A Bitcoin Based Secure Electronic Voting System

by

DMGK Wimalarathne (168278C)

A thesis submitted to University of Moratuwa in partial fulfilment of the requirements for the

Master of Computer Science, *Specialized in Security Engineering*

Department of Computer Science & Engineering
University of Moratuwa, Sri Lanka

February 2020

A Bitcoin Based Secure Electronic Voting System

by

DMGK Wimalarathne (168278C)

A thesis submitted to University of Moratuwa in partial fulfilment of the requirements for the
Master of Computer Science, *Specialized in Security Engineering*

Department of Computer Science & Engineering
University of Moratuwa, Sri Lanka

February 2020

Declaration

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to University of Moratuwa the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

DMGK Wimalarathne: Date

Approved by:

Lt Col Dr Chandana D. Gamage Date
Department of Computer Science and Engineering
University of Moratuwa

Copyright Statement

I hereby grant the University of Moratuwa the right to archive and to make available my thesis or dissertation in whole or part in the University Libraries in all forms of media, subject to the provisions of the current copyright act of Sri Lanka. I retrain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

DMGK Wimalarathne: Date

I have supervised and accepted this thesis/dissertation for the award of the degree.

Lt Col Dr Chandana D. Gamage Date
Department of Computer Science and Engineering
University of Moratuwa

Abstract

Over the last few decades, several electronic systems have been proposed and implemented to as attempt to replace the traditional paper-based voting systems. Even though the e-voting system are more efficient and convenient than the traditional voting systems, it was identified that they should meet the specific security goals, such as authentication, anonymity, availability, and integrity up to the same level that is provided by manual systems.

If the voting system is centralized and controlled by one party, they may have the opportunity to manipulate the votes thereby compromise the integrity. In this paper we propose a Bitcoin based online transaction system to provide a solution to the identified integrity related threats in an electronic voting system.

We have taken an existing, well-proven, robust, scalable e-cash system as the basis for implementing the e-voting system. A comprehensive list of properties and features expected of an e-cash system and e-voting system have been analysed in the paper to show how different properties/features of an e-voting system map to an e-cash system. We have shown how various functionalities of a bitcoin-like system directly provide the required features/properties of an e-voting system. Also, we have shown how various functionalities of a bitcoin-like system can be modified and/or adapted to provide some of the other required features/properties of an e-voting system.

Based on the outcomes of the methodology, we discuss how the complete e-voting system is going to be built on blockchain technology. Further, we discuss how strongly various security and performance requirements are being met in the research work related to the proposed e-voting system.

Acknowledgements

I would like to express profound gratitude to my supervisor, Dr. Chandana Gamage, for his invaluable support by providing relevant knowledge, materials, advice, supervision, and useful suggestions throughout this research work. His expertise and continuous guidance enabled me to complete my work successfully.

I am grateful for the support and advice given by the CSE Lecturer panel and the MSc course coordinators, by encouraging continuing this research till the end. Further I would like to thank all my colleagues for their help on finding relevant research material, sharing knowledge and experience and for their encouragement.

I am as ever, especially indebted to my parents and sister for their love and support throughout my life.

Abbreviations

ATM - Automated Teller Machine

BIP - Bitcoin Improvement Proposal

CPU - Central Processing Unit

DRE - Direct Recording Electronic

DVBM - Digital Vote-by-Mail

E2E - End-to-end

ECC - Elliptic Curve Cryptography

ECDSA - Elliptic Curve Digital Signature Algorithm

NFC - Near-field communication

P2PKH - Pay-To-Public-Key-Hash

PIN - Personal Identification Number

PKI - Public Key Infrastructure

PRNG - Pseudo Random Number Generator

QR code - Quick Response code

SHA - Secure Hash Algorithm

TLS - Transport Layer Security

URI - Uniform Resource Identifier

VVPAT - Voter Verified Paper Audit Trail

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

It's not the voting that's democracy, it's the counting. -Tom Stoppard, 1972

Preserving the integrity of an election has become a fundamental requirement as it preserves the democracy itself, which has also been highlighted in previous studies [1], [2]. Voting technologies including paper ballots, mechanical lever systems, punched cards, optical mark-sense, and DRE voting systems are designed and used to conduct such elections [3], [4]. Since the elections are held to allow voters to choose a preferred representative/s from a list of candidates, a voting system should be tamper-resistant, transparent, and comprehensible up to a sufficient level, to both voters and candidates.

1.1. Background

Paper based voting systems, pioneered by 'Australian ballot' has been the standard voting system since the 1980's, which is a simple, auditable scheme that ensures the privacy of the voters. With the technological advancements, electronic voting systems, often referred as DRE systems are becoming popular as it reduces the time and cost associated with counting the votes in large numbers. However, 'trusting' those systems has become a problem the voters themselves cannot be certain of the integrity of the system.

In his book, *Applied Cryptography*, Bruce Schneier has suggested six requirements of computerized voting with one additional requirement that both maintains individual privacy and prevents cheating [3]:

1) Only authorized voters can vote.
2) No one can vote more than once.
3) No one can determine for whom anyone else voted.
4) No one can duplicate anyone else's vote.
5) No one can change anyone else's vote without being discovered.
6) Every voter can make sure that his vote has been taken into account in the final tabulation.
7) Everyone knows who voted and who didn't.

Verification of current electronic voting systems against above requirements has become harder because of their centralized nature and the dependency on various hardware and software platforms.

Also in a recent report [4], Caltech/MIT Voting Technology Project has recommended "Continued strong support for voting systems security research is critical, emphasizing auditing and the verifiability of election outcomes."

Therefore, what we propose here is a Bitcoin based novel electronic voting system that will fulfil the requirements a computerized voting system up to an acceptable level. The promising decentralized cryptocurrency, Bitcoin [5], has proved that Bitcoin based systems are practically usable for a system that requires verifiable security.

1.2. Research Problem

Electronic or online based voting systems are becoming popular. With the evolution of the technology in this Information Age, it is apparent that the voting systems will become fully electronic or online based in the near future. In fact, some countries have already carried out pilot runs and even implemented the Internet voting nationally [6], [7].

Voters cannot make sure that their vote has been recorded properly and taken into account. In the electronic voting systems, even though the user has made the correct choice and obtained a receipt, without observing the software code, he/she cannot ensure that the vote has been recorded successfully. Some DREs are equipped with Voter Verified Paper Audit Trail (VVPAT) [8] printers. This printer allows the voter to confirm their selections on a paper record before storing the votes in the memory of the computer. This paper record can be used for auditing or recounting. However, this process is similar to a paper-based voting system, which take time and manpower.

Security is a fundamental requirement. It is concluded in the in the paper [7] *Attacking the Washington, D.C. Internet Voting System*, that a minor misconfiguration or incorrect implementation in the voting system infrastructure including the network or the centralized servers can make the entire election inaccurate and illegitimate. During this study, it was possible for the authors get the almost full control of the servers used for the election, allowing them to change the votes and to reveal most of the secret ballots, validating their statement.

1.3. Objective

The objectives of this research are to gather requirements of an electronic voting system, study current electronic voting systems and identify drawbacks, study Bitcoin based online transaction systems and identify how it can be incorporated with an electronic voting system, identify additional technologies and protocols required to ensure security and finally design and verify a Bitcoin based secure electronic voting system.

1.4. Organization of the Thesis

The remainder of this thesis is structured the following way:

- **Chapter 2: Literature Review on e-Voting and e-Cash Systems**
  This chapter reviews the previous work related to voting systems and technologies, functionality of electronic voting and electronic cash systems. The identified list of features and properties, strengths and weaknesses, and the implementation techniques and technologies have been discussed in the chapter.

- **Chapter 3: Methodology for Designing Bitcoin Based Electronic Voting System**
  A comprehensive list of properties and features expected of an e-cash system and e-voting system have been analysed in this chapter to show how different properties/features of an e-voting system map to an e-cash system. We have shown how various functionalities of a bitcoin-like system directly provide the required features/properties of an e-voting system. Also, we have shown how various functionalities of a bitcoin-like system can be modified and/or adapted to provide some of the other required features/properties of an e-voting system.

- **Chapter 4: System Design and Implementation**

  This chapter proposes the architecture for e-voting system that is built on blockchain technology.

- **Chapter 5: System Evaluation and Performance Review**

  This chapter discuss how strongly various security and performance requirements are being met in the research work related to the proposed e-voting system.

- **Chapter 6: Conclusions**

  This chapter contains a summary of the research work that has been conducted and discussion of the limitations and shortcomings. At the end we summarize the results, introduce possible future works, and conclude the thesis.

Chapter 2

Literature Review on e-Voting and e-Cash Systems

This chapter reviews the previous work related to voting systems and technologies, functionality of electronic voting and electronic cash systems. The first section of the review provides an overview of the existing voting systems including but not limited to electronic voting systems. The second section of the chapter consists of extracted sections from the papers and reports that highlight the identified security vulnerabilities and relevant recommendations with related to improving the integrity of the elections. Also, the previous attempts to preserving the integrity of the elections have been discussed in this section.

2.1. Existing Voting Systems

Even though the research is focused on electronic voting systems, it is important to understand how the previous and existing election systems work with their advantages and disadvantages, especially when it comes to achieving security goals. Based on the user experience these voting systems can be categorized as paper ballots and electronic voting systems. Some outdated voting systems, such as punch card voting systems and lever voting systems that are also needed to be replaced according to legal systems, Help America Vote Act of 2002 [9] are not discussed here.

2.1.1 Paper Ballots

In paper ballot systems, voters are given an official ballot, where they mark the choice with a pen or pencil, verify and put it into a sealed box. Optical Scan paper ballot systems (aka Marksense) also come under this category. When conducting the elections with the Marksense systems, the election authority scans them on optical scan systems that maybe placed precinct-based polling places. Also, it is possible to collect them in a ballot box and scan them at a central location.

The main advantage of this system is that there is a hard copy proof of the voter's choice that can be used to conduct as many as independent audits and recounts. Also, voters themselves can ensure that the vote has been cast as intended by them, before putting their ballot in a box.

2.1.2 Electronic Voting Systems

An electronic (aka DRE) voting systems, incorporate interfaces - touchscreens, dials and mechanical buttons, through which a voter can record the vote into electronic storage – a memory cartridge, diskette or smart card. DRE may also be equipped with an alphabetic keyboard to allow for the possibility of write-in votes [10].

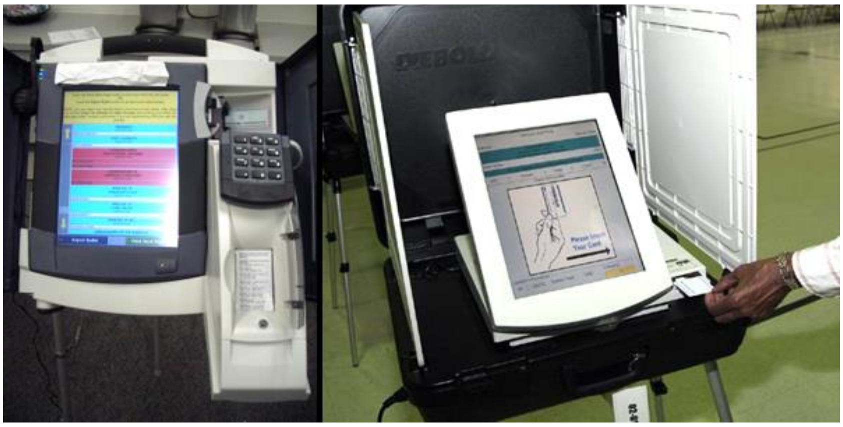
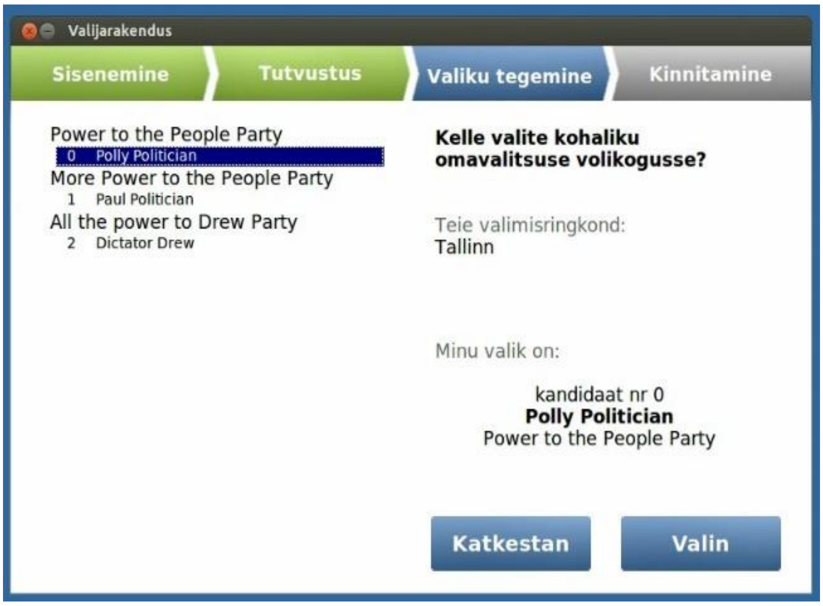


Figure 2.1 AccuVote TS and TSX touch screen DRE voting machines Source: verifiedvoting.org

2.1.3 Internet Voting

Voting over the internet is becoming popular over the time. Even though there are multiple attempts by several countries to conduct interned based elections, Estonia was the first country to implement a such internet based system and conduct a nation-wide election [6]. This system allowed user to cast signed and encrypted vote using a client application and send it to a server over a client-authenticated TLS connection. The system allowed users to vote more than once to prevent coercion, where the last vote was counted as the valid vote. Voters were able to confirm that the vote has been recorded properly using a mobile application that is developed by the election authority.



Power to the People Party
0 Polly Politician
More Power to the People Party
1 Paul Politician
All the power to Drew Party
2 Dictator Drew

Kelle valite kohaliku omavalitsuse volikogusse?

