecting the transparency aspect of Blockchain technology.

5.2.4. Cross-site scripting (XSS)

XSS vulnerabilities are a serious gap in security represented by modifying the content of the site pages that appear to the user by injecting HTML or JavaScript, exactly just like modifying HTML files and JavaScript through a web page editor. It is a type of injection attack whereby an attacker who can exploit an XSS attack could basically gain the ability to act like the victim. Moreover, both voters and the vulnerable system will often not be aware of the attack. Using all of the well-known practices outlined below is considered a great way to defeat the majority of XSS vulnerabilities (Vonnegut, 2017). See Cross-Site Scripting Attack in Figure 5.1.

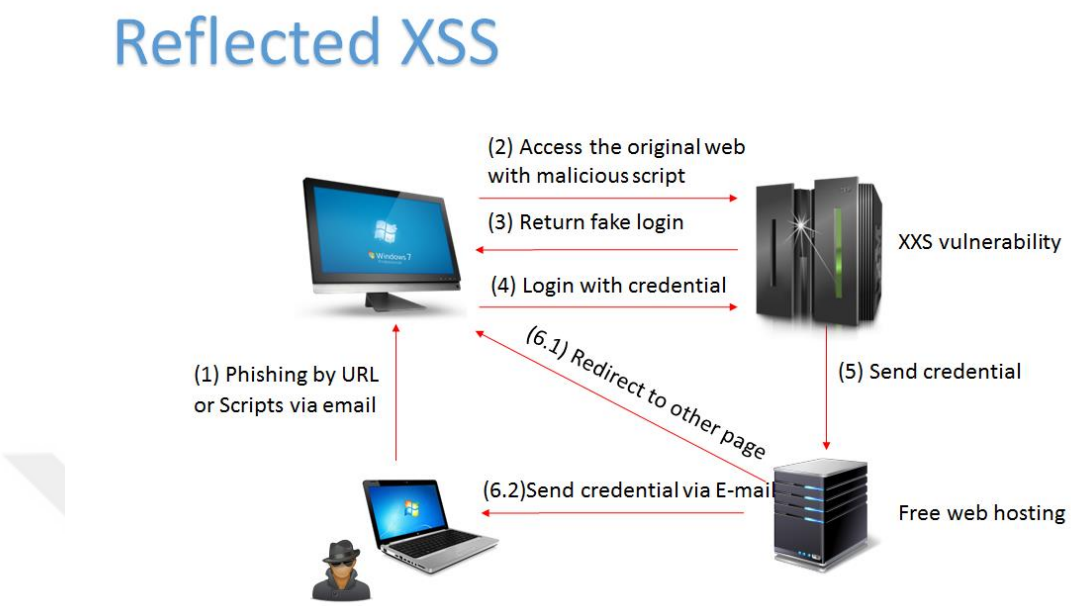

Figure 5.1. Cross-Site Scripting Attack.

Escaping. Escaping data means ensuring that the data is secure before rendering it to the end user. So escaping user input and key characters prevent received data from being interpreted in any malicious way.

Validating Input. Input validation prevents voters from inserting any special characters, instead of rejecting the request.

Sanitizing. Cleaning the data from any potentially harmful mark-ups, moreover, changing untrusted user inputs to an acceptable format.

5.2.5. Freak attack

The SSL/TLS vulnerability (Factoring Attack on RSA-Export Keys) is a devastating security gap which allows attackers to decrypt secure communications between vulnerable clients and servers (intercepting HTTPS connections) and forces them to use older and weaker encryption, also known as the export-grade key or 512-bit RSA keys (Vonnegut, 2017). However, this is a flaw in the encrypted traffic of millions of sites nowadays. Previously, cryptographers including the NSA.gov website had believed to be safe, but researchers found the seriousness of the security breach a few

years ago. More than a quarter of sites that previously believed to be safe are vulnerable to this gap, as the Freak vulnerability affects the OpenSSL protocol - used in Android - and TLS / SSL - used in the Apple Safari browser.

The US government had a ban on the use of strong encryption systems in the 1990s, and this is the reason this serious gap is evident in such a high quantity of websites. Although restrictions on encryption systems (512-bit) were revoked in the late 1990s, many programs used today are still affected by the results of that period.

