DecentraVote: A Decentralized Voting Protocol

Jeremiah Giordani, Jonathan Mindel, Estelle Botton, Eden Bendory, and Nicole Klausner

Department of Computer Science, Princeton University

Abstract

In a world where the integrity of democratic societies hinges on secure, transparent, and trustworthy voting systems, traditional methods like paper ballots and mail-in voting have raised concerns about accuracy, privacy, and potential manipulation. To address these challenges, the DecentraVote protocol presents a novel, blockchain-based voting system that offers ease of use, enhanced security, privacy, and transparency throughout the voting process.

The unique DecentraVote protocol embraces the power of smart contracts, zero-knowledge proofs (ZK-proofs), an accessible user interface, and a private blockchain. By harnessing the benefits of blockchain technology, the protocol aims to reinvigorate public trust in the electoral process and foster a more equitable democratic system. This white paper offers an in-depth analysis of existing blockchain voting systems, elucidates the design and implementation of the innovative DecentraVote protocol, underscores the advantages of the proposed approach, and reflects on the process of designing it. In doing so, the protocol offers governments the opportunity to harness this blockchain-based voting system, fortifying the democratic process and placing the trust of their citizens at the forefront.

I - Introduction

The foundation of a democratic society lies in the ability of its citizens to participate in free, fair, and transparent elections. Over time, various voting methods have been utilized, each with its own set of advantages and drawbacks. However, the rise of digital technologies and increasing concerns about the integrity and security of traditional voting systems have led to a renewed interest in the development of alternative voting solutions.[1] One such promising technology is blockchain, which offers an immutable, decentralized, and transparent platform for building secure voting systems.

This white paper introduces a novel blockchain-based voting protocol called DecentraVote, which aims to address the pressing challenges faced by traditional voting systems. The protocol focuses on ensuring ease of use, security, privacy, and transparency in the electoral process. To achieve these goals, it incorporates a unique approach that combines smart contracts, zero-knowledge proofs (ZK-proofs), an accessible user interface, and a private blockchain. The proposed solution was designed with the potential to revolutionize voting systems and restore public trust in the electoral process.

This paper will undertake a comprehensive exploration of blockchain voting systems and their critical components, delving into the challenges of creating a secure, efficient, and user-friendly voting protocol. The DecentraVote protocol emerges as a blend of existing solutions' best features and innovative approaches, overcoming current limitations. This paper will explore the inner workings of the protocol, such as smart contracts, ZK-proofs, and private blockchain utilization. The paper further delves into ownership structure and governance management, followed by an examination of tokens and the DecentraVote protocol's unique approach to accessibility. Ultimately, the paper concludes with an insightful reflection on the proposed solution and the design process involved in crafting the system.

By presenting a detailed analysis of the proposed system and its advantages over traditional voting systems, the DecentraVote protocol contributes to the ongoing dialogue surrounding the development and implementation of secure, transparent, and equitable voting systems for democratic societies.

II - Background

This section will provide the necessary background information to understand the concepts used in the system, assuming the reader has beginner to intermediate knowledge of blockchains. By discussing the technical background, the reader should understand the relevant components of the project, including the problem being addressed, the motivation for a blockchain-based voting system, the approach to developing the decentralized voting protocol, and the fundamental concepts related to the implementation.

Problem Statement and Motivation

In today's rapidly evolving digital age, the integrity, security, and accessibility of voting systems have

become increasingly critical for the sustenance of democracy. Traditional voting methods, characterized by opaque procedures, centralized control, and susceptibility to fraud, are proving inadequate to address the growing demands for transparency, efficiency, and inclusivity. This has led to a decline in public trust and disillusionment with the electoral process. The problem that motivates the development of a blockchain-based voting system is the urgent need to revolutionize the way we vote. Harnessing the power of blockchain technology allows the creation of a decentralized, secure, and transparent voting protocol that not only restores public confidence in the electoral process, but also fosters greater political participation and empowers citizens to actively shape the course of their democratic institutions.

Blockchain Technology and Voting Systems

There are several blockchain-related technical components that support the decentralized voting system.

