A Blockchain-Based Approach to Developing Scalable and Auditable E-Voting Systems Without Requiring a Trustworthy Central Authority

by

Samuel Marple

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2021 by the
Graduate Supervisory Committee:

Sik-Sang Yau, Chair
Dijiang Huang
Ni Trieu

ARIZONA STATE UNIVERSITY

August 2021

ABSTRACT

The purpose of an election is for the voice of the voters to be heard. All the participants in an election must be able to trust that the result of an election is actually the opinion of the people, unaltered by anything or anyone that may be trying to sway the vote. In the voting process, any “black boxes” or secrets can lead to mistrust in the system. In this thesis, an approach is developed for an electronic voting framework that is transparent, auditable, and scalable, making it trustworthy and usable for a wide-scale election. Based on my analysis, linkable ring signatures are utilized in order to preserve voter privacy while ensuring that a corrupt authenticating authority could not sway the vote. A hierarchical blockchain framework is presented to make ring signatures a viable signature scheme even when working with large populations. The solution is evaluated for compliance with secure voting requirements and scalability.

ACKNOWLEDGEMENTS

I wish to express my gratitude for all those that have participated in the journey of this thesis. I want to recognize my committee chair, Professor Stephen Yau, for making himself available at all hours to give of his time to help me with this research. His insight helped greatly in the development of my research but more importantly helped in the development my character. I also want to thank my committee members, Professors Dijiang Huang and Ni Trieu, for giving their time and input to help finalize this thesis.

I recognize the financial support of the National Science Foundation’s CyberCorps: Scholarship for Service and appreciate the blessing it has been to me during my college career.

I also wish to thank my family and friends for listening to my ideas and supporting my efforts in this work.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

I. Introduction

Democracy, by definition, is rule by the people [1]. With the almost daily advancement of technology seen in the world today, democratic processes are under greater threat from cyber-attacks and manipulation than ever before. When dealing with a political election, these threats come from everywhere. Many come from outside in the form of enemy nation-states or lone foreign actors, but perhaps more dangerous are the threats from within: disgruntled citizens, opposing political parties, or the government itself. In order to combat these threats, a secure voting system must be established to ensure that no malicious actor can manipulate an election.

As critical as security in an election process is the trustworthiness of the results of the election. Belief that an election will not reflect the actual results will discourage voter turnout and will sow discord after the election when defeated parties claim the results are inaccurate [2]. In order for everyone to trust the legitimacy of an election, a transparent and fully auditable voting system is required.

Transparency is required during the entire election process if the system is to be considered trustworthy. Many voting systems attempt auditability but consider it a violation of privacy if all steps of the process are revealed to the general public. The step of verifying voter eligibility, for example, may be kept private to preserve a voter's personal identifiable information (PII) from being published. This becomes a problem when anonymous ballots claim eligibility but cannot be linked to voters for the sake of privacy, allowing a corrupt authenticating authority to grant eligibility to ineligible ballots or voters.

In the context of e-voting systems, a central authority (CA) is a single entity that is in charge of an election and controls at least a portion of that election. For example, a central authority could be a state government, a student body election committee, or the CEO of a company. Central authorities exist to set and enforce the rules of an election, such as determining the voting period and how to ensure that only eligible voters participate. A central authority is inherently an important component of any election because it is the entity that brings all the voters together. For whose benefit would an election be without some larger entity to act on the outcome? Without a government, for instance, how would results of any political election be enforced?

However, even while recognizing their importance, caution must be exercised when involving any central authority in the voting process. A corrupt CA often has the capability of altering the results of an election if it so chooses. Even trustworthy CAs can create a single point of failure which a malicious actor could penetrate in order to wreak havoc on an election.

In many modern voting systems, the central authority runs the election, determines which voters can participate, counts the ballots, and announces the outcome; the voters have to trust that the announced results are accurate. This makes an enormous assumption that the central authority is incorruptible. In order for election results to be more trustworthy, something in this process needs to change.

At first glance, it may seem prudent to remove any controlling central authorities from an election in order to increase trustworthiness. The trustworthiness of a computer network, for example, increases with the decentralization and distribution of that network [3]. Without a central authority, however, elections lose much of their utility and security [4].

This central authority is the link between the voters; it is the reason all the voters are contributing input on the same issue.

For the same reason, the central authority is often in charge of authenticating its voters and making sure that only those that belong to the group are allowed to vote. If involved in the authentication process, a corrupt central authority would have the opportunity to manipulate an election by validating ineligible votes, invalidating legitimate ones, casting otherwise uncast ballots, or even changing votes. This is the cause of voter distrust in election processes that involve secret-keeping central authorities. For the results of an election to be trusted by the voters, any central authority must be watched over by the voters themselves. Since the presence of a central authority is required, the following research does not eliminate it, rather it creates a way for the central authority to authenticate voters without keeping any secrets.

This manuscript will present a scalable electronic voting approach that increases auditability without sacrificing voter privacy and without requiring trust in a central authority. It will include a hierarchical blockchain framework that allows for linkable ring signatures to be used realistically at a large scale while storing votes securely. The improvement presented in this research is the scalability of an auditable e-voting system without a requirement to trust that the election's managing authority is incorruptible. The application of linkable ring signatures guarantees a fair election since all participants will be able to audit each vote for eligibility. No other proposed e-voting system, to the knowledge of the author, has both the characteristics of security and scalability that the following approach does.

The scope of this research will not include implementation details specific to any country or organization, although the democratic process of the United States will be used as an example. The remainder of the paper will proceed as follows: background, current state-of-the-art approaches, an overview of the proposed solution, a highlight of the innovation presented in this solution, an illustrative example, an evaluation of the research, conclusions drawn, and future work to be considered.

II. Background

In order to evaluate the e-voting approach that will be presented hereafter, it is important to understand several key terms that will be defined in this section.

A. Secure Voting

In their paper, Hardwick et al. lay out five principles of a secure voting protocol [5]. These include fairness, eligibility, privacy, verifiability, and coercion-resistance. For this approach, additional principles of secure voting are added: scalability and immutability.

1) Scalability: The electronic voting system is viable for use in large-scale applications, such as regional or national elections. Inefficient and theoretical approaches exist which do not improve the security of e-voting systems because they are not legitimately usable in an actual voting situation.

2) Immutability: Ballots are stored in a secure manner that makes them difficult to change. Once votes are cast, no entity has the ability to alter the record. Immutability is not exclusive to error-correction, rather errors are corrected by adding more to the record instead of deleting and replacing data.

3) Fairness: No results of the election should be available before the voting period has concluded. This keeps later voters from being influenced by the decision of previous voters. This could be a problem if, for example, results show an obvious landslide and voters decide not to cast their ballots because it seems futile.

4) Eligibility: Only authorized voters are eligible to cast their vote. In a political election, eligibility is usually dictated by laws based on an individual's criminal history, place of residence, etc. Voters must reliably prove their identity to a central authority in order to receive the authorization to vote. The principle of eligibility is often a question of authentication methods since the central authority must be sure that the voter is the individual he claims to be.

5) Privacy: No particular votes can be attributed to any particular voter. This is often referred to as a “secret ballot” and makes sure voters make decisions without worrying about external consequences.

6) Verifiability: There are two types of verifiability – individual and universal. These are key to a trustworthy voting process.

a) Individual Verifiability: Any voter can ensure that their vote was counted the way in which it was cast.

b) Universal Verifiability: Anyone can ensure that all the votes are legitimate and that they were tallied correctly.

7) Coercion-Resistance: An entity trying to influence voters will have no way of knowing if voters were influenced correctly. While this may sound similar to privacy, this principle is often a tug-of-war between privacy and individual verifiability. With complete privacy, no one would be able to tie an individual to their vote, but then the voter would not be able to verify that their vote was recorded correctly. On the other hand, an individual with the ability to verify their own vote might be able to share that ability with others, allowing coercion or vote-selling. An element of coercion-resistance is receipt-freeness – that the system does not provide a receipt to link the voter to his vote [6]. Forgiveness, or the ability to re-cast one’s ballot, can be seen as a lesser form of coercion-resistance [5].

The principles of scalability, immutability, fairness, eligibility, privacy, and verifiability will be set as requirements of a secure voting system, while coercion-resistance will be preferable but not required. This is due to the conflict between individual verifiability and coercion-resistance, with the priority being placed in this instance on the former.

Compliance with each of these secure voting principles will ensure the security of a voting system. Individual verifiability, for example, defends against man-in-the-middle attacks and data corruption of individual votes. Universal verifiability protects a system from improper tally calculation by an adversary changing code to sway the totals. Fairness prevents anyone, including the central authority, from taking action to change the results if votes appear to be trending in a direction contrary to their desires. The verification of voter eligibility prevents intrusion in the system and invalid votes from being counted. The approach presented in this thesis improves on the security of e-voting systems by more completely adhering to all of these principles than previous approaches.