Let's highlight some necessary precautions against the Freak attack by:

1) Disable the support for TLS export cipher suites
2) Disable the support for insecure known ciphers (not only RSA export ciphers)
3) Disable support for ciphers with 40- and 56-bit encryption
4) Enable forward secrecy

5.2.6. Malware

Malicious attackers who tend to manipulate systems to have more access by promoting their privileges can install viruses, worms, Trojans, rootkits, spyware, key-loggers, backdoors, and many other types by using pre-existing botnets and target a specific country or region. 32% of computers around the world are infected with viruses and malware, making it easy for attackers to control or hack the infected voters' computers (URL-1).

Let us demonstrate some of the ways that malware could distort the elections as follows:

1) Malware might alter voters’ selections
2) Malware might record the vote casting processes for the favor of a third party
3) Malware read the passwords used by the individuals and copy them

Voters must be cautious and familiar with the computer protection instructions to be able to protect their own computers. Some of the standard critical steps to protect voters’ computers are highlighted below:

1) Installing a firewall
2) Installing security software from a reliable company
3) Setting the operating system and the web browser to update automatically
4) Making sure that the web browser's security setting is high enough to detect unauthorized downloads

6. CONCLUSIONS

6.1. Summary

This research has fundamentally explored the basic concept of e-Voting and the use of permissioned double Blockchains. Furthermore, the redesign of a new block for the vote asset based on the Bitcoin address algorithm concept was explored. The protocol was presented with six individual stages and described explicitly along with its mechanism. The protocol has partially been developed to prove our point of view. Finally, this research evaluated the protocol and potential security threats for the protocol, further recommendation and limitations have been debated in the end.

6.2. Limitations

This system proposed is based on the assumption of voters using a secure device to cast votes. Even though our scheme is secure, state-level attackers are known to employ firmware-based malware already installed or going to be installed on the voters' devices could help these attackers cast or alter votes. Our system was designed to accept the first vote only, based on what applied in traditional paper-based elections, which may be considered a drawback in our design, but is in our opinion an advantage because it prevents many attempts of vote buying. In addition to this, it does not allow any ballot to be overwritten by an attacker or the same person under the influence of an external force. Finally, the voter is prevented from being manipulated by opinion polls or potential result leaks.

6.3. Discussion

This research proposes a novel e-Voting structure to properly facilitate digitalized elections, maintaining vote and voter privacy while preventing fraud. The key processes of the structure, namely, voter ID creation, voting, vote verification and vote tallying are introduced in the logical design of the digital voting system. The key technologies used in affecting such results are the distributed ledger of Bitcoin, namely the Blockchain technology, and RSA public-private key system. Moreover, finding some solutions to overcoming many server- and client-side attacks faced by earlier elections were indicated.

Fundamentally, Blockchain technology's ledger decentralization and blocks serialization provide great tools against results manipulation. So, using the proposed Blockchain based architecture leads to achieve sound and fair election. Let's not forget the issue of impersonation. By deliberately opening the system sometime before the election, voters will be allowed to establish their identity. If there is an impersonation, it will be discovered. Also, eID's legitimate owner only will be able to use it due to the time-critical one-time password algorithm.

Furthermore, the practical approach proposed to address the issue of vote-selling, that is by making vote vendors unreliable is one of the most significant solutions of this structure. Several ways and possibilities may allow a candidate/party to buy voters' ballots. Firstly, by taking all the login information of the voter and loging into the system in his place during the polling day and casting a vote. Secondly, asking a voter to attend a specific place and make him cast his vote for them remotely. Thirdly, a voter can be asked to video himself during the voting process. Our proposed approach involves allowing voters to cast a vote unlimited times, but only the first vote cast will be accepted because the structure of the Blockchain in itself prevents double voting.

It should be noted that the proposed scheme permits absentee ballot due to its web-based nature allowing remote vote casting; however mail-in ballot (postal vote) and proxy voting are not permitted. The identified approach involves trade-offs and may not be suitable for all cases. Some citizens, especially older ones, may not be able to cope with the complexity of digital voting: eID creation, PKI use, or even using a browser. It can be solved by posting guidance videos or preparing help centres to guide voters who have difficulty following the voting procedures. As a stop-gap measure, postal and proxy vote may be authorized in advance in certain exceptional cases, such as inability and incapacity to reach the polling station.

Eventually, after we introduced our e-voting protocol we can answer the research questions in section (1.5) of Introduction chapter as below:

1. In our protocol voters are able to cast their vote facilitated and secure way and

candidates receives the votes without interfering. So it is possible to have an e-voting that satisfies all parties by employing the disintermediation, immutability, and decentralization of Blockchain.