Blockchains A blockchain is a distributed ledger that consists of blocks, where each block contains a set of transactions. The blocks are cryptographically linked, providing security and immutability to the stored data. In the context of a voting system, a blockchain can be used to validate and store votes in a transparent and tamper-proof record.[2]

Smart Contracts Smart contracts are self-executing contracts with the terms of the agreement directly written into code. They are typically deployed on a blockchain platform, such as Ethereum. Implementations of blockchain voting systems vary in their use of smart contracts, including recording or counting votes and validating voter identity.[3]

Zero-Knowledge Proofs (ZK-Proofs) Zero-knowledge proofs (ZK-Proofs) are cryptographic techniques that allow one party to prove to another party that they know specific information without revealing the information itself. In the context of a voting system, zk-proofs can be used to verify voter identities while maintaining their privacy. [4]

Private Blockchain A private blockchain provides a secure and controlled environment for implementing a decentralized voting system.[5] In some decentralized voting protocols, a private blockchain is used to store voter data and maintain a transparent and tamper-proof record of the election process. This infrastructure ensures that the voting system is both secure and accessible to authorized participants. The nature of a private blockchain allows the entity that governs the blockchain to exert safety controls without having access to modify or tamper the records.

Decentralized Voting Protocol Development

There are several components of decentralized voting systems that are required, in order to ensure the system will function correctly.

Voter Verification and Registration Voter verification and registration are crucial aspects of a decentralized voting system. They ensure that only eligible voters can participate in the election process and prevent double voting or other forms of voter fraud. For example, many decentralized voting systems use voter tokens for voter identification [6], which ensures the integrity of the election through the inherent rules that govern tokens.

Vote Privacy and Immutability Vote privacy and immutability are indispensable features for a reliable voting system. It is vital that votes are recorded and stored in a manner that protects each voter's anonymity while preventing modifications. Blockchain technology inherently lends itself to these goals, securing voter privacy and preserving the integrity of the electoral process.

User Interface and Accessibility A user-friendly interface is essential for a successful voting system. Voters should be able to cast their votes easily and without requiring any additional software or technical knowledge. Most decentralized voting systems focus on providing an interface that can be easily accessed and used by all eligible voters.

III - DecentraVote Protocol

To create a secure and transparent voting system, the protocol combines several key components, including smart contracts for voting and verification, zk-proofs for voter identification, and a private blockchain for secure vote storage. The protocol also includes a web-based user-friendly interface to ensure that voters can easily participate in the election process without the need for extensive technical knowledge.

Overview of Components

Before exploring the specifics of the protocol's unique approach, it is imperative to understand the foundation for most basic implementations of blockchain voting protocols. The core of almost every decentralized voting protocol includes the following 4 fundamental steps: voter registration, vote casting, vote counting, and result verification. Table 1 provides a visualization that categorizes and lays out the traditional structure for easy reference. Accordingly, the DecentraVote protocol is based off of this structure.

By extending this framework and modifying some of the components of the traditional design, the protocol offers a unique, decentralized voting system.

Stage	Description
Voter Registration	Voters register their identity and receive voting credentials
Vote Casting	Voters cast their votes using the provided credentials
Vote Counting	Votes are tallied, and preliminary results are determined
Result Verification	Final results are verified and confirmed

Table 1: *Traditional Blockchain Voting Protocol Structure*

The Protocol

The DecentraVote protocol adheres to the following structure. Figure 1 displays a pictorial representation of the two main phases of the protocol.

1. Government sets up a private blockchain and establishes rules, consensus mechanism, etc.
2. Government initializes a list of approved users to serve as nodes for the private blockchain.
3. Smart contracts (VoterVerification and Voting) are deployed on the network
4. Government registers candidates by adding their information to the Voting smart contract.
5. Voter visits web-based interface and undergoes identity verification, interacting with the VoterVerification contract, providing their personal information.
6. Voter's personal information is used to generate a Zero-Knowledge Proof (ZK-Proof) for secure identity verification.
7. Voters use the user interface to upload and verify their ZK-proof and cast their vote, which is encrypted and linked to their ZK-Proofs.
8. Encrypted votes are stored on the private blockchain and cannot be tampered with.
9. After the voting period ends, a secure consensus mechanism ensures no vote tampering.
10. Results are easily auditable, given that cryptographic proof validates the voting record
11. Results are validated and tallied, and the election outcome is announced by the government.

These components work together to produce the DecentraVote protocol. The proceeding sections lay out more specific explanations of the various components.

IV - Operations

The protocol is sustained through a series of technical components. Each component is critical to the successful functions of the system. The protocol is launched via two smart contracts, which are responsible for managing voters, recording the results, and carrying out the protocol, generally.