B. Blockchain

The votes in this approach will be stored on a blockchain. A blockchain is a ledger maintained in common among a decentralized network of nodes. It was first implemented in 2008 when an article by Satoshi Nakamoto was published describing a new type of currency known as Bitcoin [7]. When an addition is made to a blockchain ledger, the entire network must come to a consensus in order to add the new block to the chain. Cryptographic primitives make it difficult to alter the history of a blockchain, and combined with the decentralized nature of the network, it is nearly impossible to change the ledger retroactively. For this reason, it has been a topic of much research in relation to e-voting [6].

C. E-Voting Process

The life cycle of an e-voting system can be broken up into three phases: setup, voting, and tallying. Because an election should have a defined start and end time to accept ballots, the phases setup, voting, and tallying fall chronologically before, during, and after the window for voters to cast their votes, respectively. While the contents of each phase vary widely depending on the e-voting approach being utilized, the general outline of these phases is as follows.

1) Setup: During this phase, everything is prepared in advance of the voting period. At this point, all procedures for the election are decided, including the voting period, the format in which votes should be submitted, and the manner of verifying the eligibility of voters. The environment for ballot submission is created. The voting population is established and the eligibility of the voters is verified. Finally, the issues and candidates to be voted on are laid out and put on the ballot.

2) Voting: During the specified voting period, voters have the opportunity to view the ballot and specify their decisions. Filled-out ballots are submitted via the environment prepared in the setup phase and the votes are stored until the voting window has closed.

3) Tallying: All votes are totaled. These tallies, especially the winners, are reported.

D. Signature Schemes

Since the creation of electronic voting systems, there have been many ways to prove an eligible ballot while trying to maintain voter privacy. In this section, a variety of these methods will be mentioned.

1) Token-Based Systems

With the primary application of blockchain being cryptocurrencies, a natural next step would be to use virtual coins for other purposes, like voting. In the early stages of e-voting, coin-based election systems were common [4, 6, 8]. Simply put, every voter gets a unit of currency, or token, and sends it to the wallet of the candidate or resolution of their choice. After the deadline, the wallet with the most tokens wins.

One of the main problems with token-based voting systems is the lack of voter privacy. The ability to track the life of a unit of cryptocurrency is vital to the function of digital money but is not a desired property of e-voting tokens. While anonymity is possible, tokens are easily tracked by design. To only allocate votes to eligible voters, a central authority would have to send tokens to the wallet or account of each eligible voter, which means that the central authority would know the account of each voter and be able to track the tokens to their final destinations, the candidates. In such a system, there is no voter privacy. In order to maintain voter privacy, a central authority must be able to authenticate voters without knowing how each voter casts their ballot. Blind signatures fill this need.

2) Blind Signature Schemes

Blind signatures are commonly used in e-voting systems to preserve anonymity while proving voter eligibility [6, 9]. However, this requires a trusted central authority to maintain voter registers and sign electronic ballots in secret, giving opportunity for election manipulation by the central authority or malicious actors with access to central authority records.

A blind signature can be thought of as a document inside a carbon paper-lined envelope [9]. A voter can put their ballot inside the envelope, labeled with their name on the outside and any other form of identification required, and mail it to the signing authority. The central authority would sign the outside of the envelope to legitimize the ballot and return it to the voter. The voter would then remove the ballot from the envelope, and it would still have the authority’s signature on it from the carbon paper. This ballot would be submitted to be counted without any forms of identification attached. The ballot would be provably legitimate due to the presence of the central authority’s signature, but the vote would still be anonymous.

In the digital realm, this is done by encrypting and digitally signing files. A ballot can be encrypted and then sent to the central authority along with proof of the voter’s identity. Because the vote is encrypted, the central authority does not know how the voter is voting. Once the identity of the voter has been authenticated, the central authority signs the encrypted vote and returns it to the voter. The voter can then decrypt the vote and submit it for tallying without any identification attached. The digital signature remains on the ballot – proving eligibility – but privacy is maintained.

An obstacle with blind signatures is the amount of trust it places in the central authority. The central authority is a single point of failure for the system, so if the authority is breached, broken, or corrupt, the whole system fails. For example, a corrupt signing authority would be able to track which voters have already voted and cast any uncast votes at the end of an election without detection [10].

3) Multi-Blind Signatures

With a multi-blind signature scheme, a group of authenticating authorities must unanimously approve each ballot [11]. Initially, it seems that this solves the problem with the single blind signature scheme of a corrupt central authority being able to cast any uncast votes without detection. Upon further inspection, this does not actually protect against corrupt central authorities. The signing authorities still maintain secrets of who they authenticate, and if the authorities collude to cast votes together, it becomes effectively the same as a single blind signature scheme. Arguments have been made to ensure that the authenticators would never work together, as opposing parties in a political election [10]. This legitimately keeps any of the central authorities from faking ballots. However, it may not keep CAs from discarding the votes for another party. In addition to the characteristics of the scheme failing to protect against corrupt central authorities, the complexity introduced into the system when requiring multiple signatures may present a challenge.

4) Ring Signatures

While not immune to scalability issues, ring signatures provide a much better solution to the corrupt central authority problem. Originally designed to leak secrets [12], ring signatures prove membership in a group of people without revealing which individual member of the group provided the signature. The only requirements to create a ring signature are the public keys of the entire group and the private key of a single member.

At its inception, the goal of ring signatures was to prove authority on a subject without exposing one's own identity. For example, if a member of a corporate board wants to publish digital evidence of corruption, he signs the documents with the public key of each board member, including his own, and his own private key. This will prove to any viewer that the documents came from a member of the board but will not reveal which member of the board they came from. The same can be easily applied to an e-voting scheme. With the public keys of a voter population, a ring signature could be applied to a ballot that validates the legitimacy of the ballot while keeping the identity of the voter a secret.

This scheme alone has its own limitations. It is not very scalable, so with an increase in the members of a ring, the computation time of creating and validating the signature will increase significantly. It also has no defense against the double-voting problem – the ability of a single voter to cast more than one ballot. In order to solve this problem, linkable ring signatures can be used.

5) Linkable Ring Signatures

Linkable ring signatures (LRS) are a version of the ring signature scheme that allows multiple messages from the same member of the group to be identified as such. While the identity of the signer remains anonymous, if the signer sends two messages, anyone can validate that both messages originated from the same signer [13, 14]. This principle is known as linkability. Enforcing linkability in a voting scheme solves the double-voting problem. This is a common issue when permitting voter privacy in the digital realm, but it is solved with LRS.

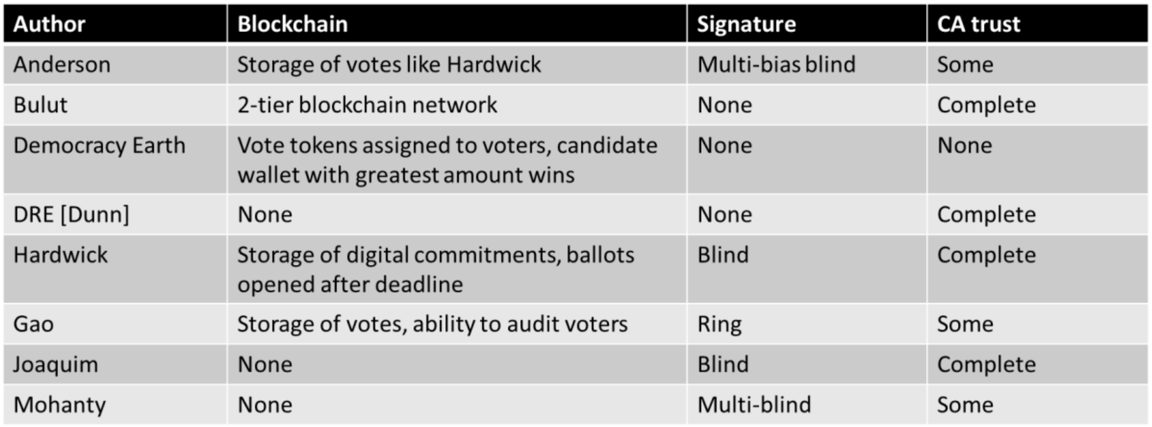
All in all, linkable ring signatures are the most trustworthy way to verify the eligibility of a ballot because they do not require a trustworthy central authority. Because anyone with the public keys of the voters can check each ballot’s legitimacy, voters do not have to depend on anyone else to know that the votes are authentic. The main challenge surrounding LRS is scalability, a problem that will be solved by the approach presented in this paper.