2. As we introduced in section (1.3) of Introduction chapter that several real-world application and proposals for e-voting system have been practiced over a wide range, some of them are used so far and others have cancelled due to security problems that have been encountered. Some of these systems used public key encryption as a major encryption method of protection, and some used the public key encryption with private Blockchain, but the combination of public key encryption and Blind signature with the permissioned Blockchain is a new way to solve many of the problems in previous systems. Also, we may have violated some of the previous protocols for logical reasons (such as accepting the first cast vote).
3. Combination of public key encryption and Blind signature with Blockchain prevents impersonators from impersonating others and also prevents violation of voters' privacy. Besides, to the distribution of the ledgers provides great resistance to fraud and manipulation of entered data.
4. Besides the infinite immutability and high availability of Blockchain, the technology provides an audit trail, distribution, robustness, and disintermediation. All those properties make the system more secure and sufficient. Thus, it is better to use Blockchain instead of regular databases.

6.4. Accomplishments

In this thesis, we discussed the problems and concerns which have always been obstacles to the implementation of e-Voting, and we put forward many solutions to overcome these, starting with the use of the Blockchain technique to protect immutability of voting result as well as the use of public and private keys to protect ballot / ballotter privacy. Besides this, the use of the blind signature grants vote eligibility only to those who have the right to vote without infringing on the privacy of their ballots.

As a summary, in reference to the research questions posed at the outset and satisfactorily addressing them, what we have achieved in this thesis specifically are as follows:

1. Facilitate the voting process for citizens by making it available online
2. Reduce much cost spent on ballot papers, polling stations, etc.
3. Examining several potential attacks and some attacks faced by previous experiments to propose solutions to overcome them
4. Designing a new transaction used in vote casting based on Bitcoin's transaction cryptography standards
5. Providing secure communications through the system between all election parties(voter, trustee, and candidates)
6. Prevent any attempt to violate ballot/ballotter privacy
7. Allow only eligible voters to cast ballots
8. Obtain the essential fairness for a democratic election
9. Prevent vote sale/buying attempts

6.5. Recommendations

Some of the recommendations by the researchers through their experiences in these specific fields of research and software development are as follows:

1) As usual, the current Web-based system uses Secure Hyper Text Transfer Protocol (HTTPS), yet there is a need for a further configuration affected by the system administrators to secure the connection between the server and the clients' computers, and thus preventing Logjam and Freak attacks. By disabling the

support for TLS (Transport Layer Security) export cipher suites and using a 2048-bit Diffie-Hellman group, also disables other cipher suites that are known to be insecure and enable forward secrecy to obtain the required server immunity (Adrian et al., 2015; Durumeric, Adrian, Mirian, Bailey, and Halderman, 2015).

2) A practical recommendation helping to design a functional Blockchain voting system is to implement a shift in Domain Modelling as well as the following:
   a) Business Object Notation (BON),
   b) Extended Business Object Notation (EBON),
   c) Implementation-independent high-/medium-level specification,
   d) Implementation-independent medium-level specification.
3) Static analysis tools "process the source code and specifications of the system to provide information about the system without executing the code."
4) Dynamic analysis tools monitor a running system measuring the aspects of its operation and detect undesirable behavior. Researchers also point out that dynamic analysis tools can also be used to identify the following issues:

a) Issues related to memory (leaks, corruption),
b) Concurrency (deadlock, spinning, data races),
c) Resource allocation (unclosed sockets and files),
d) Security (buffer overflows and other vulnerabilities). Some of these issues are mitigated through the various language methods.

5) Another important suggestion would be model checking, which would comply with all the specifications. While recommending against centralized version control systems such as Subversion and CVS, for these affect working especially for offline workers, tracking tools must be used extensively.
6) Most importantly, the awareness to be spread among social and political domains is crucial to the implementation of e-Voting systems. The complete implementation shall never be compromised or the system altered, as this could pose a threat of fraudulent activities in the election process.

REFERENCES

Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J. A., Valenta, L., 2015. Imperfect forward secrecy: How Diffie-Hellman fails in practice, 22nd ACM SIGSAC Conference on Computer and Communications Security, USA, 5-17.

Ayed, B. A., 2017. A conceptual secure blockchain-based electronic voting System, International Journal of Network Security and Its Applications (IJNSA), 9, 3, 1-9.