Use of Smart Contracts for Voting and Verification

Smart contracts are self-executing agreements with the terms and conditions of the contract directly embedded into the code. The DecentraVote protocol utilizes smart contracts for both voting and verification purposes. The smart contracts ensure that the voting process is transparent, tamper-proof, and highly secure, eliminating the possibility of fraud and manipulation.[7]

The protocol incorporates two main smart contracts: VoterVerification and Voting. These smart contracts are designed to work together to create a secure and transparent voting process. Table 2 analyzes the function and purpose of the two contracts.

VoterVerification Smart Contract The VoterVerification smart contract is responsible for verifying the identity and eligibility of voters. It serves as a gatekeeper that ensures that only eligible voters can participate in the election process. This smart contract plays a vital role in maintaining the integrity of the voting system by preventing unauthorized voting and double voting.

Voters interact with this smart contract by providing their personal information, biometric data, any relevant documentation, etc. This data is then used to verify their eligibility to vote. The verification process is designed to maintain voter privacy by using Zero-Knowledge Proof (ZK-Proof) techniques that securely verify voter eligibility without revealing sensitive data. This ensures that voter information remains confidential while still enabling a transparent and secure verification process. It ensures that only those who are eligible to vote are able to vote, without revealing any information about the voter or their vote[8]

Voting Smart Contract The Voting smart contract is responsible for maintaining the list of candidates, allowing voters to cast their votes, and tallying the votes securely. This smart contract plays a crucial role in ensuring that the election process is transparent, accurate, and unbiased.

Voters interact with the Voting smart contract through a user-friendly interface, which allows them