III. CURRENT STATE-OF-THE-ART

As seen in Table 1, the current state-of-the-art approaches can be categorized according to their use of blockchain, the signature scheme used to verify the eligibility of a ballot, and the amount of trust placed in the central authority of the election. In Table 3 (see Section VII), each approach is evaluated according to the same requirements that will be applied to the approach presented in this paper. Small-scale approaches are efficient to a maximum of a few thousand voters, whereas an approach that is usable to a large scale can effectively receive, store, and tally ballots for up to millions of voters, such as national elections. The analysis of scalability, along with the other principles of secure voting, is based on the authors' own claims since many of these approaches have not been fully implemented.

Direct-recording electronic voting (DRE) is used commonly throughout the political voting world today [15]. While it is scalable and generally complies with the principle of privacy, it is untrustworthy for a number of reasons. Privacy is only maintained because none of the information is public. Complete trust is placed in the election's central authority since there is no verifiability, immutability, or ability to audit for eligibility. A corrupted central authority in this environment would be able to change votes and sway elections without any external indications. This is the approach utilized by much of the United States today.

In a system called REVS, Joaquim et al. implemented an e-voting system that could tolerate real-world faults by using redundant servers as backups [16]. While this increases the immutability of the system, it does not make it immutable to the same degree as blockchain. Eligibility is checked through a blind signature from the CA, requiring a trustworthy CA in order for the scheme to be secure [9].



Author	Blockchain	Signature	CA trust
Anderson	Storage of votes like Hardwick	Multi-bias blind	Some
Bulut	2-tier blockchain network	None	Complete
Democracy Earth	Vote tokens assigned to voters, candidate wallet with greatest amount wins	None	None
DRE [Dunn]	None	None	Complete
Hardwick	Storage of digital commitments, ballots opened after deadline	Blind	Complete
Gao	Storage of votes, ability to audit voters	Ring	Some
Joaquim	None	Blind	Complete
Mohanty	None	Multi-blind	Some

Table 1. Categorization of Current State-of-the-Art Approaches

Mohanty and Majhi presented a protocol for proving the eligibility of a ballot without placing complete trust in a single CA [11]. Working with multiple central authorities, the multi-blind signatures remove some of the risk associated with a CA single point of failure. However, they do not remove the possibility of corrupt central authorities working together to validate ineligible ballots or invalidate eligible ones.

Gao et al. proposed a system that utilizes ring signatures to verify the eligibility of ballots and allows for auditing of voters [17]. The ring signatures used in this approach allow anyone the ability to confirm the legitimacy of votes [18]. This negates the threat of a corrupt central authority being able to change votes, but other aspects of the approach prevent it from being a viable e-voting system. The authors' auditing system gives the CA the ability to revoke voter anonymity in order to find potential rule violators, but this defies the principle of privacy. Additionally, the authors mention that the system is best applied at a small scale, which is often not the setting in which an election system is necessary.

A unique approach in the e-voting realm is that proposed by the Democracy Earth Foundation in their "manifesto" [4]. The Foundation declares its stance against authority and government, so it presents a completely decentralized e-voting system based on blockchain. Authentication is done by the community, where only users approved by their peers are permitted to join. Vote tokens are given to everyone and the idea or candidate with the most votes in their wallet after an election is the winner. There are several issues with this approach. First, the lack of a central authority means that this approach is not based in any concrete system that exists today. While perhaps a good statement against authoritarianism, the system is not practical. Also, being a token-based system, there is no voter privacy since the tokens must be trackable to where they came from.

An e-voting system for the country of Turkey was proposed by Bulut et al. [19]. In order to assist with scalability, the blockchain system is split into two levels. The lower level consists of several blockchains that synchronize with the upper level every few minutes so that all nodes maintain a universal ledger. While this decreases latency in the blockchain and increases scalability, the authors fail to address issues like privacy and fairness. In addition, the authentication is done via a single central authority, making it very vulnerable to a corrupt CA being able to change votes and sway the results without detection.

Hardwick et al. presented an e-voting system that satisfies the principles of fairness, privacy, verifiability, eligibility, and coercion-resistance [5]. A very well-made and well-documented approach, it details a digital commitment system that allows for fairness while maintaining verifiability. Unfortunately, the approach relies on blind signatures from a single CA which leaves it vulnerable to a corrupt CA. It also faces scalability issues that limit its application in the real world.

Of the current state-of-the-art approaches analyzed here, the work by Anderson is by far the most secure [10]. He adopted Hardwick's digital commitment to maintain fairness, the votes are stored on a blockchain, forgiveness allows a degree of coercion-resistance, and other principles of secure e-voting are complied with. The method of verifying eligibility is the innovative aspect of Anderson’s approach. He requires a blind signature from each candidate in an election to prove the legitimacy of each ballot, what he calls a “multi-bias blind signature.” According to the paper, this ensures that no central authorities will collude to fake a ballot since they are naturally opposing one another. In reality, this would work well only in a two-party system since two or more central authorities could collude to discard ballots that are likely to be cast in favor of another. The lack of CA oversight in this system allows for corruption in the signers, albeit more difficult to execute than in a traditional blind signature scheme.

While discussing a hypothetical solution that utilizes ring signatures, Anderson concedes that “while this solution does work it has very poor efficiency in large scale elections as linkable ring signatures must be constructed and verified against the entire set of keys for each ballot. This inefficiency requires us to find alternative approaches to construct trustworthy election systems” [10]. The approach presented in the following sections solves the scalability issues associated with linkable ring signatures and allows for a more trustworthy system.

IV. Overall Approach

The approach follows the same three main steps of the e-voting process – setup, voting, and tallying – with substeps as outlined below.

Table 2: Notation

Symbol	**Meaning**
CA	Central authority
v	Voter
V_g	Set of all voters in voting group g
P	Set of all voters in voting population
g	Voting group
G	Set of all voting groups
l_g	List of public keys of group g
b, b_{vf}	Digital ballot, empty ballot, filled ballot of v
k_{pub}/k_{priv}	Voter's asymmetric key pair
LIM_g	Voting group size limit
LIM_n	Blockchain network size limit
e	Encryption key
c	Ciphertext
obj_{LRS}	Object signed by linkable ring signature

Step 1) Setup

Step 1.1) CA authenticates voters: ∀ v in P, CA authenticates v to ensure that v is eligible to participate.

Step 1.2) Voter registers public key with CA: If v is authenticated by CA, v registers k_{pub} with CA and CA stores k_{pub} with the identity of v.

Step 1.3) Registered voters are divided into small groups: Having established beforehand the items on the ballot, the CA will have $b_1...b_i$ where i is the number of different ballots. P will be divided into G, where in each g, $\forall$ v1 $\in V_g \wedge \forall$ v2 $\in V_g$, $b_{v1} = b_{v2}$. The size of G $\geq$ i and the number of v in each g $\leq LIM_g$. If the number of v that have the same ballot is greater than LIM_g, all v with identical ballots should be divided evenly into groups.

Step 1.4) Scalable blockchain network prepared

Step 1.4.1) Blockchain network created for each voting group: $\forall$ g $\in$ G, a blockchain network is created with a practical byzantine fault tolerance (PBFT) consensus algorithm (see details in Section V, Part B).

Step 1.4.2) Upper tiers of network are prepared: A hierarchy of blockchain networks connects all the local networks. There will be j levels in the hierarchy, where

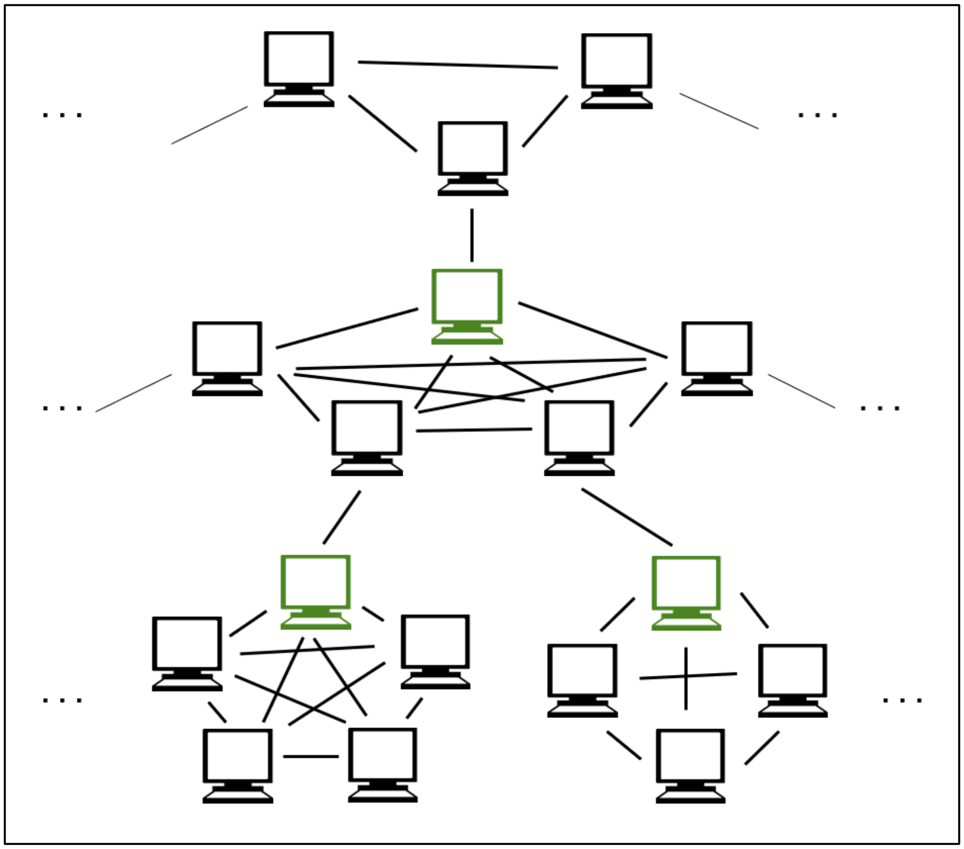


Fig. 1. Simplified Hierarchical Blockchain Network Diagram (Transmission Nodes in Green)