Teie valimisringkond:
Tallinn

Minu valik on:

kandidaat nr 0
Polly Politician
Power to the People Party

Figure 2.2 I-voting client used by Estonians to cast votes online [8]

2.2. Related Work

As discussed in the introduction of the chapter, this section provides a summary of the papers and reports that highlight the identified security vulnerabilities and relevant recommendations with related to improving the integrity of the elections. Moreover, the end-to-end voting systems which are used to preserve the integrity of the paper and semi-electronic based systems have been discussed at the latter part of the section.

2.2.1. Voting: What Has Changed, What Hasn't, & What Needs Improvement

The Caltech/MIT Voting Technology Project was formally announced December 15, 2000, “to prevent the recurrence of the problems that threatened the 2000 presidential election.” The team was formed by ten faculty members and around fifty graduate/undergraduate students from the Massachusetts Institute of Technology and California Institute of Technology.

In the report [4] presented in 2012, they have examined how the US election administration and voting technologies have changed since the debated presidential election that was held in 2000, and what have not changed since then. They have presented the perspectives of many individuals their study and analysis, who were involved in the election advocacy communities, the election administration as well as voting technologies.

Under the section “Technical proposals for security improvements” they have mentioned proposals for end-to-end voting systems. This “end-to-end” (E2E) voting system are the system that can provide verifiability from the starting to the end - that is from choices in the voter’s mind to the final tally. Votes should be able to verify that the vote bas been cast and recorded accurately and anyone else should be able to verify that it is tallied as recorded.

Mentioned two proposals “Prêt à Voter” and “Scantegrity’’ [11] [12] are paper base elections and use a cryptographically verified implementation with a public website where the voters are provided with a facility to verify that their encrypted votes are correctly logged.

2.2.2. Security considerations for remote electronic voting

The paper [2] discusses how remotely operated electronic voting systems should be secured under four main areas: the voting platform, the communications infrastructure, social engineering and specialized devices. It also highlights following risks associated with remote electronic voting:

- **Registration** – if the online registration is allowed, a mechanism should be available to prevent or control the fraud.
- **Vote solicitation** – it will be difficult to prevent or limit vote solicitation by political parties during the time the polling is taking place
- **Coercibility** – it is possible that a voter could be forced or threatened to vote for a candidate since the polling is not held inside a public polling place
- **Vote selling** – voters can sell their votes

The paper concludes that as of 2002, the infrastructure is inadequate for remote Internet voting. It suggested that there should be hardware-level support to establish a 'trusted path' between the election server and the computers used by the voters.

2.2.3. On the notion of 'software independence' in voting systems

It is proposed in the paper [13] that it is required to design and implement software-independent voting systems and it is required to avoid software-dependence in voting systems. For example, if the e-voting software is containing undetected program level bugs, software manipulation or is vulnerable to malicious codes, the result of the entire election will be unacceptable. Therefore, a proper software development practice should be followed and making the software open source will make it easier for others to verify the code and suggest improvements.

2.2.4. Voting Systems Audit Log Study

An audit log record of a voting system will consist of event and performance data of the processes and systems that is required to improve the performance and security of the system. The report [14] emphasizes the need for an audit log along with what should and should not be included in a log file of a voting system while addressing the legal and privacy issues. When designing a secure electronic voting system, it is noted that this aspect also should be taken into the consideration.

2.2.5. An Introduction to Electronic Voting

The potential benefits and risks of electronic voting systems are discussed in the paper [15] with a brief discussion of the current "environment" in terms of current and pending legislation, standards and testing programs, preferred characteristics of voting systems and manufacturers of existing electronic voting systems. The following are the unacceptable events or behaviors that could cause by a malicious code:

- Modification of votes that have been recorded previously
- Modification of vote total
- Denial of service
- Revealing vote total before the polling is finished
- Detecting the votes casted by voters

- Noncompliance and braking of election rules
- Display that the vote has been casted correctly while it is being recorded otherwise in the system
- Allowing someone who is not a registered voter to cast a vote
- Modification of audit trails
- Failing to record votes in the system
- Calculating vote totals incorrectly

Even though it is possible to misuse any of the existing voting systems, including paper ballots, the characteristics of electronic voting systems make them more vulnerable and expose them to significant risks than the others.

It is essential and ideal to have both accuracy and privacy in a perfect voting system. However, it is not simple and easy to achieve both simultaneously. The paper states that to ensure that eligible voter's intention is reflected in the final tally, it is required to have a back-channel to the voter. It suggests that the backchannel to the voter would compromise the voter's privacy.

While the above literature provides an adequate background and expected features of an electronic voting system, following papers provides an overview of current electronic voting systems.

2.2.6. Security Analysis of the Estonian Internet Voting System

The paper [6], analyzes the security features of the I-voting system that is used in Estonia to conduct a nationwide internet based election in 2014 where the ballots cased online was beyond 30%. The analysis consists of adversarial testing, review of the code and in-person election observation

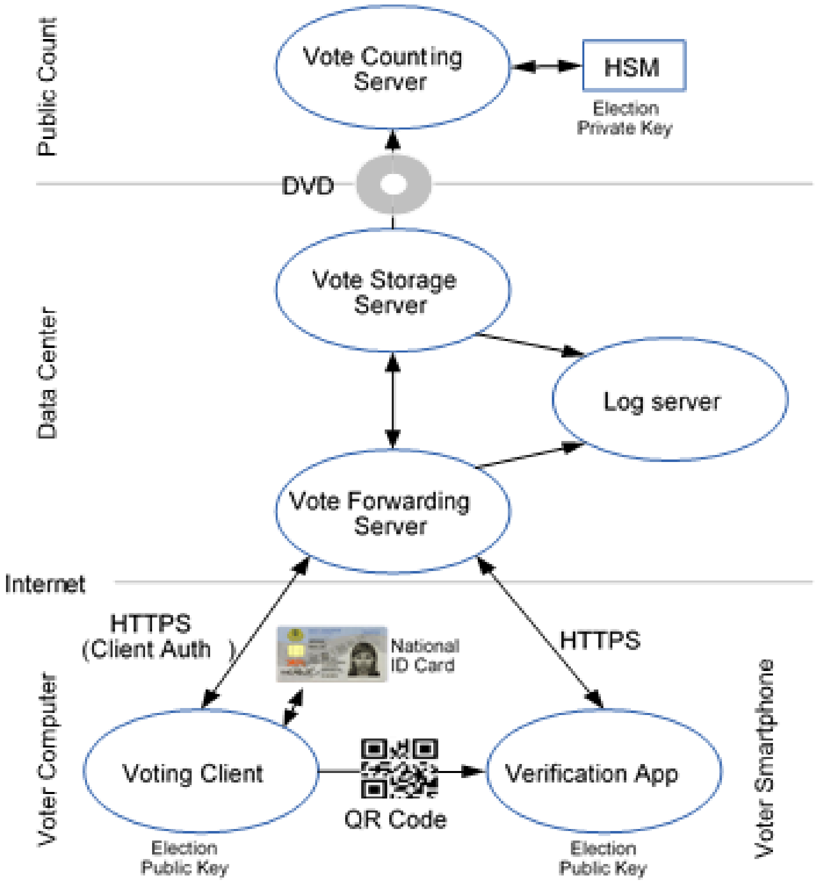
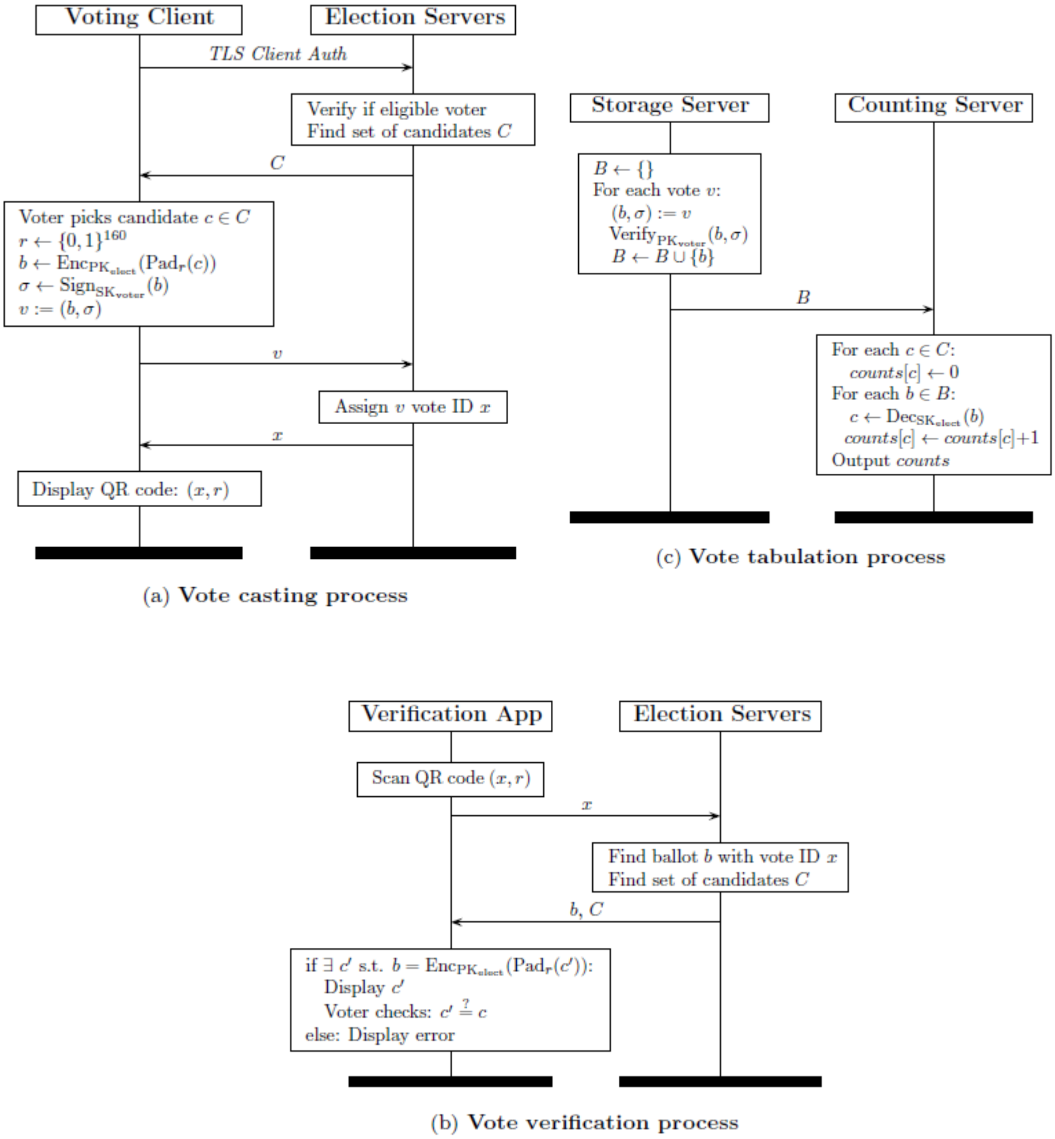


Figure 2.3: The ovevirew of the i-voting system

Even though authors recommend to discontinue the I-voting system after an extensive analysis, we may use the approach of Estonia along with improvements suggested by the authors to design a basic architecture for the proposed system.


Voting Client
Election Servers
TLS Client Auth
Verify if eligible voter
Find set of candidates C
C
Voter picks candidate $c \in C$
$r \leftarrow \{0,1\}^{160}$
$b \leftarrow \mathrm{Enc}_{\mathrm{PK}_{\mathrm{elect}}}(\mathrm{Pad}_r(c))$
$\sigma \leftarrow \mathrm{Sign}_{\mathrm{SK}_{\mathrm{voter}}}(b)$
$v := (b, \sigma)$
v
Assign v vote ID x
x
Display QR code: (x, r)
Storage Server
Counting Server
$B \leftarrow \{\}$
For each vote v:
$(b, \sigma) := v$
$\mathrm{Verify}_{\mathrm{PK}_{\mathrm{voter}}}(b, \sigma)$
$B \leftarrow B \cup \{b\}$
B
For each $c \in C$:
$counts[c] \leftarrow 0$
For each $b \in B$:
$c \leftarrow \mathrm{Dec}_{\mathrm{SK}_{\mathrm{elect}}}(b)$
$counts[c] \leftarrow counts[c]+1$
Output $counts$
Verification App
Election Servers
Scan QR code (x, r)
x
Find ballot b with vote ID x
Find set of candidates C
b, C
if $\exists\, c'$ s.t. $b = \mathrm{Enc}_{\mathrm{PK}_{\mathrm{elect}}}(\mathrm{Pad}_r(c'))$:
Display c'
Voter checks: $c' \overset{?}{=} c$
else: Display error

(a) **Vote casting process**

(c) **Vote tabulation process**

(b) **Vote verification process**

Figure 2.4: (a, b, c) i-Voting system protocols

2.2.7. Attacking the Washington, D.C. Internet Voting System

The paper [7] highlights one of the key dangers in many Internet voting designs: a slight error in the design or operation of the central voting servers or the network infrastructure around them can easily undermine the credibility of the entire election. Even though we are not interested in the attacking part, the report can be used to obtain an idea about how the internet system designed and what were the weaknesses. The implementation had a vulnerability in the code that is used to encrypt the voted ballots that were uploaded by users. The authors have compromised the application server by exposing this shell injection vulnerability.