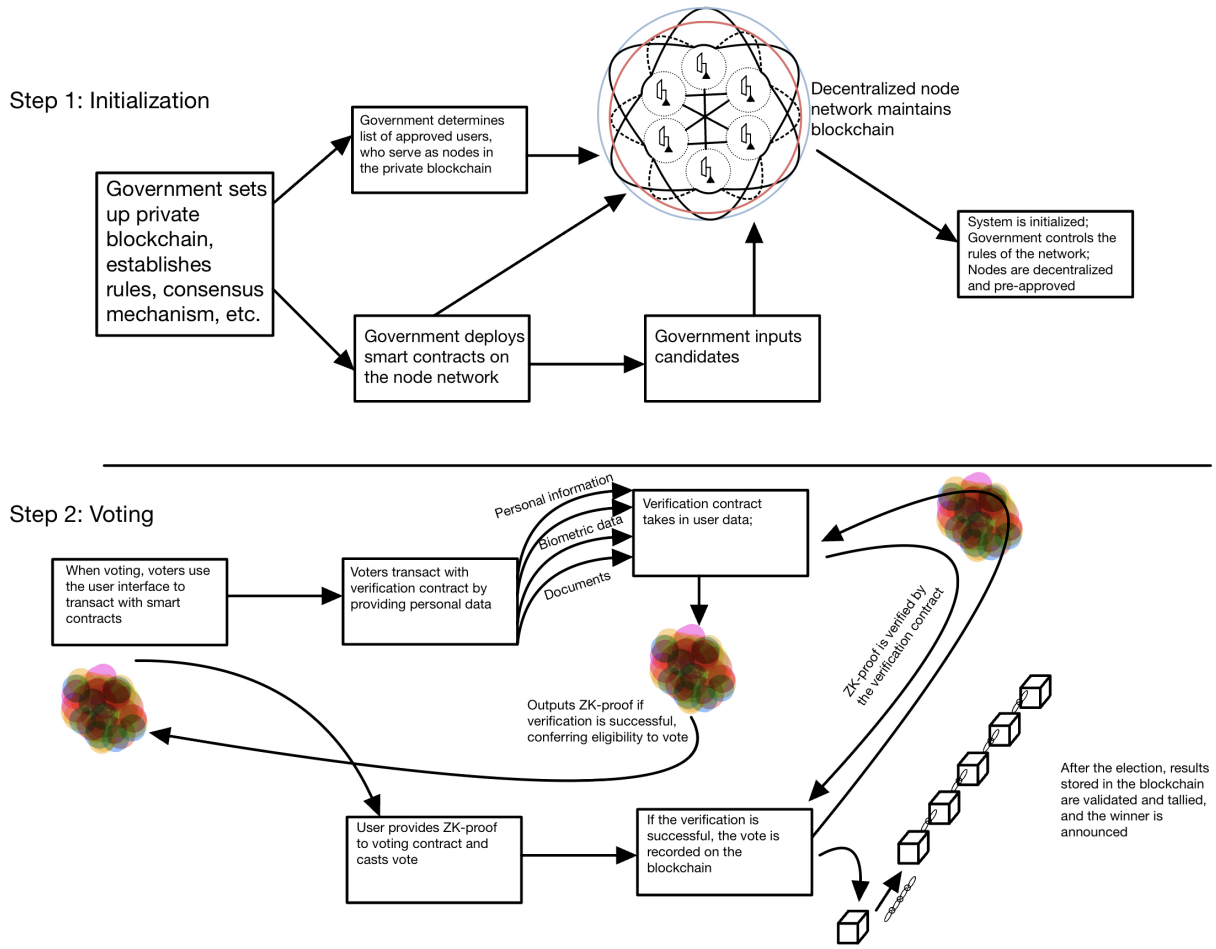

Figure 1: *Voting System Protocol*

to cast their vote for their preferred candidate. The smart contract then validates the voter's eligibility with the verification contract - using the user's previously issued ZK-Proof - and records the vote on the blockchain. This ensures that only eligible votes are counted, and the election outcome accurately represents the will of the voters.

During the vote counting stage, the Voting smart contract automatically tallies the votes and determines the election results. This process is transparent, as anyone with access to the blockchain can independently verify the vote count and confirm the election's fairness and accuracy.

Smart contracts are a critical component of the DecentraVote protocol because they provide a reliable, automated, and secure means of managing the election. By harnessing the inherent strength and capabilities of these automated smart contracts, the protocol effortlessly manages its most prized asset - the votes. The VoterVerification and Voting smart contracts carefully manage the electoral process, ensuring the fidelity, security, and transparency of each vote [9]. Table 2 gives a comprehensive analysis of the function and purpose of the two contracts.

Utilization of ZK-Proofs for Voter Identification

Zero-Knowledge Proofs (ZK-Proofs) are cryptographic techniques that allow one party to prove the validity of a statement without revealing any additional information about the statement itself. The DecentraVote protocol uses ZK-Proofs for voter identification, in order to ensure privacy, immutability, and ease of use.[10]

Benefits of ZK-Proofs Using ZK-Proofs in the design offers several advantages:[11]

- **Privacy:** ZK-Proofs protect voter privacy by allowing them to verify their eligibility without revealing sensitive personal information.
- **Immutability:** Once a ZK-Proof is generated and accepted by the VoterVerification smart contract, it cannot be altered or tampered with, ensuring the integrity of the verification process.
- **Ease of Use:** ZK-Proofs simplify the user experience by enabling voters to prove their eligibility using a compact proof that can be easily transferred between smart contracts.

Smart Contract Components			
VoterVerification Contract		**Voting Contract**	
Function	Verifying voter identity	**Function**	Managing candidate list
	Checking voter eligibility		Recording and tallying votes
Purpose	Maintain voting integrity	**Purpose**	Ensure transparent process
	Prevent unauthorized voting		Validate eligible votes

Table 2: *Smart Contract Analysis*

ZK-Proof Generation and Verification Process The process of utilizing ZK-Proofs for voter identification in the DecentraVote protocol involves the following steps:

1. A voter submits their personal information to the VoterVerification smart contract.
2. The smart contract generates a ZK-Proof by hashing the voter's personal information and other relevant data. This ZK-Proof serves as a compact representation of the voter's eligibility without revealing their actual information.[12]
3. The voter obtains the ZK-Proof from the VoterVerification smart contract. This can be done by downloading a file, copying a line of text, scanning a QR code, or any other convenient method of transferring the proof.
4. The voter interacts with the Voting smart contract and submits their ZK-Proof as a means of validating their identity and eligibility to vote.
5. The Voting smart contract verifies the ZK-Proof by calling the VoterVerification contract, ensuring that the voter is eligible without gaining access to their personal information. If the proof is valid, the voter is allowed to cast their vote.

By incorporating ZK-Proofs in the DecentraVote protocol, the system creates a secure, privacy-preserving, and user-friendly solution for voter identification. This approach enables a transparent and tamper-proof voting process while ensuring that voter privacy remains protected throughout the entire election lifecycle. [13]

Web-Based User Interface Design

A well-designed user interface (UI) is essential for the success of any decentralized voting system, as it ensures that voters can easily understand and navigate the process. The DecentraVote protocol prioritizes usability, accessibility, and security in the design of the UI.

To make the voting process as intuitive as possible, the web-based UI includes clear instructions and visual cues that guide voters through the registration, vote casting, and result verification stages. This includes step-by-step instructions, progress indicators, and tooltips that provide additional context when needed.