$j = \lceil \log_{LIMn}(\text{size of } G) \rceil$. Each lower network is connected to its superior network via a single node, known as a transmission node, as shown in green in Fig. 1.

The grouping of networks is based on ballot content, similar to the division of voting groups in step 1.3. Lower-level networks that share the most issues on the ballot should be connected to the same superior network.

Step 1.5) Voting group public key lists are published: For each g, $\forall$ v $\in$ V_g, $k_{v,pub}$ is published with voter identity to make l_g.

Step 2) Voting

Step 2.1) Voters fill out their digital ballot: Any voter that wishes to vote fills out their ballot such that b becomes b_f.

Step 2.2) Voters encrypt their digital ballot: v encrypts their ballot b_f with a unique encryption key e_v such that it becomes $enc_{ev}(b_f) = c$.

Step 2.3) Voters sign their encrypted ballots with LRS: v signs c with LRS using l_g and their own k_{priv}, such that c_{LRS} or $enc_{ev}(b_f)_{LRS}$.

Step 2.4) Voters submit signed, encrypted ballot to blockchain network: v sends c_{LRS} to blockchain network of their g.

Step 2.5) Network verifies that the LRS comes from voting group: Signature on c_{LRS} is verified to be signed with the k_{priv} corresponding to a k_{pub} from l_g.

Step 2.6) Signed, encrypted ballot is added to the blockchain: If authenticated, network comes to consensus and adds c_{LRS} to blockchain.

Step 2 can be repeated multiple times, but only the last vote cast by each voter will count toward the final tally.

Step 3) Tallying

Step 3.1) Voters submit signed decryption keys: $\forall$ v $\in$ P, v signs e_v with LRS using l_g and submit e_{vLRS} to network of g.

Step 3.2) Ballots are decrypted: In each group, the signature on each e_{vLRS} will be compared with the set of c_{LRS} to find the latest-submitted c_{LRS} that originated from the same signer. If e_{vLRS} and c_{LRS} are linked, e_v decrypts c. $dec_{ev}(c) = dec_{ev}(enc_{ev}(b_f)) = b_f$.

Step 3.3) Votes in each blockchain network are tallied: Each network totals results found in each b_f into a single block of data.

Step 3.4) Totals are transmitted to network of larger area: The tally block is transmitted to the immediately superior blockchain network via the transmission node.

Steps 3.3 and 3.4 are repeated recursively until reaching the highest level of the hierarchy.

V. Innovation

In this approach, hierarchical blockchain is applied to e-voting in order for linkable ring signatures to be viable in a large-scale election environment. While there are three steps that exhibit innovation, step 1.4 highlights the major innovation of the hierarchical blockchain that makes the ring signatures possible.

A. Step 1.3

In step 1.3, voting groups are created by dividing up the voter population. Every member of a voting group must have an identical ballot, which is to say that each voter in the group will have the same issues to vote on. This is still considered a private ballot because there are many other ballots with the same options on them, maintaining the anonymity of the voter through numbers. Using just the ballot, the likelihood of a person guessing the identity of the voter is $1/b$, where b is the number of voters with an identical ballot. It is important to note that the answers on the ballot do not affect the privacy of the voter, only the questions on the ballot. The same principle follows for the voter privacy of this approach. The anonymity of each voter will depend only upon the size of the voting group that they are a part of. The ability of a person to identify a voter based on their ballot will be $1/v$, where v is the number of members of the voting group.

Since every member of each voting group is required to receive an identical ballot, no voter privacy is lost by this division of the population. These groups are used as rings for the linkable ring signatures. Because the size of the voting groups is fixed, the approach does not suffer from the same scalability issues that other e-voting approaches do.

The division of the voting groups should be based on identical ballot and size. The number of different ballots will be the minimum number of voting groups, but the largest a group will be is LIM_g. If an entire population has the same ballot, the population will be divided to maintain scalability of the ring signatures. If each voter were to receive a different ballot, the election would have no privacy with or without the use of this approach since each ballot could be connected to its voter by checking the options on the ballot.

B. Step 1.4

Step 1.4 outlines the setup for the hierarchical blockchain network. Blockchain is used to ensure the immutability of the system. Voters can be confident that the data viewed on the blockchain will remain the same due to the decentralization of the network and the cryptographic primitives that make the data difficult to alter retroactively.

A blockchain network is created for each voting group in substep 1.4.1, reducing the workload for each of the nodes since the nodes need only maintain a ledger of ballots for a single voting group with a fixed number of voters, rather than the entire population. This assists with latency issues encountered by other approaches when all votes are cast on a single blockchain. With this division of the workload, the entire population can participate in an election with the security of linkable ring signatures without losing effectiveness when used at a large scale.

The consensus algorithm chosen for the blockchain networks in this approach is PBFT. PBFT is an efficient consensus algorithm that has the important characteristic of non-forkability. This removes the possibility of lost votes or version disputes since only one version of the blockchain can ever exist. While this approach is flexible enough to accommodate many consensus algorithms since network and voting group sizes can be easily adjusted, PBFT best satisfies the efficiency and non-forkability requirements for this approach [20].

These voting group networks are then connected with a series of hierarchical blockchain networks, as shown in step 1.4.2. The division of computation in this setup allows for computationally intense security to be deployed at the scale of a lightweight system. While all the local voting groups are computationally independent, the hierarchy connects them such that a large government or organization still can control the entire voting process, rather than managing many small elections and totaling the votes afterward.

Each voting group blockchain network will be connected to a superior blockchain via a transmission node. While the number of blockchain networks connected to each superior blockchain network must be less than or equal to LIM_n, which lower-level blockchain networks are linked to which superior blockchain network is decided by ballot content, similar to the voting group division in step 1.3. The most similar ballots, or those ballots that have the most propositions in common, will be grouped together so that they report their results to the same superior blockchain in steps 3.3 and 3.4. This means that the transmission nodes of these local voting groups will be connected to the same superior blockchain network. This allows efficiency in the tallying process, such that if all the votes on a particular issue have been totaled, they need not be continuously propagated up the hierarchical network.

The number of levels in the hierarchical system will depend upon the size of the voting population and the computational limitations of the physical environment in which the voting system resides. It is expected that the latency of each superior blockchain will increase due to geographic distribution, therefore LIM_n will be less than LIM_g. Therefore, if voting groups are limited to one thousand participants, the number of networks that can contribute to the same superior network may be limited to a few hundred. This means that with a few thousand voting groups, the second level of the hierarchy could have a handful of blockchain networks, and the third level would consist of a single blockchain that collects the global total of all the votes.

C. Step 2.3

In step 2.3, when a voter signs their ballot with a linkable ring signature, the ballot becomes auditable by everyone with access to l_g, the list of public keys for that voting group. In most other e-voting approaches, the final ballots are not auditable – maybe voters can view them, but they have no way of confirming that they are from legitimate voters. They must trust that the central authority has not corrupted any results. The LRS in this approach removes the requirement of a trustworthy CA since every voter can check for themselves that the votes are authentic.

The linkable ring signature scheme used in this approach must have the following characteristics [13]:

1) Unforgeability: Only a voter that is a member of the voting group is able to make a ring signature that would be validated as originating from a member of the voting group.

2) Anonymity: A ring signature from a member of a voting group gives no hints as to the identity of the signer, other than the fact that they are one of the members of the voting group.

3) *Linkability:* Two signatures from the same voter will always be recognized as such. This avoids the double-voting problem and allows decryption to work while the system remains receipt-free.

4) *Nonslanderability:* Voters are unable to cast a ballot so that it is attributed to another voter. Because linkability dictates that only the last vote from each voter be counted, the system must also protect against voters trying to change someone else's ballot by means of impersonation. Nonslanderability makes it impossible to impersonate another voter to change their vote.