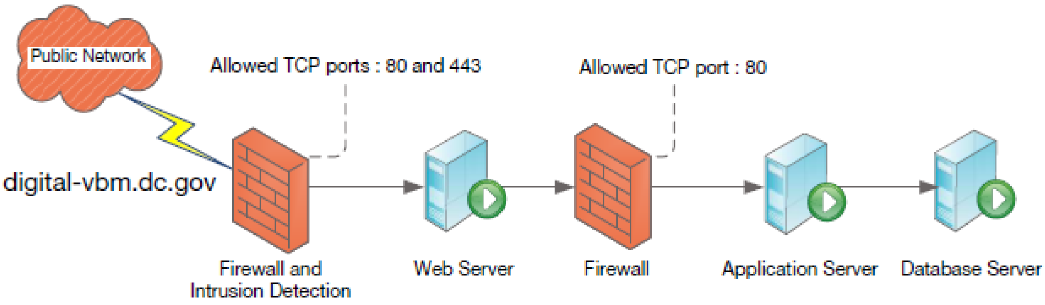
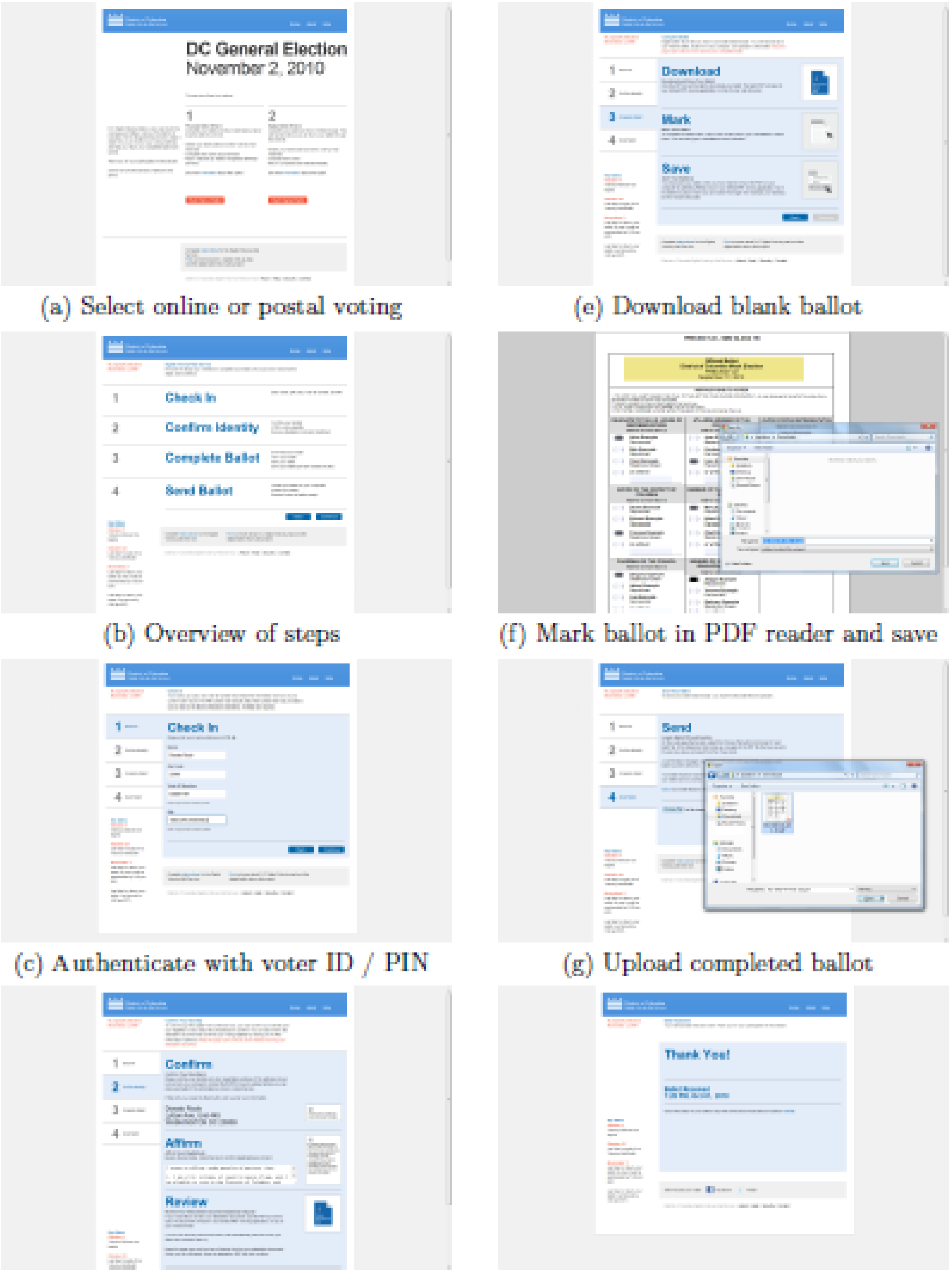


Figure 2.5 Network architecture of the Washington, D.C. Internet Voting System

The architecture was consisting of several firewalls to reduce the attack surface. The front-ending web server accepts HTTPS requests from voters and reverse-proxies those requests to the application server. The application servers hosts the DVBM election software. It is also used to stores blank ballots and completed ballots. A MySQL database server was used to store voted ballots and voter credentials. The firewalls are used to deny outbound TCP connections. Even though there was an intrusion detection system in front of the web server, it has failed to decrypt the HTTPS connections that carried the authors' exploit, proving that it is ineffective.



(a) Select online or postal voting

(e) Download blank ballot

(b) Overview of steps

(f) Mark ballot in PDF reader and save

(c) Authenticate with voter ID / PIN

(g) Upload completed ballot

(d) "Affirm" identity

(h) "Thank you" screen

Figure 2.6: Screenshots of the Washington, D.C. Internet Voting System

It is concluded that although new end-to-end verifiable cryptographic voting schemes have the potential to reduce the trust placed in servers and clients, these proposals are significantly more advanced than systems like D.C.'s. It is stated that it will be much more difficult for developers to implement such system correctly.

2.2.8. End-to-end voting systems.

An "end-to-end" (E2E) voting system provides verifiability from the starting point (the choices in the voter's mind) to the final tally. Votes should be able to verify that the vote bas been cast and recorded accurately and anyone else should be able to verify that it is tallied as recorded. Overall, an E2E voting system will this provide a level of verification of the election outcome that is not available in the existing and commonly used voting systems.

Many proposals for E2E voting systems have been made which usually involve the use of cryptography and a website where voters can verify that their encrypted votes are logged correctly. Checking that the encryption of ballots is carried out correctly and ensuring that the count of the encrypted ballots is accurate is usually difficult but is not impossible.

The "Prêt à Voter" system [11] is an "end-to-end" voting system that uses a two-part paper ballot. One of the two parts contain the candidate names in a scrambled order. The remining part contain the voter's choices and an encoding of the name permutation. The voter discards the first part and casts only the second part.

The "Scantegrity'' system [12] uses an innovative invisible-ink method on optical-scan paper ballots that are similar to the regular ballots. However, it reveals a secret "confirmation code'' when the voter marks a bubble using a special pen. The voter can visit a website later and search these codes to confirm that his or her ballot has been recorded properly. This E2E voting system used to conduct two binding governmental elections, in Takoma Park, Maryland and proven to be successful.

2.2.9. Blockchain based e-voting systems.

Zcash based e-voting system [16] was proposed in 2017 as an attempt of implementing blockchain based e-voting systems. The cash system uses four types of addresses that are used to spend, view, pay and transmit of transactions. The following diagram summarizes the voting process of the proposed system.

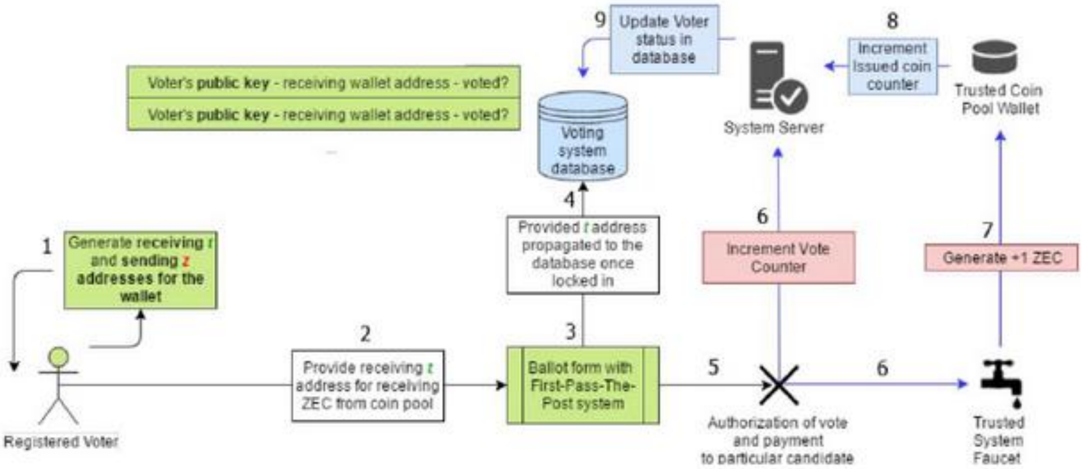
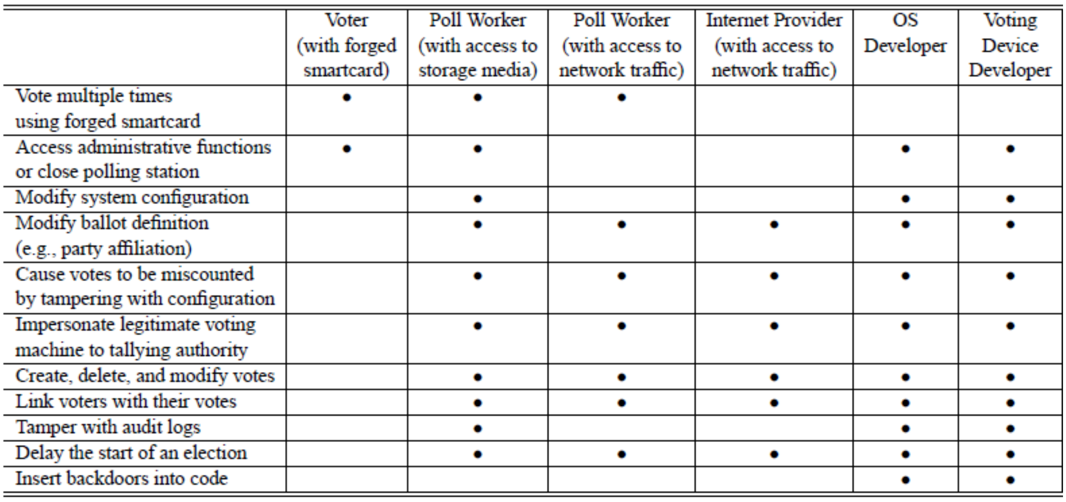

9
Update Voter status in database
8
Increment Issued coin counter
Voter's public key - receiving wallet address - voted?
Voter's public key - receiving wallet address - voted?
Voting system database
System Server
Trusted Coin Pool Wallet
4
6
7
1
Generate receiving t and sending z addresses for the wallet
Provided t address propagated to the database once locked in
Increment Vote Counter
Generate +1 ZEC
2
3
Provide receiving t address for receiving ZEC from coin pool
Ballot form with First-Pass-The-Post system
5
6
Registered Voter
Authorization of vote and payment to particular candidate
Trusted System Faucet

Figure 2.7 Zcash based e-voting system

2.2.10. Summary

After looking at the existing electronic voting systems it was identified that the trust that the voter has to place on voting systems is very high. Parties including poll workers, Internet service providers, OS developers and voting device developers can attack or misuse the system.

Table 2.1 Attacks on the Diebold AccuVote-TS 4.3.1 system



	Voter (with forged smartcard)	Poll Worker (with access to storage media)	Poll Worker (with access to network traffic)	Internet Provider (with access to network traffic)	OS Developer	Voting Device Developer
Vote multiple times using forged smartcard	•	•	•			
Access administrative functions or close polling station	•	•			•	•
Modify system configuration		•			•	•
Modify ballot definition (e.g., party affiliation)		•	•	•	•	•
Cause votes to be miscounted by tampering with configuration		•	•	•	•	•
Impersonate legitimate voting machine to tallying authority		•	•	•	•	•
Create, delete, and modify votes		•	•	•	•	•
Link voters with their votes		•	•	•	•	•
Tamper with audit logs		•			•	•
Delay the start of an election		•	•	•	•	•
Insert backdoors into code					•	•

For example, above table summarizes some of the attacks on the Diebold AccuVote-TS 4.3.1 system [17] that was used in US elections. It was written in C++ and was designed to run on a Windows CE device.

The code level vulnerabilities can exist in any application. However, making the software open-source and available to the public will ensure that there is no hidden malicious code or vulnerabilities in the application. It is was decided to limit the scope of the solution proposed in this paper to focus on the end-to-end verification using the Bitcoin based implementation where it will rely on trustworthy code level implementation and the presence of legal system to prevent the misuse.

Chapter 3

Methodology for Designing Bitcoin Based Electronic Voting System

3.1. Introduction

The technology behind the solution proposed in this paper relies on Blockchain, a decentralized ledger system that has become popular with the introduction of Bitcoin [5] digital currency. The decentralized nature of blockchain allows peers to perform transactions across a network without control by any single entity. An implementation similar to Bitcoin system will effectively fulfil the confidentiality and integrity requirements of an electronic voting system, as both share a common feature set.

As the methodology for designing and implementing a secure electronic voting system, we first take an existing, well-proven, robust, scalable solution called bitcoin, which is an e-cash system. The chapter provides a comprehensive list of properties and features expected of an e-cash system and present how Bitcoin satisfies the listed properties and features expected of an e-cash system with the implementation details.

Thereafter, we provide a comprehensive list of properties and features expected of an e-voting system and show how different properties/features of an e-voting system map to an e-cash system.

Then we show how various functionalities of a bitcoin-like system directly provide the required features/properties of an e-voting system. We also show how various functionalities of a bitcoin-like system can be modified and/or adapted to provide some of the other required features/properties of an e-voting system.

Finally, we show what features/properties of an e-voting system are not being provided by the bitcoin-like system.