In terms of accessibility, the web-based UI is designed to be compatible on any device that can connect to the internet, such as desktop computers, laptops, tablets, and smartphones. This allows voters to participate in the election using the device of their choice. Utilizing a web-based interface prevents the technical and hardware requirements associated with an application interface. Furthermore, the UI adheres to accessibility standards, to accommodate voters with disabilities or language barriers. [14]

To enhance security, the UI incorporates features like two-factor authentication (2FA) and encryption to protect voter credentials and sensitive information. Additionally, the protocol implements strict input validation to prevent potential attacks, such as SQL injection or cross-site scripting (XSS).[15]

V - Governance

In the DecentraVote protocol, the carefully designed governance structure delicately weaves together the concept of authority, responsibility, and engagement. It addresses the question of who holds the reins and how the broader community interacts with the system, ultimately shaping the advantages that emerge from this structure.

Implementation of Private Blockchain

The DecentraVote protocol utilizes a private blockchain to maintain a record of votes. This allows for a certain level of governmental control, while still allowing for decentralized vote validation and recording. As a result, the government can protect and safeguard the process, without having the ability to directly tamper with the vote. [16]

Benefits of Private Blockchains Using a private blockchain in the protocol offers several advantages:

- **Control:** A centralized agency can set the environment rules, determine who is eligible to be a node, and manage the overall operation of the blockchain. Foreign bad-actors will be prevented from interacting with the blockchain. [17]

- **Security:** Private blockchains provide a more secure environment, as the centralized agency can enforce strict security measures, use robust consensus mechanisms, and ensure that only trusted parties participate as nodes. [18]
- **Scalability:** Private blockchains can be more scalable than public blockchains, as the centralized agency has the ability to optimize the network for specific use cases, such as voting. [19]
- **Transparency:** Although managed by a centralized agency, private blockchains still maintain a transparent and tamper-proof record of votes, ensuring the integrity of the voting process. [20]

Implementing a private blockchain in the DecentraVote protocol strikes a balance between the benefits of decentralization, such as security and transparency, and the advantages of centralization, such as control and scalability. [21]

Participants

There are three main categories of participants:

- Node Owners
- Government Officials
- Voters

Individuals running nodes provide computing power to the system. They lend their machines to continuously record, update, and maintain the blockchain ledger. Government officials are responsible for maintaining the system, determining the rules of the blockchain, and ensuring smooth operations. Voters are the actual citizens tasked with using the system in order to cast their vote.

Incentives for Participation Two of the classes of participants are not incentivized to participate: government officials and voters. Instead the incentive to carry out their role comes from external sources. For the voter, that may be their principal belief in the necessity of voting. For the government officials, the incentive to engage with the system comes from the incentive to do their job correctly and not get fired. However, one class of participants, node owners, is unique.

Typically, the nodes that maintain a blockchain will be incentivized with the blockchain's native token. [22] However, the private blockchain used in the DecentraVote protocol does not mint, transfer, or use coins.

Even though the system does not have a central token for incentivizing miners, there are alternative incentives for node owners to participate in the voting system:

- **Democratic Participation:** Participants are motivated by the desire to contribute to the democratic process and help ensure its fairness and transparency. [23]
- **Tax Breaks:** The government could introduce tax breaks or other financial incentives for citizens who participate as nodes in the voting system.
- **Reputation and Recognition:** Participants could gain recognition for their contributions to the voting process, potentially leading to opportunities for community engagement or collaboration with the government on future projects.

This approach enables a highly structured voting process, while offering various incentives to encourage citizen participation.

Advantages of the Governance Structure

The specified control structure of the protocol lends itself to several advantages. Table 4 lays out these advantages and the impact of each advantageous design component.

Security One of the primary advantages of the system is the enhanced security it provides for the voting process. In addition to the benefits of the private blockchain, other control structures that maintain governance, such as smart contracts and zero-knowledge proofs, allows the protocol to ensure that the voting system is secure against tampering, fraud, and other malicious activities. [24] Table 3 lays out a variety of beneficial security features.

Security Feature	Description
Private Blockchain	Permissioned network with controlled access
Smart Contracts	Automated, tamper-proof execution of voting tasks
Zero-Knowledge Proofs	Privacy-preserving voter identification
Two-Factor Authentication	Enhanced protection of voter credentials

Table 3: *Key security features of the DecentraVote protocol*

Transparency The system promotes transparency in the voting process by providing an auditable and verifiable record of all voting-related activities on the private blockchain. This allows authorized voters, election officials, and third-party observers to independently verify the integrity of the election results, boosting public trust in the democratic process.