Linkable ring signatures allow for the universal verifiability that is required for a voting system to be truly trustworthy. Blockchain provides an immutable storage functionality that is missing from many widely implemented voting systems today. While other modern e-voting approaches have considered linkable ring signatures as a method for verifying voter eligibility, none, to the knowledge of the author, have presented such a trustworthy approach that can be realistically used at a large scale.

VI. An Illustrative Example

The political system of the United States will be used as an example to demonstrate the advantages of the approach. Voting in each state is dictated by the state legislature, but the basic principles that govern the voting process throughout the country are the same.

Step 1) Setup

Setup takes place prior to the election. In the setup phase, the environment for the voting and tallying phases is prepared. It is important to consider the security implications of the actual setup and environment of the voting system. Many of the security vulnerabilities of networked e-voting systems are raised in [21]. Although this approach is a verifiable system that would overcome many of the challenges referenced in [21], care should be taken when considering the physical devices that will make up the nodes of the blockchain network used in this approach.

Step 1.1) CA authenticates voters

In order to prepare the system for the voters, the government must first know how many voters there are and what will be on their ballots. To do this, voter registration takes place. This is governed differently in each state and only requires that the government confirm that the voter is who he claims to be and that he is eligible to vote.

Step 1.2) Voter registers public key with CA

After authentication, this approach only requires that the voter register their public key with the government. The state must keep a list of each public key and with which voter each key is associated.

Step 1.3) Registered voters are divided into small groups

The voting population will be divided into small clusters, or voting groups. The groups should be divided according to identical ballots, or that every voter in a voting group should have all the same issues to vote on. In the United States, a voter will receive a ballot with options related to national, state, county, congressional district, city, and even school district decisions. Therefore, while the ballot is anonymous, by looking at the ballot it is possible to determine the residence of a voter down to their local school district. This is still considered a private ballot because there are many other ballots with the same options on them, maintaining the anonymity of the voter through numbers.

Based on current computational restrictions of linkable ring signatures, voting groups should have a maximum of about 2^{10} registered voters [13]. Conveniently, in the 2004 national elections, the average size of a voting precinct was 1,083 registered voters [22]. Because all voters that belong to the same U.S. voting precinct receive the same ballot, voting precincts could be a natural divider for voting groups in the United States.

Step 1.4) Scalable blockchain network prepared

Step 1.4.1) Blockchain network created for each voting group

Each voting group will have its own blockchain network, so a blockchain network must be prepared for each precinct. This division of the population will decrease the required network size and workload for each node, creating a much less restrictive and a more scalable environment. The function of a blockchain network is reliant on a sufficient number of nodes to keep any single entity from controlling a majority, so nodes of the blockchain should not be limited to just the polling stations of a single precinct [19]. Considerations should be made to distribute the networks and to keep them as decentralized as possible [3].

Step 1.4.2) Upper tiers of network are prepared

Each precinct blockchain network must be connected with a hierarchy of blockchain networks such that no single network has too many nodes or too many contributions that create excessive latency. Regional identifiers for a voter could include, in order from largest to smallest, their state, county, city, congressional district, and voting precinct [23]. Fig. 2 illustrates an example hierarchical blockchain network that utilizes these regional identifiers to divide the workload amongst the blockchain networks. In this diagram, a circle represents a blockchain network [24]. Depending on the state's population size and distribution, an implementation could make the other levels congressional district, county, and state as seen in Fig. 2; or city, county, state; just city and state; etc. The size of the network depends entirely on the size of the voting population and computational limitations. If a network has too much latency due to its size, it can be divided and another layer to the hierarchy can be

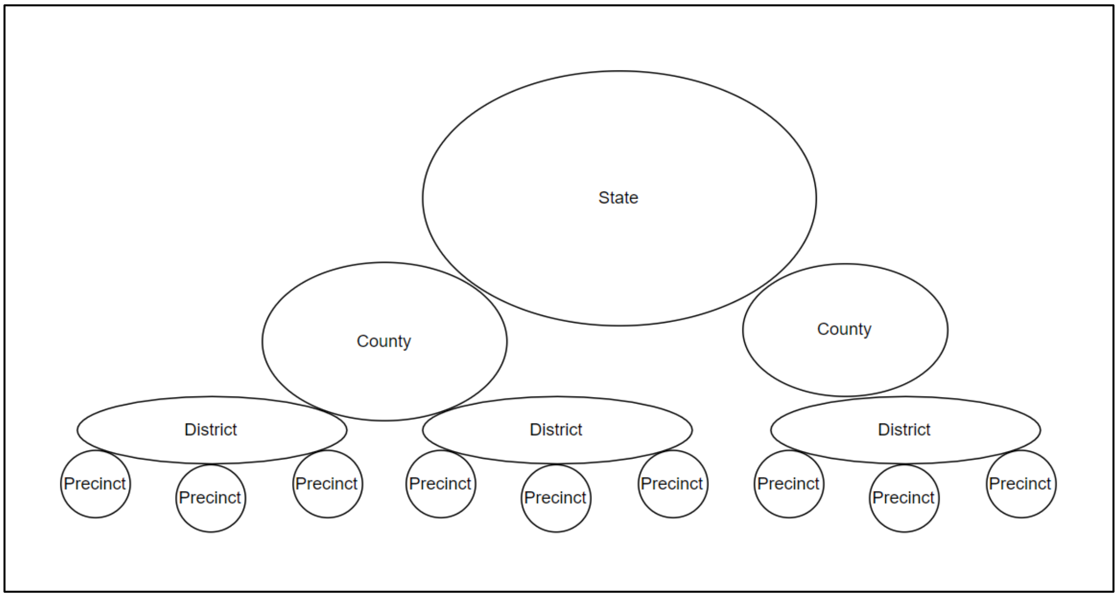


Fig. 2. Example U.S. Scalable Blockchain Network Diagram

added to connect them. This step is important to satisfy the requirement of scalability in an e-voting system, and the flexibility it provides allows it to work in any implementation.

In a U.S. state such as Arizona, which has around 4,000 precincts, there will be about 4,000 local voting groups, one corresponding to each precinct [22]. There are 15 counties in Arizona, so due to the relatively small population of the state, counties could be the next superior level of the hierarchy. This would mean an average of about 266 precincts connected to each county network. The 15 county blockchains would be connected via the transmission nodes to the single state-wide blockchain network.

Each blockchain network consists of nodes that communicate with each other to all maintain the same ledger and authenticate each addition to the blockchain. Each precinct network will be connected to its superior network via a transmission node, a single node that is connected to both networks, as shown in Fig. 1 [25]. Similarly, each network must have a transmission node to connect with its directly superior blockchain network.

Step 1.5) Voting group public key lists are published

Each precinct will have its own blockchain network and ring. The public keys of each voter designated to that group will be published, preferably on the blockchain for the sake of immutability. If a voter loses her private key, she will no longer be able to validate a ballot as a member of the voting group and therefore will be unable to vote. This is easily solved through the central authority, however, by simply republishing the list of public keys to exclude the public key corresponding to the lost private key and to include a newly generated public key that corresponds to the voter’s new private key.

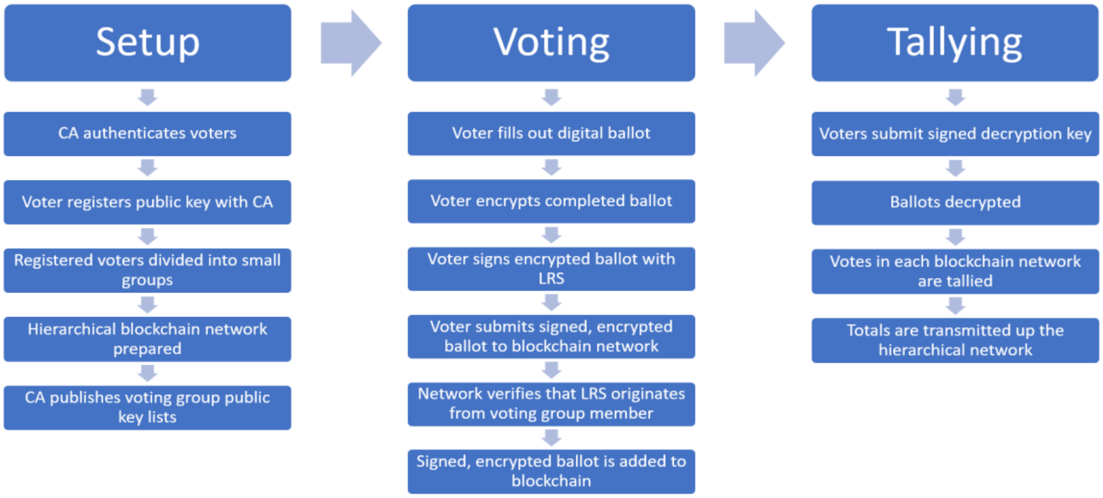

Setup
CA authenticates voters
Voter registers public key with CA
Registered voters divided into small groups
Hierarchical blockchain network prepared
CA publishes voting group public key lists
Voting
Voter fills out digital ballot
Voter encrypts completed ballot
Voter signs encrypted ballot with LRS
Voter submits signed, encrypted ballot to blockchain network
Network verifies that LRS originates from voting group member
Signed, encrypted ballot is added to blockchain
Tallying
Voters submit signed decryption key
Ballots decrypted
Votes in each blockchain network are tallied
Totals are transmitted up the hierarchical network

Fig. 3. Steps of Scalable and Auditable E-Voting Approach

Step 2) Voting

The voting phase lasts only as long as the election is accepting votes. It has a specific start and end time established by the government beforehand.