3.2. List of Properties and Features Expected of an e-cash System

For an electronic cash system to be accepted among the public, it should possess the properties and features of traditional paper or coin-based cash systems and optionally any additional advantages. It is suggested that the value of e-cash or cryptocurrencies exists as long as users have trust and acceptance in the system [18]. Many papers suggest that the security and trust are the most expected properties of an e-cash system [5], [19], [20]. The identified security and other properties and features expected of an e-cash system are given below:

1. **Transferability** – Easy circulation, transferring money from one to another should be facilitated.
2. **Anonymity** – It is expected that user's identity is not revealed during a transaction and the e-cash cannot be tied with user's identity.
3. **Authenticity/recipient verification** – To make sure the payment is made to the intended party, for example, to ensure the transaction is made to the correct account number.
4. **Tamper-resistance -** Transaction details (such as the transaction amount and account number) should not be altered during or after the transaction.
5. **Unforgeability -** The e-cash should not be produced by anyone but an authorized party
6. **Double spending detection** – It must be ensured that the e-cash can be spent only once.
7. **Date/Time attachability** – Date and time of the transactions such as withdrawing, paying, and depositing should be recorded. This date and time can be used for interest calculation or to make sure that the payment is made on time.
8. **Divisibility -** It should be possible to divide e-cash into small denominations.

9. **Portability** – the portability or mobility allows users to perform transactions using desktop or mobile without being physically present at a bank/merchant.
10. **Anonymity revocation** – the desired feature will allow to trace the owner of the e-cash if there is a misuse or an illegal activity is taken place.

3.3. Properties and features of Bitcoin

Bitcoin, is the most popular peer-to-peer version of electronic cash that allows online payments to be sent directly from one party to another without going through a financial institution [5]. The following subsections demonstrate how Bitcoin satisfies the list of properties and features expected of an e-cash system.

3.3.1. Transferability

The diagram below shows how a Bitcoin transaction is performed between Alice and Bob, where Alice pay to Bob and Bob can spend that later. It will use the Pay-To-Public-Key-Hash (P2PKH) which is common among other transaction types used in Bitcoin. This P2PKH payment method lets Alice spend money to a Bitcoin address owned by Bob. Bob can further spend those money using a simple cryptographic key pair.

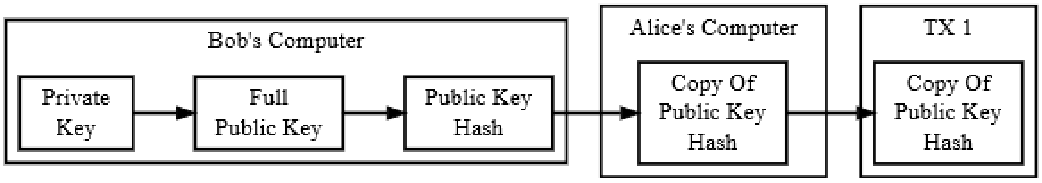


Figure 3.1 Creating a P2PKH Public Key Hash to Receive Payment *[21]*

First, Bob will create his Private/Public Key pair using Elliptic Curve Digital Signature Algorithm (ECDSA) with the secp256k1 curve [22]. The graph of secp256k1's elliptic curve is provided by $y^2 = x^3 + 7$.

These secp256k1 private key consists of 256 bits of random data. The sextuple $T = (p,a,b,G,n,h)$ are the elliptic curve domain parameters over finite field Fp associated with a Koblitz curve secp256k1 [23]

With the recommended parameters, Fp is defined by:

$p =$ FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFE FFFFFC2F
$= 2^{256} - 2^{32} - 2^{9} - 2^{8} - 2^{7} - 2^{6} - 2^{4} - 1$

The curve E: $y^2 = x^3 + ax + b$ over Fp is defined by:

$a =$ 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000

$b =$ 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000007

The base point G in compressed form is:

$G =$ 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB
2DCE28D9 59F2815B 16F81798

and in uncompressed form is:

$G =$ 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB
2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465
5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F
FB10D4B8

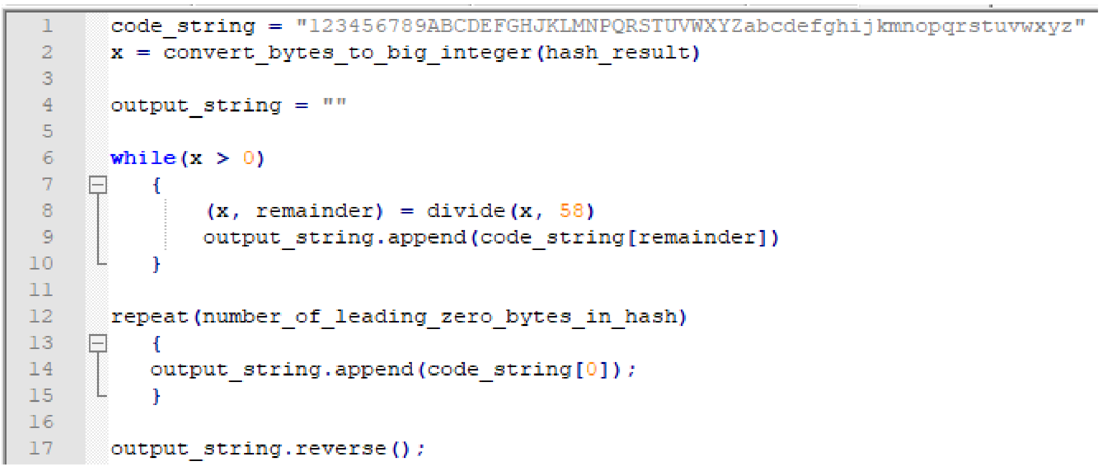
Finally, the order n of G and the cofactor are:

$n =$ FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6
AF48A03B BFD25E8 CD0364141

$h =$ 01

The derived public key is then hashed which obfuscates and shorten the public key.

Bob then provides a copy of the 160-bit public key hash to Alice, as a *Bitcoin Address* which is encoded using Base 58 binary-to-text encoding [24] along with the address version number, hash and a checksum.



```
code_string = "123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz"
x = convert_bytes_to_big_integer(hash_result)

output_string = ""

while(x > 0)
   {
       (x, remainder) = divide(x, 58)
       output_string.append(code_string[remainder])
   }

repeat(number_of_leading_zero_bytes_in_hash)
   {
   output_string.append(code_string[0]);
   }

output_string.reverse();
```

Figure 3.2 Example Base 58 Encoding Algorithm *[24]*

The Bitcoin Address can be transferred as a QR code with the Bitcoin URI. The URI may contain additional information such as the amount to be paid.

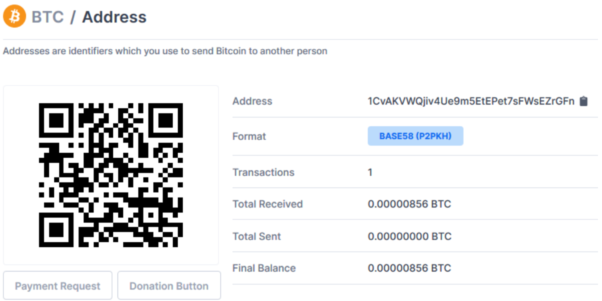
Low
Medium
Quartile
High
bitcoin:mjSk1Ny9spzU2fouzYgLqGUD8U41iR35QN?amount=.1

Figure 3.3 QR Codes Containing a Bitcoin URI

Alice decodes the Bitcoin address back into a standard hash, to create the first transaction. She creates a standard P2PKH transaction output containing instructions which allow anyone who has Bob's private key to spend it. These instructions, which are identified as the pubkey script or scriptPubKey [25] are broadcasted and add to the block chain. This will be added to the Bob's wallet application as a spendable balance.

3.3.2. Anonymity

There is no credit card number or any other personal identification information that malicious actors can collect in order to steal e-cash. Also, the decentralized nature of the system allows users to send a payment without revealing your identity, almost like with physical money.



BTC / Address

Addresses are identifiers which you use to send Bitcoin to another person

Address	1CvAKVWQjiv4Ue9m5EtEPet7sFWsEZrGFn
Format	BASE58 (P2PKH)
Transactions	1
Total Received	0.00000856 BTC
Total Sent	0.00000000 BTC
Final Balance	0.00000856 BTC

Figure 3.4 Only the Bitcoin Address is Required to Make a Payment

As discussed in the previous sub-section, Bob must generate a private/public key pair and convert it to a bitcoin address before giving it to Alice, so she can transfer money to that address. Note that Bob's identity is not revealed during the process. Also, Bitcoin recommends users to generate new address for each transaction as others can easily track the receiving and spending habits of that person (even though they do not know the identity), including the balance they maintain in known addresses.

3.3.3. Authenticity and Tamper-Resistance

Bitcoin uses Elliptic Curve Digital Signature Algorithm (ECDSA) to prove that payee (Bob) owns the address. Scripts, which sets the conditions that must be fulfilled for e-cash to be spent, combine secp256k1 public keys and signatures with conditional logic to create a programmable authorization mechanism.

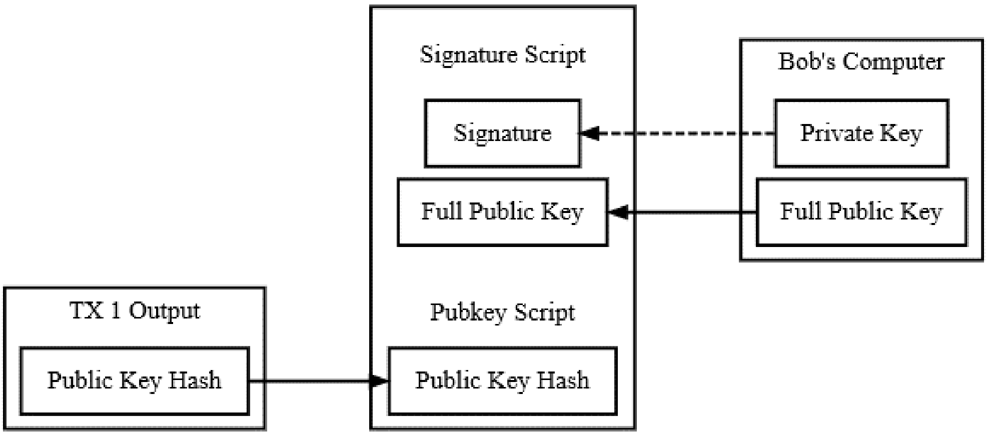


Figure 3.5 Transaction Data is Signed with Bob's Private Key

Bob’s secp256k1 signature proves that Bob controls his private key and also makes the non-signature-script parts of his transaction tamper-proof, so it can be safely broadcasted over the peer-to-peer network.

3.3.4. Unforgeability

The first transaction is always created by a miner, with no outputs from a previous transaction. While standard transactions use 'inputs' section to refer to their previous transaction outputs, a generation transaction does not have any previous transactions, and creates new coins from nothing. Each subsequent transaction is signed with the spender’s private key using the Elliptic Curve Digital Signature Algorithm.

3.3.5. Double spending avoidance/detection

Bitcoin timestamps the transactions by hashing them and attaching them into an ongoing chain of hash-based proof-of-work. This forms a record that is unchangeable without doing the proof-of-work again, hence providing an effective solution against double spending.

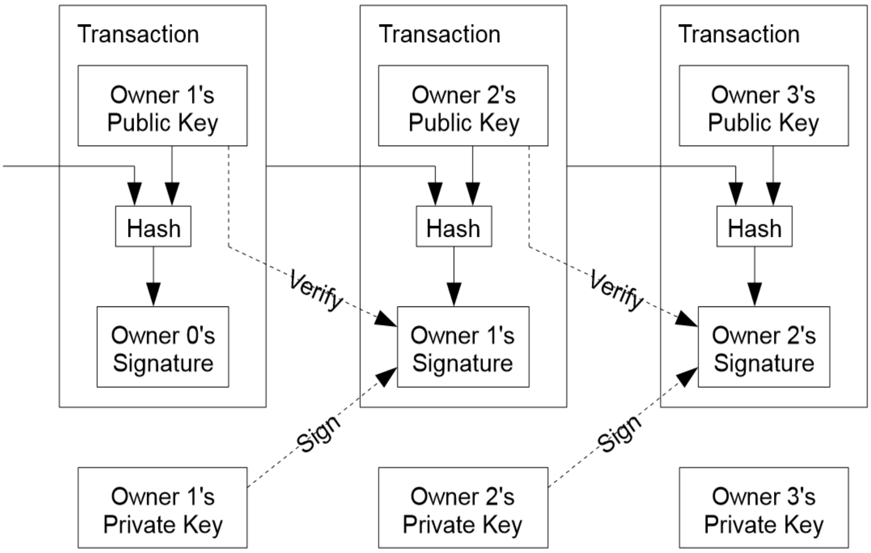


Figure 3.6: Bitcoin Transactions - an electronic coin is defined as a chain of digital signatures. Source: bitcoin.org

In Bitcoin cash system, illustrated in figure 3.1, each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

3.3.6. Date/Time attachability

Bitcoin maintains a timestamp server. The system takes a hash of a block of transactions that needs to be timestamped and then it publishes the hash into the network. The timestamp is the proof that the data was existed at the time, in order to get into the hash. Since the previous timestamp is included in the new hash, forming a chain, it is impossible to change it without doing the proof-of-work again.

3.3.7. Divisibility

Denominations of Bitcoin value can be measured in fractions or as multiples of a satoshi. One bitcoin is equivalent to 100,000,000 satoshis.