Bartolucci, S., Bernat, P., and Joseph, D., 2018. SHARVOT: secret SHARe-based voting on the blockchain, 1st International Workshop on Emerging Trends in Software Engineering for Blockchain, Gothenburg, 30-34.

Baudron, O., Fouque, P.-A., Pointcheval, D., Stern, J., and Poupard, G., 2001. Practical multi-candidate election system, 20th Annual ACM Symposium on Principles of Distributed Computing, Chicago, 274-283.

Bellovin, R., 2009. Cryptography: Authentication, Blind signatures, and digital cash.

Benaloh, J., and Tuinstra, D., 1994. Receipt-free secret-ballot elections, 26th Annual ACM symposium on theory of computing, Montreal, 544-553.

Blais, A., Gidengil, E., and Nevitte, N., 2001. Do polls influence the Vote, 263-279.

Brightwell, I., Cucurull, J., Galindo, D., and Guasch, S., 2015. An overview of the iVote 2015 voting system, NSW Electoral Commission.

CasaDo-VaRa, R., and CoRCHaDo, J. M., 2018. Blockchain for democratic voting: how blockchain could cast of voter fraud, Oriental Journal of Computer Science and Technology, 11, 1, 01-03.

Cetinkaya, O., and Cetinkaya, D., 2007. Towards secure e-elections in Turkey: requirements and principles, International Dependability Conference - Bridging Theory and Practice, Vienna, 903-907.

Chaum, D. L., 1981. Untraceable electronic mail, return addresses and digital pseudonyms, technical note programming techniques and data structures, Advances in Information Security, 7, 211-219.

Cohen, J. D., and Fischer, M. J., 1985. A robust and verifiable cryptographically secure election scheme, 26th Annual Symposium on Foundations of Computer Science, Portland, USA, 372-382.

Constantinescu, N., 2010. Elliptic curve cryptosystems and scalar multiplication, Annals of the university of Craiova-Mathematics and Computer Science Series, 37, 1, 27-34.

Cranor, L. F., and Cytron, R. K., 1997. Sensus: A security-conscious electronic polling system for the internet, System Sciences, Proceedings of the Thirtieth Hawaii International Conference, 561-570.

Crosby, M., Nachiappan, Pattanayak, P., Verma, S., and Kalyanaraman, V., 2015.

BlockChain technology beyond bitcoin.

Czepluch, J. S., Lollike, N. Z., and Malone, S. O., 2015. The use of block chain technology in different application domains, IT University of Copenhagen, Copenhagen.

DeMillo, R. A., Lynch, N. A., and Merritt, M. J., 1982. Cryptographic protocols. 14th Annual ACM symposium on theory of computing, California, 383-400.

Dunphy, P., and Petitcolas, F. A., 2018. A first look at identity management schemes on the blockchain, IEEE, 16, 4, 20-29.

Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J., 2015. Tracking the FREAK attack.

Elçi, A., 2018. Unavoidable Rise of Blockchain. Yet, how far, 11th International conference on security of information and networks (SIN 2018), Cardiff.

ElGamal, T., 1985. A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory, 31, 4, 469-472.

Evans, D., and Paul, N., 2004. Election Security: Perception and reality, IEEE Security and Privacy, 2, 1, 24-31.

Eyal, I., and Sirer, E. G., 2018. Majority is not enough: Bitcoin mining is vulnerable, ACM, 61, 7, 95-102.

Freeman, A., 2014. Pro AsP. Net MVC 5 platform. In pro ASP.NET MVC 5 Platform, Springer, 3-8.

Fujioka, A., Okamoto, T., and Ohta, K. A., 1992. A practical secret voting scheme for large scale elections, 11th International conference on Frontier of computer science and technology, Brisbane, 514-519.

Gaetani, E., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., and Sassone, V., 2017. Blockchain-based database to ensure data integrity in cloud computing environments, First Italian Conference on Cybersecurity (ITASEC17), Venice, 1816, 146-155.

Gibson, R., 2013. Elections Online: Assessing internet voting in light of the Arizona democratic primary political science quarterly, The journal of public and international affairs, 116, 4, 561-583.

Glaser, F., 2017. Pervasive decentralisation of digital infrastructures: A framework for blockchain enabled system and use case analysis, 50th Hawaii International Conference on System Sciences (HICSS), Hawaii.