In order to resist coercion, this approach supports the principle of forgiveness which allows for a single voter to cast multiple ballots, meaning that step 2 can be repeated multiple times. Only the last ballot cast will count towards the final tally, so that each voter only has a single vote.

Step 2.1) Voters fill out their digital ballot

To vote, a voter fills out a ballot with their choices. While this can come in many forms, it is important that the end result be a digital representation of their ballot that is human-readable [21].

Step 2.2) Voters encrypt their digital ballot

Voters encrypt their completed ballots with a unique key. The voter must store the encryption key for the purpose of decryption after the election deadline.

Step 2.3) Voters sign their encrypted ballots with LRS

After filling out and encrypting their ballot, voters will then use the list of public keys from their precinct and their own private key to construct a linkable ring signature for their ballot [12, 13].

Step 2.4) Voters submit signed, encrypted ballot to blockchain network

After signing the ballot, the voter will submit the encrypted ballot with its signature to the blockchain. This will leave the ballot unreadable, but the signature will be verifiable.

Step 2.5) Network verifies that the LRS comes from voting group

Before adding a ballot to the blockchain, the network will verify that the signature attached to the ballot originates from a member of the voting group. A ballot will not be allowed on the blockchain that does not have a valid ring signature from that precinct, and these signatures will be auditable afterwards to preserve the trustworthiness of the ballots.

Step 2.6) Signed, encrypted ballot is added to the blockchain

After verification of the LRS, the ballot must be stored on the precinct blockchain. Since all nodes in a blockchain network should maintain the same ledger, the network must come to a consensus to add the encrypted ballot to the chain. Once cast, the entire ballot will be logged on the blockchain as a single block, where they will be stored until the conclusion of the election when no more votes will be accepted.

Step 3) Tallying

The tallying phase will take place after the conclusion of the election.

Step 3.1) Voters submit signed decryption keys

After the voting window has closed, voters must submit the decryption key for their ballot in order for it to be read and counted. Each voter will sign their decryption key with

LRS like they did their ballot and submit it to the voting group network. Similar to previous steps 2.3-2.6, the key will be signed by the voter, submitted to and verified by the blockchain network, and added to the blockchain.

Step 3.2) Ballots are decrypted

The linkability of the ring signature will confirm that it is from the same owner as one of the ballots already on the blockchain. Using this property, the signature on a decryption key can be compared with the signature on each ballot until the last ballot is found which is determined to have come from the same signer. Once a matching key/ballot pair is found, the key can be used to decrypt the ballot. Using the submitted keys, all ballots can be decrypted and published, either on the blockchain or otherwise.

Step 3.3) Votes in each blockchain network are tallied

Once the ballots have been decrypted, each precinct network will tally all the votes on the blockchain.

Step 3.4) Totals are transmitted to network of larger area

Totals are compiled and published. All of the votes from a single precinct can be combined into a format not much bigger than a single ballot, with totals for each candidate or issue.

Once all the votes from the local blockchain are tallied, it is possible that a part of the election is already complete. If, for example, the local voting group is all the voters from a small town, then a mayoral election only requires votes from the local voting group to confirm a winner. These votes then are not required at a higher level in the election system and any completed results need not be propagated to a wider region.

After filtering out such unnecessary data, the tallies can be passed via the transmission node to the blockchain directly superior in the hierarchy. These will be logged as a single block on the blockchain and consensus will be reached with the other nodes in the network.

Repeat steps 3.3-3.4 as necessary

Once all local blockchains have submitted their totals to the second level network, that blockchain will then repeat the steps of totaling votes, filtering the results, and transmitting the tallies to the superior blockchain. This will be repeated for all the levels of the hierarchical blockchain network.

Once all votes have been tallied, the results of the election will be published by the state government. The benefit of this approach is that voters need not blindly trust that the results announced by the government are true. Voters can verify the results of the election themselves by viewing the blockchains and verifying the signatures on the ballots.

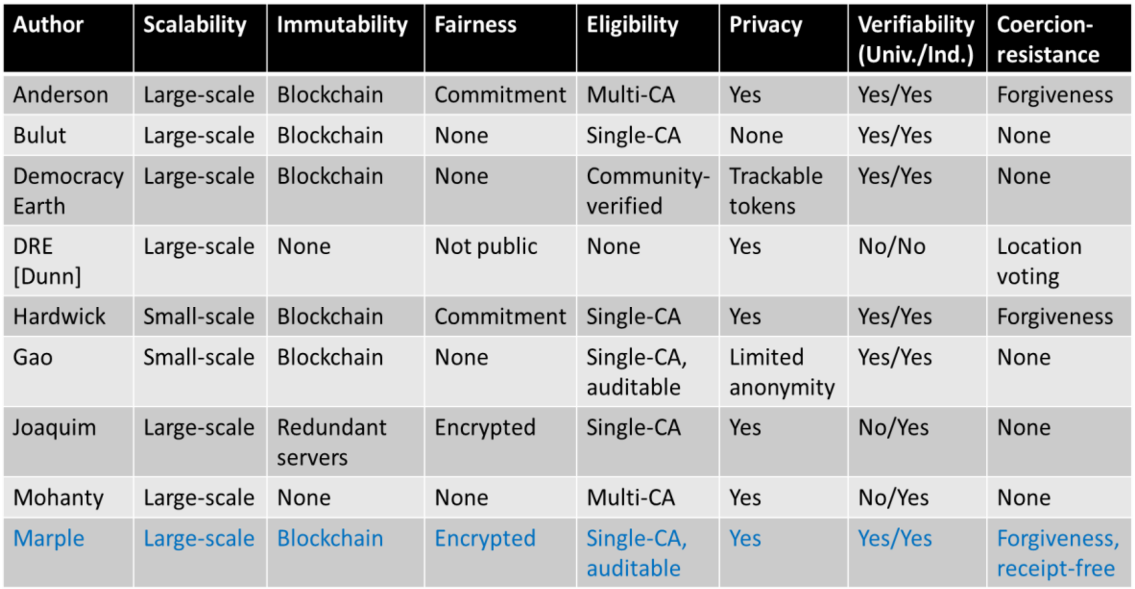
VII. Evaluation

To evaluate the system, compliance with the established requirements will be assessed in this section, along with an analysis of the overall security of the approach.

A. Scalability

In order to portray the scalability of the system, the complexity of each step in the voting process must be analyzed.

For both the encrypted ballot and the encryption key, when adding a block to the blockchain, the network must verify the LRS and come to a consensus to add the block. The LRS verification is linear complexity, based on the number of voters in the voting group, which is capped. Consensus algorithms vary greatly depending on which algorithm is chosen, but they usually depend on the size of the message and the number of nodes in the blockchain network. Message size will be relatively small since large amounts of data need not be submitted in a ballot, and the size of the network is very flexible.



Author	Scalability	Immutability	Fairness	Eligibility	Privacy	Verifiability (Univ./Ind.)	Coercion-resistance
Anderson	Large-scale	Blockchain	Commitment	Multi-CA	Yes	Yes/Yes	Forgiveness
Bulut	Large-scale	Blockchain	None	Single-CA	None	Yes/Yes	None
Democracy Earth	Large-scale	Blockchain	None	Community-verified	Trackable tokens	Yes/Yes	None
DRE [Dunn]	Large-scale	None	Not public	None	Yes	No/No	Location voting
Hardwick	Small-scale	Blockchain	Commitment	Single-CA	Yes	Yes/Yes	Forgiveness
Gao	Small-scale	Blockchain	None	Single-CA, auditable	Limited anonymity	Yes/Yes	None
Joaquim	Large-scale	Redundant servers	Encrypted	Single-CA	Yes	No/Yes	None
Mohanty	Large-scale	None	None	Multi-CA	Yes	No/Yes	None
Marple	Large-scale	Blockchain	Encrypted	Single-CA, auditable	Yes	Yes/Yes	Forgiveness, receipt-free

Table 3. Compliance of Current State-of-the-Art Approaches with Secure Voting Principles, This Approach Included

In order to have an approach that is receipt-free, the matching of an encryption key with its ballot is done by leveraging the linkability of the LRS. While this is not the most efficient way to link these two entries on the blockchain, it is the most secure since no extra information is required that could connect the identity of the voter to their vote. The complexity of this step is at worst O(v^2), where v is the number of voters in a voting group. Again, v is a fixed size controlled by the central authority.