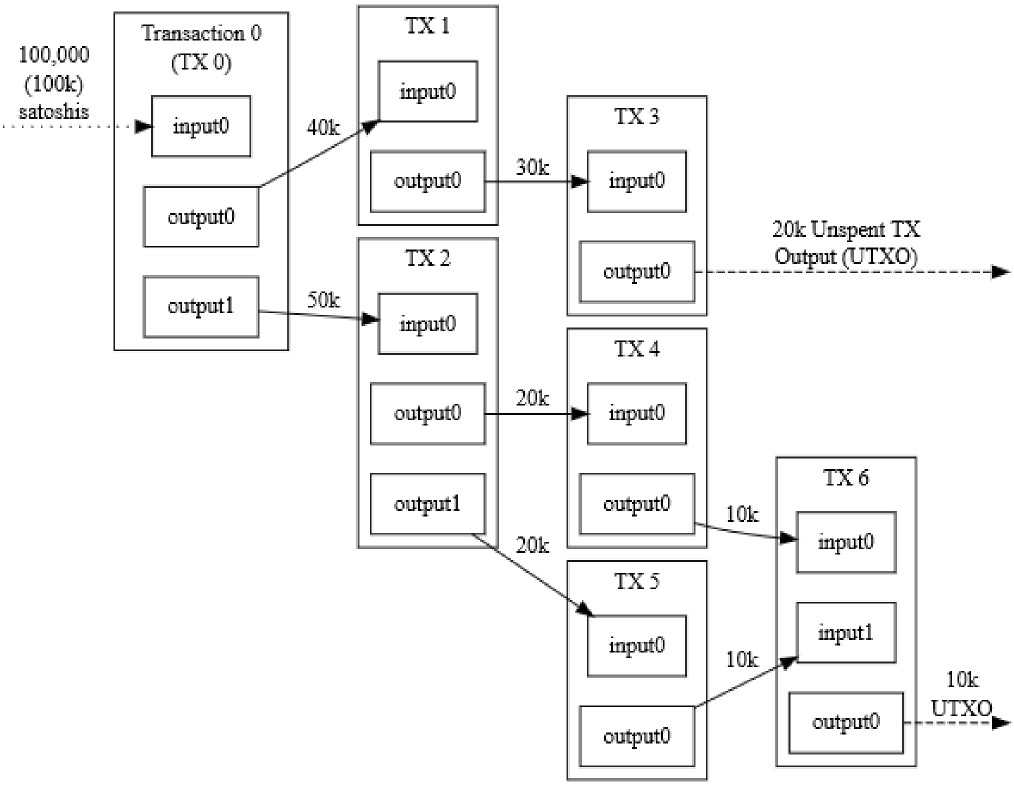


Figure 3.7 Each Transaction Spends Previously Received Satoshis

3.3.8. Portability

The portability of Bitcoin allows users to perform transactions using desktop or mobile without being physically present at a bank/merchant. Payments can be made from a wallet application, that is installed on a computer or smartphone. The recipient's address and the payment amount are the only parameters required during a transaction. Most wallet applications allow filling the recipient’s address by scanning a QR code or using NFC.

3.4. List of Properties and Features Expected of an e-voting System

Electronic voting or e-voting is a relatively novel concept similar to e-cash and are introduced to replace the traditional voting systems such as popular paper ballots. Even though adopting technology has advantages such as efficiency and convenience, it may introduce additional security and privacy issues if adequate security measures are not taken place [26]. Following are the identified properties and features expected of an e-voting system:

1. **Authorization** - Only authorized voters can vote.
2. **Anonymity** – It should not be possible to bind an identity with a vote. No one can determine for whom anyone else voted.
3. **Tamper-resistance-** No one can change anyone else's vote without being discovered.
4. **Restriction of attempts** - No one can vote more than once.
5. **Prevention of impersonation** - No one can duplicate anyone else's vote.
6. **Individual Verifiability** - Voter should be able to verify that the vote has been counted.
7. **Universal Verifiability** - Everyone can verify the election result without compromising the privacy of the voters

There are three types of actors involved in a typical election process. Those include voters, registration authority and tallying authority. In some cases, the registration and tallying body could be operated under the same governing body. As the first step of the election process, the voter registration should be taken place by the registration authority to identify and register eligible voters with the system. Later, the authorization process will disallow any unregistered person to participate in the election.

Even though there are circumstances where it is possible to conduct open ballots; in which voters can vote openly; most of elections are conducted as closed ballots. In such systems, the voter's choice should be kept confidential from others by making the vote secret which will be a key expectation of the voter.

Tamper-resistance is another key requirement of an e-voting system. Studies suggest that a significant number of people are not willing to trust e-voting systems as they are uncertain about the authenticity and integrity of the voting machines, and the votes cast using those machines. Also, it is required to make sure that one can cast a vote more than once, either using the same identity or using someone else's identity.

Finally, the verification is required, especially when there is no trusted party or independent auditors to ensure the integrity of the voting process. We further categorize the verifiability as individual verifiability and universal verifiability. In an individual verifiable system, each individual voter is responsible for insuring that his or her vote has been accounted for in the final tally. However, in this setup, it is not possible to conduct an independent audit by a third party to verify the integrity of the election without compromising the voter's privacy. A universally verifiable system will provide a solution to this by allowing anybody to verify the election without compromising voter's privacy

3.5. Mapping the properties/features of an e-voting system to an e-cash system

After a careful analysis of the properties and features of an e-cash systems and e-voting systems, it has been observed that both systems share some common characteristics.

Table 3.1 Similar Features in e-cash and e-voting systems

Property/Feature	**Requirement in e-cash systems**	**Requirement in e-voting systems**
Anonymity	Ensures that payer's or payee's identity cannot be mapped with e-cash	Voter's identity is not revealed during a secret ballot
Authenticity/recipient verification	To make sure the payment is made to the intended party	Ensures that vote is casted for the intended party

Tamper-resistance	Transaction details should not be altered after making a payment	Vote should not be changed after casting
Verifiability	To make sure that the payment is approved	To ensure the vote is counted in the final tally
Double spending detection	the e-cash can be spent only once	No one can vote more than once.

The anonymity in the e-cash system is identical to the anonymity requirement in a e-voting system which help users to maintain privacy within the system. Similarly, the authenticity/recipient verification, tamper-proofing, verifiability and double spending detection are identified as the mutual expectations.

Apert from the above security specific features, both e-cash and e-voting systems share common usability, accessibility, and performance requirements. Both maybe implemented as a dedicated/embedded device (ATM or voting machine) or as an application running on an ordinary computer or mobile device.

3.6. Adopting Functionalities of a Bitcoin-like System for an e-voting System

A clear identification of similarities and differentiations between e-cash system and e-voting system make the designing and implementation of the Bitcoin based e-voting system effortless. Also, the well proven current architecture, protocols and code level implementation will make the development and secure.

We identify that the functionalities of a bitcoin-like system directly provide the anonymity, authenticity/recipient verification, tamper-resistance, and verifiability that are expected of an e-voting system.

A minimal Bitcoin payment protocol consists of following steps:

1. Customer completes selection of items (adding items to a shopping cart) and decides to make a payment using Bitcoin.

2. Merchant provides a newly generated unique payment address which is associated it with the order and send it to the customer.

3. Customer enters Bitcoin address in the wallet application manually or by scanning a QR code.

4. Customer authorizes payment and broadcasts the transaction through the Bitcoin peer-to-peer network.

5. Merchant's server detects payment and considers the transaction final after sufficient transaction confirmations.

The Bitcoin Improvement Proposal (BIP) 70 adds following additional security features to the initial protocol which is directly applicable to a similar e-voting implementation:

1. Customers may be asked to authorize payment to an identity such a hostname that is verified by X.509 public key certificate

2. A proof of payment, a digital receipt, that can be used for dispute resolution.

3. Resistance to man-in-the-middle attacks which can be used to replace the payment address of the merchant with the address of the attacker before a transaction is authorized.

4. Payment acknowledgement messages to notify the end of the successful transaction

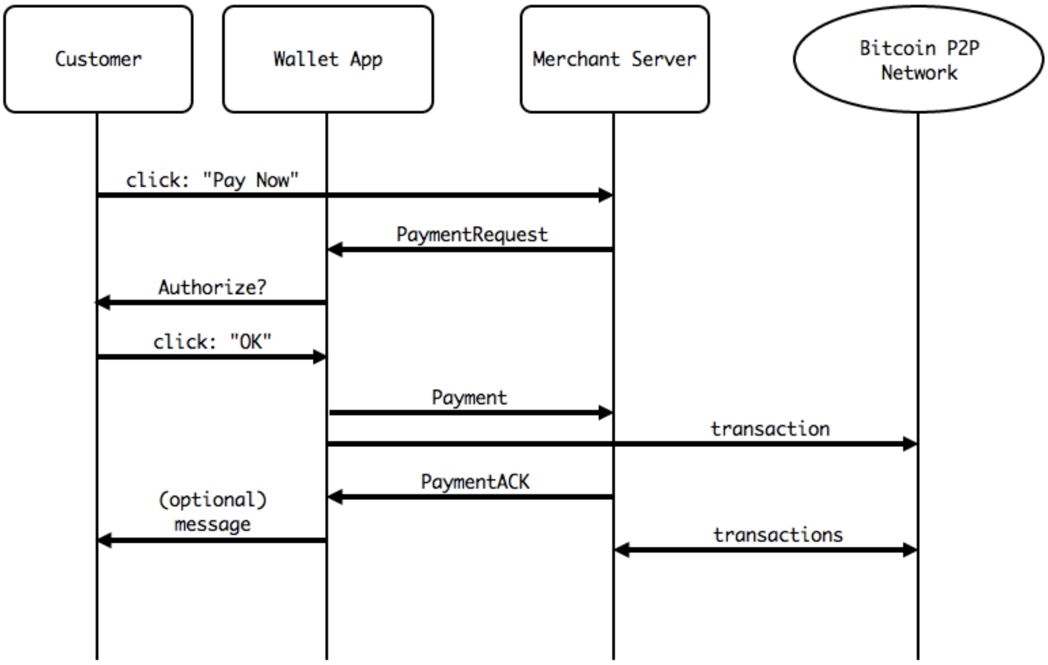


Figure 3.8 Bitcoin Payment Protocol Proposed in BIP 70

When the Bitcoin wallet software receives a PaymentRequest, the following actions are taken place to authorize the payment:

1. If it a payment to an identity (hostname, individual or an organization), validate the merchant's identity and signature using the PKI

2. Validate that the payment request is not expired by comparing the PaymentDetails.expires parameter with customer's system unix time (UTC). If expired, reject the payment

3. The merchant's identity such as "Common Name" in the certificate is extracted and displayed to the customer with an option to submit/reject the payment.

4. To mitigate denial-of-service attacks, PaymentRequest messages that are larger than 50,000 bytes must be rejected by the wallet application.

After authorizing the payment, the Bitcoin wallet application will:

1. Creates and signs transactions that is provide PaymentDetails.outputs parameter of the PaymentRequest. Based on the PKI type, SHA-1 or SHA-256 is used for signing

2. Validate that the payment request is still not expired by comparing the PaymentDetails.expires parameter with customer's system unix time (UTC). If expired, cancel the payment.

3. Broadcast the transaction on the peer-to-peer network.

Further, we can modify the double spending avoidance requirement of the e-cash system to adopt to the requirement of the e-voting system. In our implementation, we will consider the earliest vote is the one that counts, so we can discard later attempts as invalid or illegal transactions. Timestamping solutions provided through Bitcoin-like systems help identifying the order of transactions.

The registration of users based on the eligibility is not a feature that is directly provided though a bitcoin-like system. We can have a closer look at *Bitcoin Currency Exchanges* to provide a solution for this requirement. Bitcoin currency exchanges operate as banks or a regular currency exchange. An interested party who wants to buy Bitcoin can deposit money in the currencies supported by the exchange, and in return, the exchange will transfer Bitcoins to the buyer.

Usually the purchasing Bitcoin at a currency exchange will require a registration and identity verification to avoid financial fraud. A similar verification can be done in an e-voting system to ensure that only the eligible voters are registered with the system. Communications with the Bitcoin currency exchanges are performed using a standard web browser, over an SSL connection which can be applicable to an e-voter registration process.

Chapter 4

System Design and Implementation

4.1. Introduction

This chapter describes the proposed architecture design of the Bitcoin based secure electronic voting system. The first section of the chapter describes the assumptions that are made during the design for the successful operation of the system. The remaining three sections provides architecture for the major phases of the election process, that are:

- Pre-election
- The election
- Tabulation

4.2. General Assumptions

The E2E verifiable voting system that we have proposed here focuses on the election process starting from voter obtaining an electronic ballot paper/vote to the final tabulation and release of results. However, some pre- and post-election activities that are included in a typical election, such as voter registration, are not within the scope of the proposed system. Therefore, the trustworthiness of the entire election process relies on the assumptions and the context defined below.

- *Presence of an election authority.* The election authority/election department/election committee or a similar body should exist. The primary responsibility of the election authority is registering all the eligible voters with the system thus preparing the *electoral register*.
- We assume that the election committee is trusted during the pre-election setup, which ensures that only and all the eligible voters are included in the electoral register prior to the election and transfers exactly one vote to the voters.