Halderman, J. A., and Teague, V., 2015. The New South Wales iVote system: Security Failures and Verification Flaws in a Live Online Election, International Conference on E-Voting and Identity, Bern, 9269, 35-53.

Hardwick, F. S., Akram, R. N., and Markantonakis, K., 2018. E-voting with blockchain: An e-voting protocol with decentralisation and voter privacy.

Hirt, M., and Sako, K., 2000. Efficient receipt-free voting based on homomorphic

encryption, International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, 1807, 539-556.

Hjalmarsson, F. P., Hreioarsson, G. K., Hamdaqa, M., and Hjalmtysson, G., 2018. Blockchain-based e-voting system, 11th International Conference on Cloud Computing (CLOUD), IEEE, 983-986.

Ibrahim, S. Kamat M. Salleh, M., and Aziz, S. R. A., 2003. secure e-voting with blind signature, 4th National Conference of Communication Technology, Shah Alam, Malaysia.

Jan, J., Chen, Y., and Lin, Y., 2001. The design of protocol for e-voting on the Internet, 35th Annual 2001 International Carnahan Conference on Security Technology, London.

Jason, P. C., and Yuichi, K., 2017. E-voting system based on the bitcoin protocol and blind signatures, E-voting system based on the bitcoin protocol and blind signatures , 10, 1, 14-22.

Jonker, H., Mauw, S., and Pang, J., 2013. Privacy and verifiability in voting systems: Methods, developments and trends, In Computer Science Review, 10, 1-30.

Juang, W.-S., and C.-L., L., 1997. A secure and practical electronic voting scheme for real world environments, IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences.

Juels, A., Catalano, D., and Jakobsson, M., 2010. Coercion-resistant electronic elections, In Towards Trustworthy Elections, 37-63.

Kizhakkedathil, N., 2016. A study into the prospects of implementing end-to-end verifiability in Estonia voting, Tallinn University of Technology, Estonia.

Lars, H. N., and Hole, K., 2012. Building and maintaining trust in internet voting. IEEE, 45, 5, 74 - 80.

Liu, Y., and Wang, Q., 2017. An e-voting protocol based on blockchain, IACR Cryptol. ePrint Arch., Santa Barbara, 1043, 2017.

Lopez, J., Oppliger, R., and Pernul, G., 2005. Classifying public key certificates, European Public Key Infrastructure Workshop, 3545, 135-143.

Lyon, D., 2009. National IDs in a global world: surveillance, security, and citizenship, Case W. Res. J. Int'l L., 42, 607.

M'Raihi, D., Machani, S., Pei, M., and Rydell, J., 2011. Totp: Time-based one-time password algorithm, 6238, 1- 16.

Moser, M., 2017. Anonymity of bitcoin transactions, Hawaii International Conference on System Sciences, University of Münster, University of Münster, Germany.

Nair, G. R., and Sebastian, S., 2017. Blockchain technology centralised ledger to distributed ledger, International Research Journal of Engineering and Technology (IRJET).

Nakamoto, S., 2009. Bitcoin: A peer-to-peer electronic cash system.

Ohkubo, M., Miura, F., Abe, M., Fujioka, A., and Okamoto, T., 1999. An Improvement on a Practical Secret Voting Scheme. Paper presented at the International Workshop on Information Security, Kuala Lumpur, Malaysia.

Pran, V., and Merloe, P., 2007. NDI handbook: Monitoring electronic technologies in electoral processes, Washington.

Preneel, B., Bosselaers, A., and Dobbertin, H., 1997, The cryptographic hash function RIPEMD-160. Citeseer.

Quisquater, J.-J., Quisquater, M., Quisquater, M., Quisquater, M., Guillou, L., Guillou, M. A., Guillou, S., 1989. How to explain zero-knowledge protocols to your children, Conference on the Theory and Application of Cryptology, 435, 628-631.

Reid, F., and Harrigan, M., 2013. An analysis of anonymity in the bitcoin system. In Security and privacy in social networks (pp. 197-223): Springer.

Rivest, R. L., Shamir, A., and Adleman, L., 1978. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21, 2, 120-126.

Schläpfer, M., Haenni, R., Koenig, R., and Spycher, O., 2011. Efficient vote authorization in coercion-resistant internet voting. International Conference on E-Voting and Identity, 7187, 71-88.