The decryption of the ballots has linear complexity, since each ballot need only be decrypted once, and otherwise only depends on the complexity of the decryption algorithm itself.

Finally, the tallying of votes is done with linear complexity as well. By passing over each ballot one time, the votes can be compiled into one total to be transmitted to the superior blockchain. The superior blockchain network will come to a consensus, as referenced above, and tallied. This will occur recursively until reaching the highest level of the network.

The computation time depends on the size of the voting groups, the algorithm used for LRS, the consensus algorithm of the blockchain network, and the number of levels of the hierarchy, each of which is controlled by the central authority in the setup phase. With such flexibility, this approach is clearly very scalable for practical e-voting situations.

As shown in the illustrative example, in the U.S. state of Arizona with a voting population of about 4.2 million people, there would be about 4,000 voting groups in the lowest level and three levels in the hierarchical network. The state of California, however, has a much larger population with about 25 million registered voters [22]. The hierarchical network could easily scale to accommodate this population size as well by adding voting groups and levels to the hierarchy. Maintaining the same sizes of voting groups and blockchain networks, California would be made up of about 25,000 voting groups, and about 100 blockchains on the second level. A layer could be added to the hierarchy so that the third layer consists of four blockchains and the final tally is found on the sole blockchain at the fourth level of the hierarchy. In this manner, the approach is scalable to any population, large or small, by adding or removing voting groups, blockchain networks, and levels in the hierarchy.

Performance

Several existing implementations of linkable ring signatures were tested to verify the viability of the LRS scheme [26, 27, 28, 29, 30]. Over multiple iterations, the linearity of the complexity was verified. Since scalability analysis is often a question of worst-case scenarios, the data in Fig. 3 is the runtime of the slowest LRS algorithm that was tested. Written in Python, it was tested with 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, and 2048 members in the voting group [26]. Run on an Intel Core i7 1.7 GHz microprocessor, a signature averaged about 2.5 minutes to create and 2 minutes to verify with 1024 members

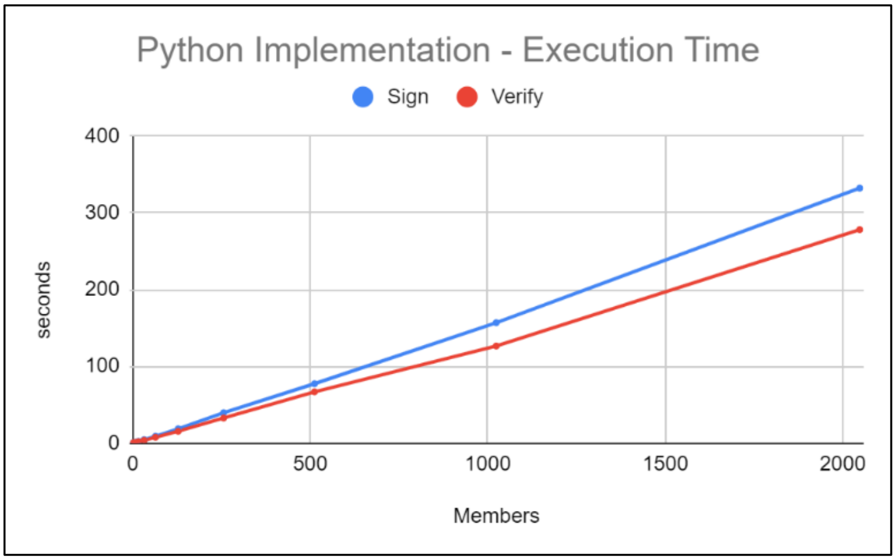


Fig. 4. Runtime Plot of LRS Implementation

in the voting group. In the same environment, other implementations performed far better, with one implementation taking as little as one second to create a signature. While two minutes seems long, even that is acceptable in the given approach since the size of the voting groups is fixed and the votes are submitted, by the nature of blockchain, to different nodes.

In [20] and [31], consensus algorithms are compared for latency and throughput performance. For example, using a blockchain with a proof-of-elapsed-time (PoET) consensus algorithm, the authors achieved a latency of 16 milliseconds with a 250-node network [31]. The throughput of 4 MB transactions in the same network is over one thousand transactions per second. This is well over the required performance for the blockchain network of a voting group, since the number of voters in a group are limited to around one thousand. Practical byzantine fault tolerance (PBFT) achieved a transaction time as low as 0.15 seconds and maintains a very reasonable latency even when some packets are lost in transmission [20]. Depending on the number of nodes in the network, other consensus algorithms range from a throughput of 8,000 transactions per second to only a few per minute.

B. Immutability

The storage of the votes on blockchains preserves the property of immutability in the system. The only legitimate current threat to the immutability of a blockchain is a 51% attack, where a single entity takes control of over half the nodes in a blockchain network and can dictate what the universal ledger contains. While considerations must be given to the number of nodes and the owners of the nodes in an implementation, given enough nodes and a diverse ownership of the network, the blockchains will make the voting ledger very difficult to alter. In addition, the verifiability of the system provides another check on its immutability. Any voter whose vote is changed or is not counted correctly would be able to verifiably assert a problem, while at the same time any votes that do not have legitimate signatures would be auditable by anyone that can view the blockchain.

C. Fairness

During the voting period, the data posted to the publicly viewable blockchain consists entirely of encrypted votes. This will not portray any information except that votes are being cast. Only upon completion of the election and submission of the encryption keys will the ballots become readable. This maintains fairness for everyone, including the central authority and the voters, since only the voter who cast the ballot will be able to decipher their own vote before the election deadline.

D. Eligibility

Eligibility in most voting systems is decided by a central authority. This approach is flexible to the registration process of a managing organization and only requires that the central authority publish the public key of any voter that is deemed eligible. When each voter casts his vote, he will sign the ballot with a linkable ring signature that will prove his eligibility without indicating his identity.

E. Privacy

Each voter's privacy is preserved with the LRS because the signature will only prove that he is a member of the voting group. The vote will be as anonymous as the size of the voting group, where an adversary could guess the identity of a voter with $1/v$ accuracy, where v is the number of voters in the voting group.

The application-specific registration process is not included in this approach, so the only PII used in the entire voting process is the private key used to create a linkable ring signature. This key is assumed to be secret to each voter, as is a standard assumption for public-key infrastructure (PKI).

F. Verifiability

1) Individual: When submitted, ballots will be signed electronically and encrypted with a key that only the voter knows. Voters will be able to verify their own ballot and confirm that the blockchain has logged their votes correctly.

2) Universal: The blockchain will be publicly viewable throughout the election, and the votes will be decrypted for everyone to see after the conclusion of the voting period. Anyone with access to the blockchain will be able to view each individual vote, verify that each ring signature was created by a member of the voting group, and tally the votes to confirm the final count reached. There are no "black boxes" or secrets kept by the central authority in this process.

G. Coercion-Resistance

The linkability of the ring signatures allows for multiple votes from the same voter. This forgiveness is a form of coercion-resistance. Ideally, this permits a coerced voter to cast a ballot when under duress but change the vote later without a coercer knowing. The process is also receipt-free, a desirable aspect of an e-voting system for the sake of coercion-resistance. While it is possible to link a voter to their vote by way of their private key, we assume that each voter maintains the confidentiality of their private key.

H. Security

The attack model for an e-voting system is focused on an adversary that wishes to influence the vote. This includes changing votes, affecting the final tally, removing votes from consideration, or even making it more difficult to cast a vote. While other approaches assume the trustworthiness of an election's central authority, this solution also defends against the possibility of corruption or intrusion in the managing authority of the election, causing the CA to be the adversary. This defense does not come at the expense of defense against other common attack vectors. The principles of secure voting outlined previously defend against data corruption, availability attacks, and intrusion situations.

For example, if an adversary intercepts the traffic of a voter and conducts a man-in-the-middle (MITM) attack to cast a vote contrary to the will of the voter, individual verifiability will solve the problem. Using their private key, the victim will be able to identify their own ballot on the blockchain and discover any changes that have been made to it. Forgiveness allows the voter to cast another ballot, preferably over a more secure connection, that can be verified again for accuracy. Likewise, if an adversary changes the data passing between blockchain networks via a transmission node, the totals will not match the ballots in the lower-level chain but will be easily caught due to the auditability of the system. In general, the universal verifiability combined with the individual verifiability of the system combine as a strong defense against any data changes.