4.3. Pre-election setup

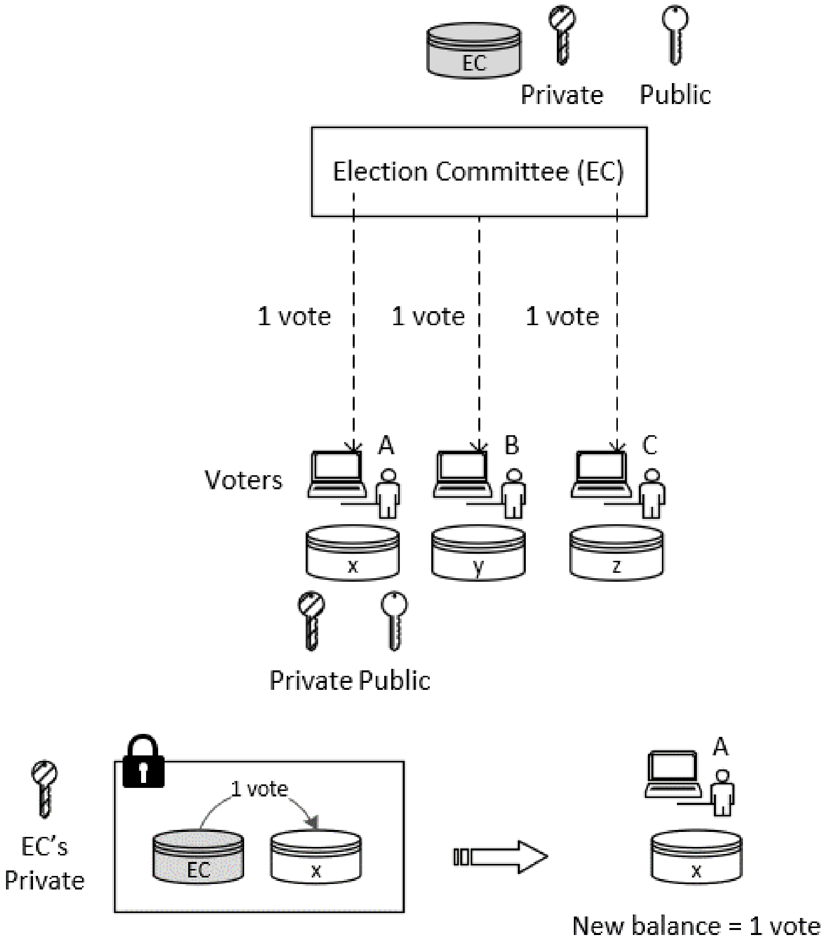


Figure 4.1 Pre-election Setup

The transactions performed during the pre-election stage is given below:

1) Election Committee (EC) has an "address EC" in the e-voting wallet application with n number of votes as the balance, where the n is the number of registered voters. The address is derived from the public key of the election committee as described later.
2) Voters generate their own address (using the same process described later) and make it available to the election committee during the registration. In the above diagram, x is the address generated by voter A's e-voting wallet application.
3) EC "transfers" one vote to each eligible voter's *address*.
4) The blockchain is initialized and transactions are broadcasted into the network

The address generation process can be adopted from the Bitcoin as follows:

- Generate a private key S_K from 256 bits of random data. The existing code libraries such as OpenSSL can be utilized for random number generation and which is input to SHA256 hashing algorithm to obtain a 256-bit private key
- Generate ECDSA public key P_K from the private key S_K using elliptic curve secp256k1 multiplication using the formula $P_K = S_K * G$, where the Generator Point G is taken as a constant from the Elliptic Curve Domain Parameters [22]
  G = 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8
- Hash the public key P_K (0x04,x coordinate, y coordinate) using SHA256; $h_1=H_S(P_K)$
- Hash the previous result again using RIPEMD-160 [27]; $h_2= H_{RIPEMD160}(H_{SHA256}(P_K))$
- Add version byte in front of h_2, this can be taken as the election identifier; $0x00h_2$
- Preform SHA256 hashing twice on the previous result, can copy the first 4 bytes of that value as checksum *c*.
- Add the checksum c at the end of $0x00h_2$ we obtained earlier

- Perform Base58Check encoding [24] on the above result to derive the e-voting address

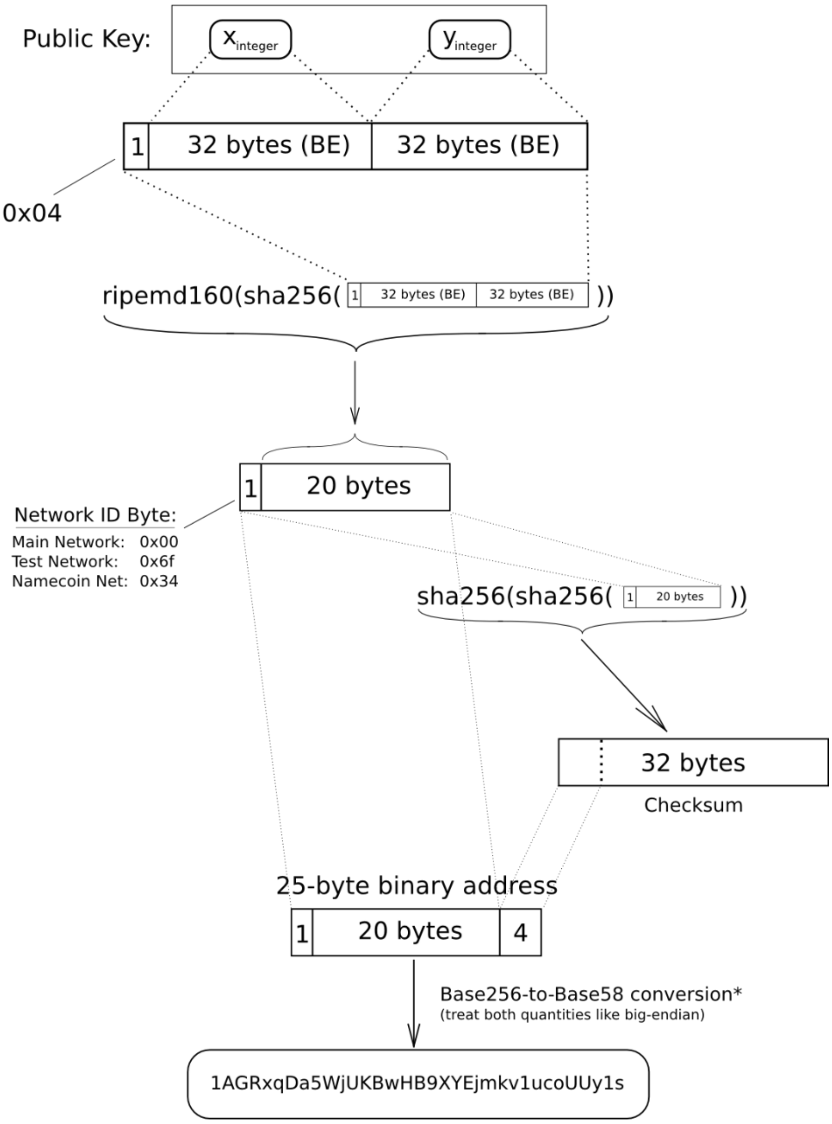


Figure 4.2 Conversion from ECDSA Public key to Address.
Source: bitcoin.it

The transaction is defined as below:

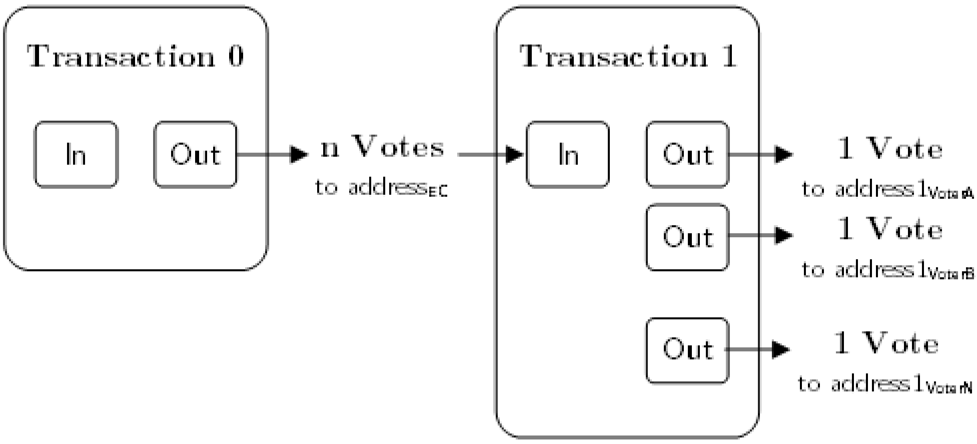


Figure 4.3 Pre-election Transactions

The *Transaction 0* is similar to a *coinbase transaction* in Bitcoin which initially provides n number of votes to the address$_{EC}$ owned by the election committee.

The *Transaction 1* is to transfer *1 vote* to the addresses presented by each eligible voter. The output consists of a script similar to P2PKH script in Bitcoin [28] that allows only the owner of the *address* can spend or cast it further. For example, whoever can present a signature from the private key corresponding to the address1$_{VoterA}$ will be able to transfer it to someone else later. Because only the voter A has the e-voting wallet application with the ECDSA private/public keys corresponding to that address, only the voter A's wallet can present such a signature to redeem this output.

Finally, the blockchain is constructed as below:

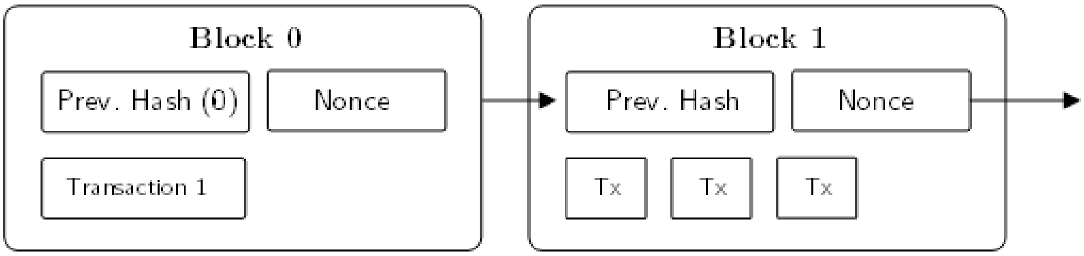


Figure 4.4 The e-Voting Blockchain

We implement a timestamp network with proof-of-work by grouping a transaction that takes place during a defined time interval into a single block. The block also contains a nonce, which is incremented until a SHA-256 hash value is found that has the required number of leading zero bits, which provides the *proof-of-work*. The non-deterministic nature of the hash ensures that there is no other way to obtain such value without calculating the hash again and again while changing the nonce. Once a hash is found that meets the requirement, the block is added to the blockchain and later blocks are added after it. The block cannot be chained without recalculating the hash, which would be difficult as the blockchain is continuously growing.

The stakeholders who are interested in ensuring the integrity of the election can take part in the *mining* process, which adds transactions to the blockchain after performing the *proof-of-work*. The interested parties may include the election commission, voluntary organizations, candidates, and voters.

4.4. Election

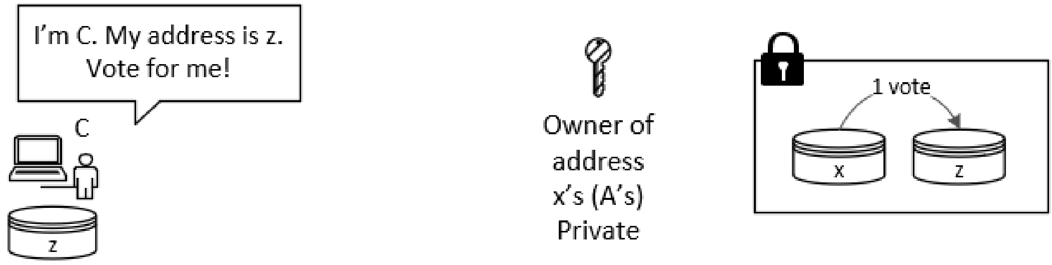

I'm C. My address is z. Vote for me!
C
z
Owner of address x's (A's) Private
1 vote
x
z

Figure 4.5 The Election

The transactions performed during the election stage is given below:

1) Candidate generates his address and present it to the Election committee.
2) The Election Committee will publish candidate information. Also, the candidate addresses will be pushed to the e-wallet application, which will provide convenient method for voters to transfer their vote.
3) Voters “transfer” their vote to the address of their preferred candidate and broadcast it to the network.
4) Any vote transfer to an address that does not belong to a candidate will be considered as an invalid vote. The e-wallet application will provide “discard

vote" option in the e-wallet application which will generate a random address and transfer the vote to that address. A voter who is not interested in voting can discard their vote by selecting that option.

5) The transactions are grouped, added to a block, and eventually will be added to the blockchain after performing the proof-of-work.

The address generation and vote transferring will follow the exact same steps described in the pre-election setup.

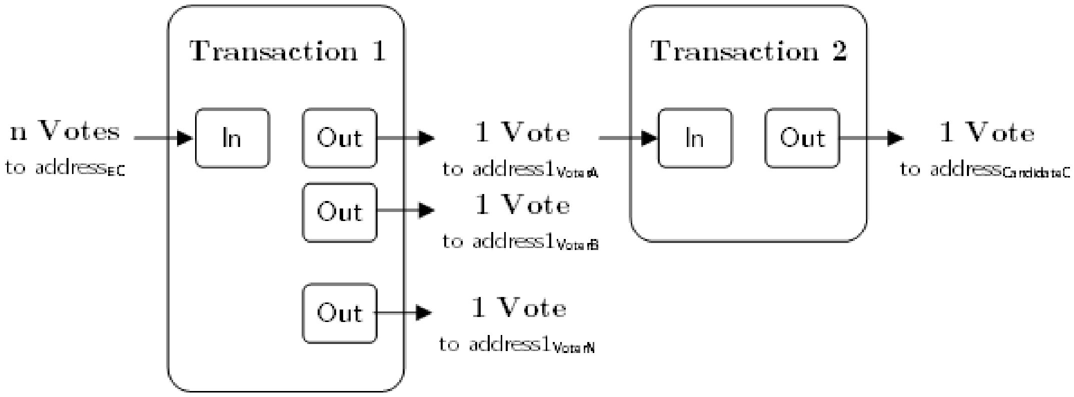
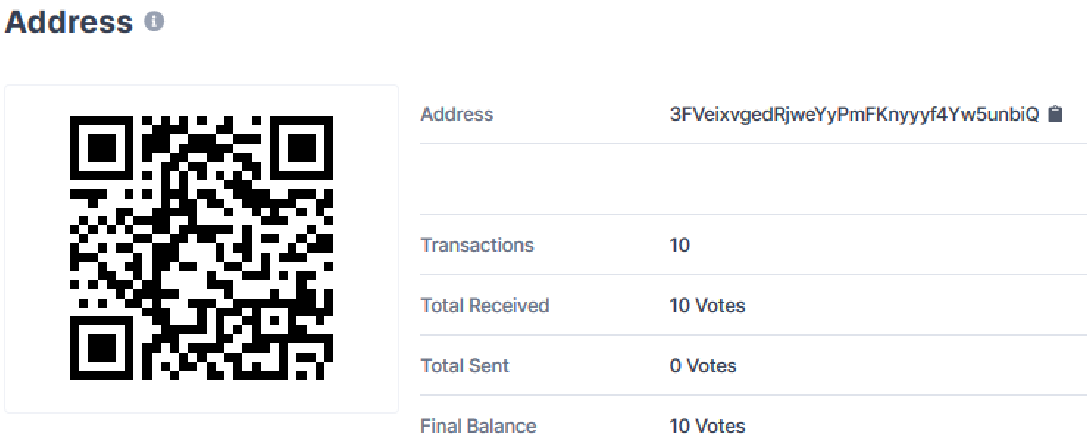


Figure 4.6 The Voting Transactions

4.5. Tabulation



Address

Address	3FVeixvgedRjweYyPmFKnyyyf4Yw5unbiQ
Transactions	10
Total Received	10 Votes
Total Sent	0 Votes
Final Balance	10 Votes

Figure 4.7 Sample Vote Balance Recorded to a Candidate's Address

The tabulation process is given below:

1) Check the final votes recorded in the ledger against the addresses of the candidates
2) Candidates present their address to the election committee with their identity and address ownership information
3) The election committee officially finalize the result, while everyone else can see the vote balances in the ledger.