Schnorr, C. P., and Jakobsson, M., 2000. Security of signed ElGamal encryption. International Conference on the Theory and Application of Cryptology and Information Security, 1976, 73-89,

Segaard, B., Christensen, D. A., Folkestad, B., and Saglie, J., 2014. Internettvalg Hva Gjør Og Mener Velgerne: Institutt for samfunnsforskning.

Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., and Halderman, J. J., 2014. Security analysis of the estonian internet voting system. The 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, 703-715.

TRUEB, B. A., 2013. Estonian Electronic ID–card application specification Prerequisites to the Smart Card Differentiation to previous versions of EstEID card application.

DDWdwDWWWVon Ahn, L., Maurer, B., McMillen, C., Abraham, D., and Blum, M., 2008. recaptcha: Human-based character recognition via web security measures. Science, 321, 5895, 1465-1468.

Vonnegut, S., 2017. Preventing XSS: 3 Ways to Keep Cross-Site Scripting Out of Your Apps.

Wolchok, S., Wustrow, E., Isabel, D., and Halderman, J. A., 2012. Attacking the Washington, DC internet voting system. International Conference on Financial Cryptography and Data Security.

Wu, Y., 2017. An E-voting System based on Blockchain and Ring Signature. University of Birmingham.

Zhao, Z., and Chan, T.-H. H., 2015. How to vote privately using bitcoin. International Conference on Information and Communications Security, Beijing, 954, 82-96.

Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H., 2017. An overview of blockchain technology: Architecture, consensus, and future trends. Big Data (BigData Congress) for IEEE International Congress, Honolulu, 557-564.

Zheng, Z., Xie, S., Dai, H., Chen, X., and Wang, H., 2017. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. International Congress on Big Data, BigData Congress 2017, Honolulu, 557-564.

URL-1 < http://dottech.org/96627/32-of-computers-around-the-world-are-infected-with-viruses-andVersion:February>, Date of access: 29.07.2018.

URL-2 <https://www.techopedia.com/will-Blockchain-technology-make-ddos-attacks-obsolete/2/33115> , Date of access: 07.08.2018.

URL-3 <https://www.pluralsight.com/courses/cross-site-forgery-request-web-app?gclid=CjwKCAjw3qDeBRBkEiwAsqeO7r8BBJHi9zeiQNXyGwCh-fpEMBwMObZwI93367KDhhgUg8JCLbEETRoCLLsQAvD_BwEandaid=7010a000002BWqGAAWandpromo=andoid=andutm_source=non_brandedandutm_medium=digital_paid_search_googleandutm_campaign=EMEA_Dynamicandutm_content=ands_kwcid=AL!5668!3!277727473382!b!!g!!andef_id=W5wJFQAABtekejx0:20181018120845:s>, Date of access: 07.08.2018.

URL-4 <https://www.valimised.ee/en/archive/statistics-about-internet-voting-estonia>, Date of access: 17.10.2018.

URL-5 <https://www.regjeringen.no/en/aktuelt/Internet-voting-pilot-to-be-discontinued/id764300/>, Date of access: 02.05.2018.

URL-6 <https://medium.com/iryo-network/public-vs-permissioned-private-Blockchains-99c04eb722e5>, Date of access: 27.07.2018.

URL-7 <http://shiflett.org/articles/cross-site-scripting>, Date of access: 20.10.2018.

URL-8, <http://www.politico.eu/article/commission-prohibits-early-publishing-of-election-results>, Date of access: 10.09.2018

CURRICULUM VITAE

Name and Surname : Mahmoud Al-Rawy

Address : ESAT CD. NO: 132/3 MERKEZ CANKAYA ANKARA TURKEY

Email : mahmoud.alrawie@gmail.com

EDUCATION

Bachelor : Al-Rafidain University, 2008-2012

Master : Aksaray University, 2016-2018

PROFESSIONAL EXPERIENCE AND AWARDS

1) Senior Software Developer Team Lead (Turkey).
2) Senior Software Developer, Hadetha Co. for software and automation (Iraq).
3) Software Developer Al-Wasama Co. (Iraq).
4) IT –Junior Software Developer, Anbar University (Iraq).

PUBLICATIONS, PRESENTATIONS AND PATENTS

1) "Al-Rawy, M., and Elci, A. (2018). A Design for Blockchain-Based Digital Voting System", at the 2018 International Conference on Digital Science (DSIC'18), where it was awarded the Best Paper Certificate.