Another common threat to e-voting systems is Denial-of-Service (DoS) attacks. If an adversary can remove the availability of the voting system, then they have affected the vote. This approach stores all votes on blockchains. Blockchain networks by nature are distributed and contain many nodes, making them a much more difficult target for a DoS attack than a traditional few number of servers.

A legitimate threat to this system would be the theft of voters' private keys. With the private key of a voter, an adversary would have the capability to view, cast, and re-cast the victim's vote. While defenses could be implemented on the administrative side by the election's central authority, this approach assumes the privacy of each voter's private key.

VIII. Conclusion

The key innovation included in this approach is the scalable hierarchical blockchain network for e-voting with linkable ring signatures. No other e-voting approach to the knowledge of the author has applied a hierarchical blockchain network in such a way to achieve both the scalability and security that this approach does. The division of the population and workload allows for computationally heavy security to be used at the scale of a lightweight approach without becoming unrealistic. This approach provides trustworthy results without the requirement of a trustworthy central authority.

The improvement on past approaches, namely the trustworthiness of the results combined with scalability, is made possible by the combination of linkable ring signatures and hierarchical blockchain technology. The linkable ring signatures create trust in the authenticity of votes without having to trust the incorruptibility of an election’s central authority. The hierarchical blockchain framework allows the security of the system to be scalable. While neither of the technologies applied are entirely novel, their combination in an e-voting environment is unique and allows for a realistic approach to e-voting security.

IX. Future Work

The evaluation of this research is based mostly on other published works and the observation of established practices. A more effective analysis of the viability of this solution could be accomplished by implementing a prototype of the hierarchical blockchain network with linkable ring signatures. This would allow more exact scalability analysis, and the scalability of the system would be better established than many other approaches that only theorize the scalability of the system, as done in this thesis.

Currently, while there is much research being done in the area of blockchain and blockchain e-voting in particular, there is still significant opposition to the implementation of blockchain voting due to concerns related to internet voting [21]. While this thesis does not include implementation details for any specific government or environment, it is important to point out that if voting is done remotely from voter-owned devices, this opens the door to a wide array of well-documented exploits that would make the election much less secure. Future research should be done in the field of remote e-voting to ensure the viability of secure mobile ballot-casting and allow for practical and convenient voting that is still trustworthy.

Similarly, further research should also be devoted to preserving the persistent availability of an internet-facing voting system. Even now, DoS attacks are a common and cost-effective attack vector for individuals trying to influence an election [32]. DoS attacks are much easier to execute than they are to defend against so having a better defense would greatly advance the field of e-voting. Consideration should be given to the application of machine learning (ML) in this aspect of e-voting security, as it is useful for detecting anomalies that may indicate DoS attacks. Although not much has been done with both ML and e-voting, the domain is certainly worth exploring.

In addition, perhaps the best way to implement this approach would utilize secure hardware to keep the voter’s private key and encryption key safe and secret. Work should be done to consider how trusted platform module (TPM) chips could assist with the privacy and coercion-resistance of e-voting systems.

REFERENCES

[1] Merriam-Webster. (n.d.). Democracy. In Merriam-Webster.com dictionary. https://www.merriam-webster.com/dictionary/democracy

[2] “The Chairman’s Report of the Election Law Study – Subcommittee of the Standing Senate Judiciary Committee.” December 3, 2020. http://www.senatorligon.com/THE_FINAL%20REPORT.PDF

[3] L. Dricot and O. Pereira, “SoK: Uncentralisable Ledgers and their Impact on Voting Systems”, arXiv, 2018.

[4] Democracy Earth Foundation, “The Social Smart Contract,” Version 0.1: September 1, 2017. https://basicincome.org/wp-content/uploads/2015/01/Miller_Sandra_Democracy_ Earth_Foundation_Paper_fort_17th_BIEN_The_Social_Smart_Contract.pd_.pdf

[5] Hardwick et al. “E-Voting with Blockchain: An E-Voting Protocol with Decentralisation and Voter Privacy,” Proc. IEEE Computer Society Conference Publishing Services, 2018, pp. 1561-1567.

[6] R. Taş, Ö. Tanrıöver, “A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting,” Symmetry, 2020.

[7] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” 2008. bitcoin.org.

[8] “BitCongress – White Paper,” http://cryptochainuni.com/wp-content/uploads/BitCongress-Whitepaper.pdf

[9] D. Chaum “Blind Signatures for Untraceable Payments,” in *Advances in Cryptology*, Chaum D., Rivest R.L., Sherman A.T. (eds). 1983. pp. 199-203. Springer, Boston, MA.

[10] B. Anderson, “Improving the Trustworthiness of Electronic Voting Systems Using Blockchain,” Master’s Thesis in Computer Science, School of Computer, Informatics, and Decision Systems Engineering, Arizona State University, May, 2020.

[11] S. Mohanty and B. Majhi, "A Secure Multi Authority Electronic Voting Protocol Based on Blind Signature,“ Proc. International Conf. on Advances in Computer Engineering, 2010, pp. 271-273.

[12] Rivest et al. “How to Leak a Secret,” Proc. Conference on Theory and Application of Cryptology and Information Security, 2001, pp. 554-567.

[13] X. Lu, M. H. Au, and Z. Zhang, "(Linkable) Ring Signature from Hash-Then-One-Way Signature," Proc. 18th IEEE International Conf. on Trust, Security And Privacy In Computing And Communications, 2019, pp. 578-585.

[14] Backes et al. “Ring Signatures: Logarithmic-Size, No Setup – from Standard Assumptions.” Cryptology ePrint Archive, Report 2019/196. 2019.

[15] M. Dunn and L. Merkle, “Overview of Software Security Issues in Direct-Recording Electronic Voting Machines,” 2018.

[16] R. Joaquim et al. “REVS – A Robust Electronic Voting System,” in IADIS International Journal of WWW/Internet. 2004.

[17] S. Gao et al. "An Anti-Quantum E-Voting Protocol in Blockchain With Audit Function," in IEEE Access, vol. 7, 2019, pp. 115304-115316.

[18] A. Kiayias and M. Yung, “Self-tallying Elections and Perfect Ballot Secrecy” in Public Key Cryptography, Lecture Notes in Computer Science, vol 2274. Springer, 2002.

[19] Bulut et al. "Blockchain-Based Electronic Voting System for Elections in Turkey," Proc. 4th International Conference on Computer Science and Engineering (UBMK), 2019, pp. 183-188.

[20] T. Loruenser et al. “Towards a Performance Model for Byzantine Fault Tolerant (Storage) Services,” arXiv, 2021.

[21] Park et al. “Going from bad to worse: from Internet voting to blockchain voting,” *Journal of Cybersecurity*, Volume 7, Issue 1, 2021.

[22] The U.S. Election Assistance Commission, “Polling Places 2004 General Election." EAC Election Day Survey. Archived from the original on December 14, 2006.

[23] The U.S. Election Assistance Commission, “EAVS 2016 Comprehensive Report.” https://www.eac.gov/sites/default/files/eac_assets/1/6/2016_EAVS_Comprehensive_Report.pdf

[24] Li et al. “A Blockchain-Assisted Intelligent Transportation System Promoting Data Services with Privacy Protection,” Sensors. 2020; 20(9):2483.

[25] H. Chai et al. "A Hierarchical Blockchain-Enabled Federated Learning Algorithm for Knowledge Sharing in Internet of Vehicles," Proc. IEEE Transactions on Intelligent Transportation Systems, 2020, pp. 1-12.

[26] F. Lobato and H. Estensen, “Linkable Spontaneous Anonymous Group Signature with Eliiptic Curve Cryptography,” GitHub repository, 2019. https://github.com/fernandolobato/ecc_linkable_ring_signatures

[27] “CLSAG,” GitHub repository, 2020. https://github.com/crate-crypto/CLSAG

[28] A. Centelles and S. Diehl, “Abe-Ohkubo-Suzuki Ring Signatures,” GitHub repository, 2019. https://github.com/adjoint-io/aos-signature

[29] H. Estensen, “Ring-Go,” GitHub repository, 2020. https://github.com/noot/ring-go

[30] J. Borgstrup, “Linkable Ring Signatures over Elliptic Curves,” GitHub repository, 2014. https://gist.github.com/jesperborgstrup/10633874

[31] A. Ahmad et al. "Performance Evaluation of Consensus Protocols in Blockchain-based Audit Systems," 2021 International Conference on Information Networking (ICOIN), 2021, pp. 654-656.

[32] T. Starks, “The Lowly DDOS Attack is Still a Viable Threat for Undermining Elections,” CyberScoop, October 2020. https://www.cyberscoop.com/lowly-ddos-attack-still-viable-threat-undermining-elections/