4.6. System Architecture

The below diagram summarizes the proposed Bitcoin based e-voting system and its transactions. The initial step (i) is the voter registration by the Election Committee.

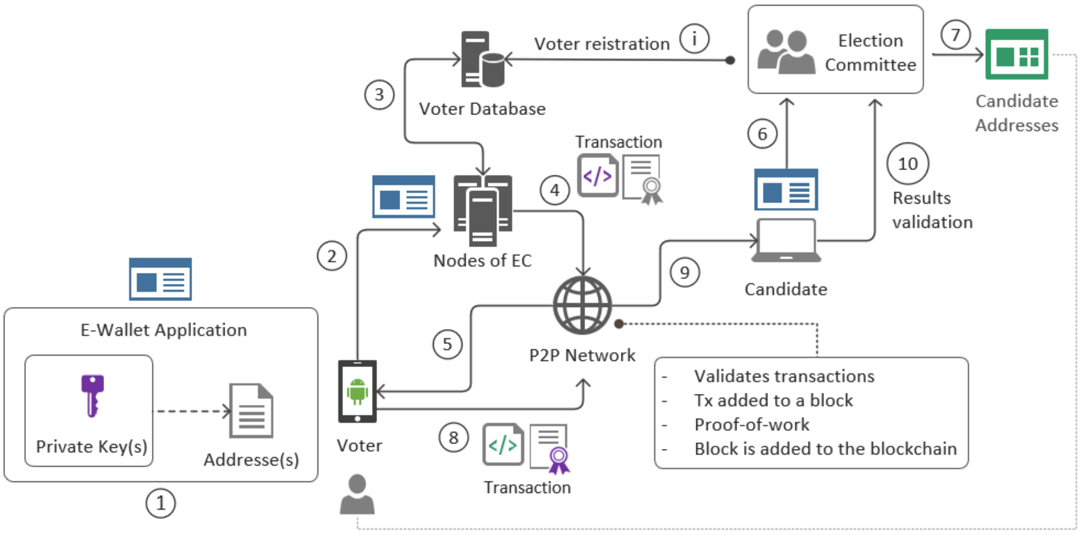


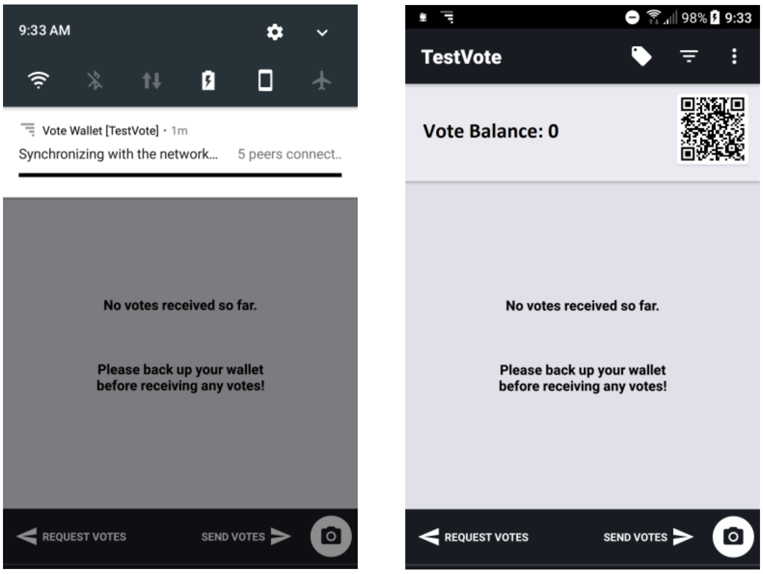
Figure 4.8 Bitcoin Based e-Voting System Architecture

1. Voter generates private key and corresponding Bitcoin address using the e-wallet application installed in his mobile/computer
2. Voter presents his address to the Election Committee
3. Election Committee verifies the voter eligibility
4. Election Committee transfer vote to voter’s address and broadcast it to the peer-to-peer (P2P) network

5. P2P network validates transaction and add it to the blockchain after performing the proof of work. The vote balance is reflected in the voter's e-wallet application after the successful confirmation.
6. Candidate generates his address and present it to the Election committee.
7. The Election Committee will publish candidate information. Also, the candidate addressed will be pushed to the e-wallet application, which will provide convenient method for voters to transfer their vote.
8. Voter transfers the vote to a preferred candidate
9. P2P network validates transaction and add it to the blockchain after performing the proof of work
10. Candidate present his vote balance to the Election Committee for validation and publishing.

4.7. Proof of Work Implementation

An Android application was developed utilizing the reusable functions/code segments intended for Bitcoin developers [29]. The Testnet and its peer-to-peer network is used as the testing environment. Testnet is a substitute Bitcoin blockchain, that allows programmers or bitcoin testers to test their applications, without needing to use actual bitcoins or compromising the main bitcoin blockchain. The Elections Committee's and Voter's applications are installed and tested on Android devices, HTC One A9 and Samsung Galaxy S10e.


9:33 AM

Vote Wallet [TestVote] · 1m

Synchronizing with the network... 5 peers connect..

No votes received so far.

Please back up your wallet before receiving any votes!

98% 9:33

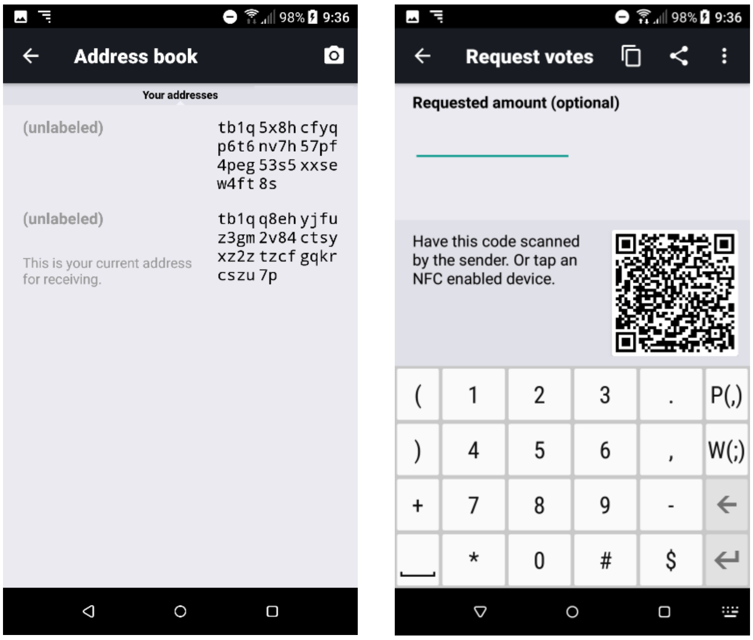
TestVote

Vote Balance: 0

No votes received so far.

Please back up your wallet before receiving any votes!

Figure 4.9 App syncs with the network(left) The initial vote balance is 0 (right)



Your addresses

(unlabeled) tb1q 5x8h cfyq p6t6 nv7h 57pf 4peg 53s5 xxse w4ft 8s

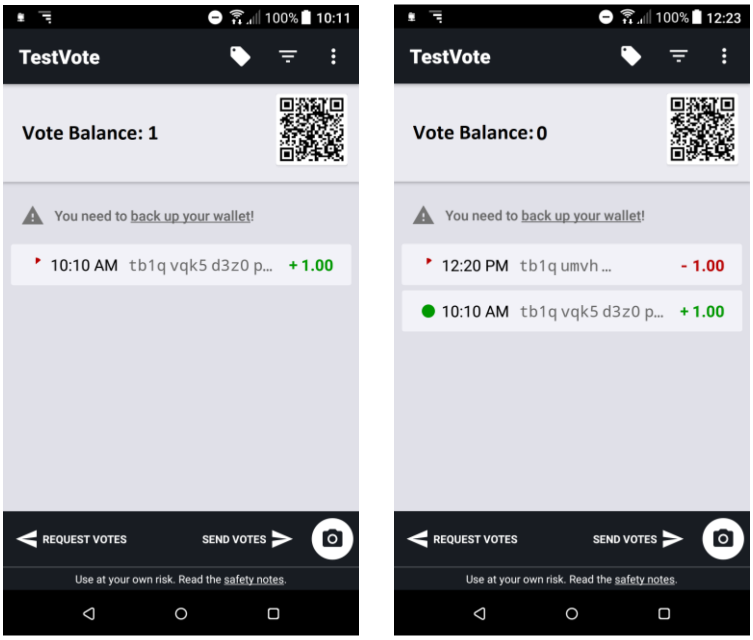
(unlabeled) tb1q q8eh yjfu z3gm 2v84 ctsy xz2z tzcf gqkr cszu 7p

This is your current address for receiving.

Requested amount (optional)

Have this code scanned by the sender. Or tap an NFC enabled device.

Figure 4.10 Voter can request votes from the Election Committee by presenting the address. Note that a voter can generate multiple address



Vote Balance: 1

You need to back up your wallet!

10:10 AM tb1q vqk5 d3z0 p... + 1.00

Use at your own risk. Read the safety notes.

Vote Balance: 0

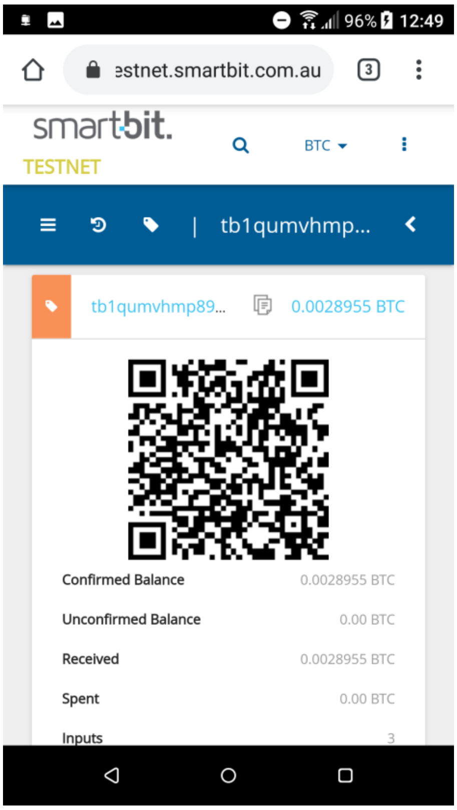
You need to back up your wallet!

12:20 PM tb1q umvh ... - 1.00

10:10 AM tb1q vqk5 d3z0 p... + 1.00

Use at your own risk. Read the safety notes.

Figure 4.11 The spendable vote balance is reflected in the voter's wallet (left) One vote is sent to the candidate's address (right)



Confirmed Balance 0.0028955 BTC

Unconfirmed Balance 0.00 BTC

Received 0.0028955 BTC

Spent 0.00 BTC

Inputs 3

Figure 4.12 Confirmed vote total recorded to candidate's address (in Testnet – the alternative Bitcoin blockchain used for testing)

4.8. Summary

The Bitcoin based e-voting system has been broken down into three stages based on the phases of the voting process: pre-election, election, and the post-election tabulation. The implementation details have been provided for each phase while the general assumptions set the scope and boundary of the e-voting system.

The key components of the e-voting system, address generation and Bitcoin based transaction process has been described in detail in the pre-election section. Since the same address generation and transaction processes are followed in the election phase. All the minor implementation details and application development process could be adopted from the Bitcoin Developer guide [30] which provide examples codes on how to develop applications for Bitcoin.

Chapter 5

System Evaluation and Performance Review

5.1. Introduction

In this chapter, we discuss how strongly various security and performance requirements are being met in the proposed Bitcoin based e-voting system. We have taken both the system architecture and the protocols and standards used in the system for the evaluation.

5.2. Evaluation of Security Requirements of an e-Voting System

5.2.1. Authorization

Authorization ensures that only the authorized voters can vote. In the proposed e-voting system, the election committee ensures that votes are transferred to only the addresses presented by authorized voters. The all the transactions made to the voters' addresses are recorded in the public ledger.

5.2.2. Anonymity

We can verify the anonymity requirement by taking the voter A and B and their corresponding addresses X and Y as examples. The address X is self-generated by A.

However, A has shared his address with the EC, therefore, only A and EC know that A has the address X. However, to protect his identity during the election, A can decide to generate a new address when he is casting the vote. None of the other voters and candidates have a way to find the owner of the new address, therefore the anonymity requirement is met among the voters and the candidates.

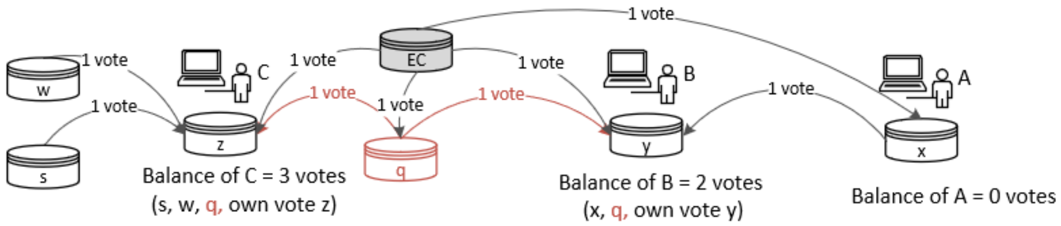


Figure 5.1 Anonymity Verification

Since all the voting transactions are distributed in the peer-to-peer network, it is possible to observe that there are transaction to the address *q*, but there is no concept called ‘from address’ in the e-voting system which operates similar to the Bitcoin.

5.2.3. Tamper-resistance

Only the eligible voters will receive exactly one vote attempt to their address by the election committee during the pre-election setup, that can be casted to the preferred candidate in the next step. Therefore, without knowing the private key of the election committee SK_V, it is not possible for a voter to have more than one vote in the e-voting application.

5.2.4. Restriction of attempts

The Bitcoin-like systems timestamps the transactions by hashing them and attaching them into an ongoing chain of hash-based proof-of-work. This forms a record that is unchangeable without doing the proof-of-work again, hence providing an effective solution against double spending. The same approach is adopted in the e-voting system to prevent a voter from voting more than once. The first timestamped transaction recorded in the ledger will be considered as the valid attempt by the e-voting system.

5.2.5. Prevention of impersonation

No one will be able to cast a vote on behalf of someone else (Y) without having access to their private key SK_Y. The proposed system is suitable for conducting a medium scale election where we assume that no one will deliberately reveal his private key to others.

5.2.6. Individual Verifiability

The transaction ID, the identifier used to uniquely identify a given transaction is used for the individual verifiability. The transaction ID is generated by hashing the transaction twice by SHA256: SHA256(SHA256(transaction)). The transaction ID is known to the voter and he can later check the ledger to verify that his vote transaction is recorded correctly.

5.2.7. Universal Verifiability

Since all the verified transactions are recorded in a shared public ledger. The ledger contains all the transactions ever processed and allow the e-voting application to verify the validity of each and every transaction.

5.3. Evaluation of the Protocols and Algorithms

5.3.1. Random Number Generation

The e-voting system that we have proposed uses ECDSA key-pairs. The e-voting client could use a toolkit created by The OpenSSL Project for generating these keys. Since the use of those keys is the basis of all operations on the proposed e-voting system, it is crucial that the generation of an identical key is impossible. The OpenSSL random generator implements a *cryptographically secure pseudo-random number generator* that takes the current time in microseconds and GUI events dev/urandom on Unix, and HKEY_PERFORMANCE_DATA on Windows operating system as the seed. Even though an earlier OpenSSL library contained a vulnerability related to Pseudo Random Number Generator (PRNG) in Debian system [31], which was fixed later on, we can consider it as a secure implementation as there are no other reported vulnerabilities to date regards of random number generation.

5.3.2 Elliptic Curve Digital Signature Algorithm and secp256k1 Curve

The proposed e-voting system uses the Elliptic Curve Digital Signature Algorithm based on the secp256k1 Koblitz Curve that is defined in SEC 2 [22]. secp256k1 defines the parameters of the elliptic curve that is used in the public key cryptography of the e-voting system. It was directly adapted from the well proven Bitcoin implementation.

The Elliptic Curve Cryptography (ECC) is considered to be more efficient in terms of CPU and memory utilization that uses shorter keys to provide equivalent security of RSA. Also, unlike the commonly implemented curves in ECC, secp256k1 implementation allows efficient computation. It is observed that a sufficiently optimized implementation of the curve is 30% faster than other curves [32]. Similar implementations of e-voting systems demonstrates that ECC based scheme can outperform the traditional hybrid symmetric and asymmetric cryptographic scheme in the context of mobile e-voting environment [33].

The secp256k1 curve used by ECDSA has not used widely before it is used by Bitcoin and is not described by the ANSI X9.62 or the NIST standards, where that is not widely analysed. The less popularity of the particular curve other than in the Bitcoin can be viewed as both an advantage and a disadvantage. Any curve-specific exploits should have to be developed targeting specifically these systems - Bitcoin or the proposed e-voting system. This prevents any attempts at finding such weakness in much popular curves affecting the e-voting system. However, there are no such exploits is the curve to date, which would also have compromised the Bitcoin cryptocurrency system. Also, the secp256k1's constants are selected in a predictable way, unlike in the popular NIST curves, which reduces the possibility that the curve's creator inserted any backdoor into the curve.

5.3.3 Private Key Management

While it is possible to store the private keys in the e-wallet application running in the voter’s mobile or the computer, it is recommended to store the keys in a hardware wallet for better security. Many modern computers and mobile phones are equipped with secure storage or have the support for hardware security modules (HSM) to store and manage the keys.

5.4. Evaluation of the Performance

The following table summarizes the average times used by the e-voting system to complete transactions at each step. The performance data is specific to the motioned mobile device platforms an provide an indication of the performace. It is required to run the e-voting application on multiple mobile and computer platforms to further validate the result.

Table 5.1 Performance Data of the e-Voting System

Transaction	Time	Device Platform
Average time to send votes to the voter by Election Committee (step 4)	16.5 s	Samsung S10e CPU: Octa core 2.27 GHz x 2, 2.31 GHz x 2, 1.95 GHz x 4 cores, Memory: 6 GB
Average time to appear the votes in voter's wallet application after confirmations (step 5)	10 mins	P2P network with 5 nodes
Average time to send votes to the candidate by voter (step 8)	19.5 s	HTC One A9 CPU: Octa Core 1.5GHz x 4, 1.2 GHz x 4, Memory: 3 GB
Average time to appear the votes in candidate's wallet application after confirmations (step 9)	10 mins	P2P network with 5 nodes

5.5. Summary

The implementation of proposed e-voting system is mostly identical to the well-established Bitcoin e-cash system. The Bitcoin system has evolved over a decade and has proven that it is secure and protected to a satisfactory level against malware and hackers. Therefore, we can expect the same level of security and performance in the proposed Bitcoin based e-voting system which is built based on the similar protocols, security algorithms and architecture. The the e-voting application developers shuold always follow the programming and security best practices, conduct valunerability assessments, and properly test the application in a pilot environment before releasing it to the users.

Furthermore, we have shown how the properties and features of an e-voting system to an e-cash system is implemented in the proposed system. We have successfully verified that the Bitcoin based e-voting system provides the features; authorization, anonymity, tamper-resistance, restriction of attempts, individual verifiability and finally the universal verifiability.

Chapter 6

Conclusions

To provide effective solution for the verifiability of an electronic voting, in this paper, we have proposed a Bitcoin based secure electronic voting system. As the methodology for designing and implementing the e-voting system, we have taken an existing Bitcoin based implementation, Bitcoin, which is a well-proven, robust, and scalable e-cash system.

The analysis of the existing electronic voting systems has suggested that authorization, anonymity, tamper-resistance, restriction of attempts, prevention of impersonation, individual and the universal verifiability are the commonly expected features of an electronic voting system. Most of these features, more importantly, the anonymity and verifiability features were readily available in the Bitcoin, which we could directly implement in our electronic voting system. The remaining features of the e-voting system were either adopted from Bitcoin to match with the requirements of the e-voting system or newly implemented.

Finally, we have proven that the proposed system meets the identified requirements of an electronic voting system. Since the architecture and protocol level implementation of the Bitcoin based e-voting system is predominantly similar to the implementation of Bitcoin, we could expect the same level of security and performance in the proposed e-voting system.

We suggest that the proposed system to be used in the elections where it is assumed that coercion and vote selling would not take place. Further security and legal measures should be taken place to avoid such misconducts in a large-scale election. The future research on preventing coercion and vote selling without an involvement of a legal body, would extend this research work.

References

[1] K. Kohno, A. Stubblefield, A. D. Rubin and D. S. Wallach, "Analysis of an Electronic Voting System," *Proc. IEEE Symposium on Security and Privacy,* pp. 27-42, 2014.

[2] A. D. Rubin, "Security considerations for remote electronic voting," *Communications of the ACM,* vol. 45, no. 12, pp. 39-44, 2002.

[3] B. Schneier, Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C, New York: John Wiley & Sons, Inc, 1996.

[4] Caltech/MIT Voting Technology Project, "Voting: What Has Changed, What Hasn't, & What Needs Improvement (2012)," October 2012. [Online]. Available: http://www.vote.caltech.edu/reports/.

[5] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," October 2008. [Online]. Available: https://www.bitcoin.org.

[6] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine and J. A. Halderman, "Security Analysis of the Estonian Internet Voting System," in *CCS '14 Proc. ACM SIGSAC Conference on Computer and Communications Security*, New York, 2014.

[7] S. Wolchok, E. Wustrow, D. Isabel and J. A. Halderman, "Attacking the Washington, D.C. Internet Voting System," in *Proc. 16th Conference on Financial Cryptography & Data Security*, Kralendijk, 2012.

[8] K. W. a. R. T. A. Villafiorita, "Development, Formal Verification, and Evaluation of an E-Voting System With VVPAT," *IEEE Transactions on Information Forensics and Security,* vol. 4, no. 4, pp. 651-661, 2009.

[9] "H.R.3295 - Help America Vote Act of 2002," 29 10 2002. [Online]. Available: https://www.congress.gov/bill/107th-congress/house-bill/3295/.

[10] fec.gov , "Direct Recording Electronic (DRE)," [Online]. Available: http://www.fec.gov/pages/dre.htm.

[11] P. Y. A. Ryan, D. Bismark , J. Heather, S. A. Schneider and Z. Xia, "Prêt à voter: a voter-verifiable voting system," *IEEE Transactions on Information Forensics and Security,* vol. 4, no. 4, 2009.

[12] D. Chaum, R. T. Carback and J. Clark, "Scantegrity II: End-to-End Verifiability by Voters of Optical Scan Elections Through Confirmation Codes," *IEEE Transactions on Information Forensics and Security,* vol. 4, no. 4, pp. 611-627, Dec. 2009.

[13] R. L. Rivest, "On the notion of 'software independence'," *Phil. Trans. R. Soc. A,* vol. 366, p. 3759–3767, 2008.

[14] D. Wagner, "Voting Systems Audit Log Study," Report commissioned by the California Secretary of State, Berkeley, 2010.

[15] L. Barlow, "An Introduction to Electronic Voting," November 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.136.2993.

[16] "THE FUTURE OF E-VOTING," *IADIS International Journal on Computer Science and Information Systems ,* vol. 12, no. 2, pp. 148-165, 2017.

[17] . J. A. Feldman, J. A. Halderman and E. W. Felten, "Security analysis of the diebold AccuVote-TS voting machine," *EVT'07: Proceedings of the USENIX Workshop on Accurate Electronic Voting Technology,* 2007.

[18] P. D. DeVries, "An Analysis of Cryptocurrency, Bitcoin, and the Future," *International Journal of Business Management and Commerce,* vol. 1, no. 2, 2016.

[19] F. Liu, X. Li and G. Gao, "The Design of an e-cash System," *International Conference On Computer Design and Applications,* 2010.

[20] Y. Baseri, J. Mohajeri and B. Takhtaei, "Secure untraceable off-line electronic cash system," *Scientia Iranica,* vol. 20, no. 3, pp. 637-646, 2013.

[21] bitcoin.org, "Transactions Guide - Bitcoin," 2017. [Online]. Available: https://bitcoin.org/en/transactions-guide. [Accessed 1 March 2017].

[22] Certicom Research, "Standards for Efficient Cryptography Group," [Online]. Available: http://www.secg.org/. [Accessed 1 March 2017].

[23] Certicom Corp, "Standards for Efficient Cryptography 2 (SEC 2)," 27 January 2010. [Online]. Available: http://www.secg.org/sec2-v2.pdf.

[24] "Base58Check encoding - Bitcoin Wiki," 27 November 2017. [Online]. Available: https://en.bitcoin.it/wiki/Base58Check_encoding#Background.

[25] "Pubkey Script, ScriptPubKey - Bitcoin Glossary," [Online]. Available: https://bitcoin.org/en/glossary/pubkey-script. [Accessed 1 March 2017].

[26] E. Abu-Shanab, M. Knight and H. Refai, "E-voting systems: A tool for e-democracy," *anagement Research and Practice,* vol. 2, pp. 264-274, 2010.

[27] "RIPEMD-160," 30 June 2014. [Online]. Available: https://en.bitcoin.it/wiki/RIPEMD-160. [Accessed 1 March 2017].

[28] Bitcoin Project, "Transactions — Bitcoin," [Online]. Available: https://developer.bitcoin.org/devguide/transactions#pay-to-public-key-hash-p2pkh. [Accessed 1 March 2017].

[29] "Bitcoin-wallet," [Online]. Available: https://github.com/bitcoin-wallet/bitcoin-wallet. [Accessed 30 June 2017].

[30] "Developer Guides — Bitcoin," [Online]. Available: https://developer.bitcoin.org/devguide/. [Accessed 31 March 2017].

[31] "Debian/Ubuntu OpenSSL Package Random Number Generator Weakness," [Online]. Available: https://knowledge.digicert.com/solution/SO9094.html. [Accessed 1 March 2017].

[32] "Secp256k1," [Online]. Available: https://en.bitcoin.it/wiki/Secp256k1. [Accessed 1 March 2017].

[33] T. Ahmad, J. Hu and S. Han, "An Efficient Mobile Voting System Security Scheme Based on Elliptic Curve Cryptography," *2009 Third International Conference on Network and System Security,* pp. 474-479, 2009.

[34] D. W. Jones, "A Brief Illustrated History of Voting," The University of Iowa Department of Computer Science, 2003. [Online]. Available: http://homepage.divms.uiowa.edu/~jones/voting/pictures/.

[35] A. Ekblaw, A. Azaria, J. D. Halamka and A. Lippman, "A Case Study for Blockchain in Healthcare: “MedRec” prototype for electronic health records and medical research data," MIT Media Lab, Beth Israel Deaconess Medical Center, 2016.

[36] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Ethereum & Ethcore, 2017.