Accessibility By leveraging modern technology and user-centric design principles, the governance structure improves accessibility for voters. The govern-

ment can allow pre-authorized citizens with a standard computer to contribute to the election by participating in the node network. Additionally, the nature of the system enables voters to cast their votes from anywhere with internet access, increasing convenience and voter turnout. [25]

Cost Efficiency The protocol has the potential to reduce the costs associated with traditional voting methods. By allowing a centralized agency to manage the blockchain, optimize the system, and determine the rules, the protocol streamlines an efficient election process. Furthermore, the use of digital infrastructure eliminates the need for paper ballots, reducing material costs and environmental impact.

Advantage	Impact
Security	Protection against tampering and fraud
Transparency	Verifiable and auditable election process
Accessibility	Inclusive and convenient voting experience
Cost-Efficiency	Reduced costs and environmental impact

Table 4: *Advantages of the DecentraVote protocol*

VI - Tokens

The DecentraVote protocol was meticulously designed to address the inherent challenges associated with traditional voting systems and to establish a more transparent, secure, and accessible democratic process. To that end, it was crucial to consider the implications of various elements that characterize blockchain systems. One such element, tokens, emerged as a point of contention.

Tokens are commonly utilized in blockchain-based voting systems to confer voting eligibility and to incentivize miners who validate and secure the network.[26] However, the DecentraVote system deliberately diverges from this convention. There are several reasons.

1. **Simplifying the Voting Process:** Introducing tokens into a voting system can inadvertently complicate the process for voters, as it necessitates the use of specific software and an understanding of the underlying technology.[27] By forgoing tokens and utilizing an alternative approach, DecentraVote streamlines the process, enabling voters to easily transfer their proof of eligibility without the need for specialized knowledge or technical expertise.
2. **Eliminating Technical Barriers to Entry:** The adoption of tokens in a voting system can create a barrier to entry for less technologically savvy individuals or those without access to digital wallets. Using a web-based user interface, DecentraVote ensures that a broader demographic can participate in the voting process, thereby fostering a more inclusive and representative democratic system.
3. **Ensuring Network Integrity:** In a private blockchain, the use of tokens as incentives for miners might prove counterproductive, as the tokens may not hold significant value outside the network. DecentraVote instead relies on alternative incentives, such as tax breaks, reputation impacts, and a genuine desire for democratic participation, to encourage miners to uphold network security and integrity.
4. **Preventing Token Manipulation:** By avoiding the use of tokens, DecentraVote mitigates potential risks associated with token manipulation, such as vote buying or other fraudulent activities.[28] This decision further strengthens the system's ability to maintain a secure and transparent voting process.

Instead of using tokens to determine voter eligibility, the DecentraVote protocol adopts zero-knowledge proofs (ZK-Proofs) representing a significant leap forward in the pursuit of a more secure, transparent, and accessible voting system. The following points elucidate the advantages of employing ZK-Proofs in contrast to voter tokens:

1. **Enhanced Privacy Protection:** ZK-Proofs enable voters to authenticate their eligibility without divulging any sensitive information, such as their identity or voting choices. This feature safeguards voter privacy and prevents potential coercion or retribution, upholding the essential principle of a secret ballot. [29]
2. **Verifiable Authenticity:** ZK-Proofs allow for the validation of a voter's eligibility without exposing their identity. This ensures that only authorized individuals can cast a vote, contributing to the system's overall security and reducing the risk of fraudulent activities [30].
3. **Simplified Voting Experience:** By using ZK-Proofs, DecentraVote eliminates the need for voters to manage digital wallets and navigate complex token transactions. Instead, voters can easily transfer their proof of eligibility through various convenient methods, such as downloading a file or scanning a QR code, thereby making the voting process more accessible to a wider range of participants.

4. **Scalability and Efficiency:** ZK-Proofs streamline the voting process by reducing the computational and storage requirements associated with token-based systems. This not only allows for greater scalability but also minimizes the environmental impact of the blockchain voting system.
5. **Resistance to Token-Related Risks:** By employing ZK-Proofs, DecentraVote circumvents the potential pitfalls associated with token-based systems, such as vote buying, token manipulation, or the creation of financial barriers to entry. This fosters a more secure and representative democratic process.

VII - Reflections

The journey of designing the DecentraVote protocol and authoring this white paper has been a multifaceted endeavor, marked by rigorous research, strategic planning, and the pursuit of innovative solutions. This Reflections section offers a unique opportunity to provide insight into our experiences, the challenges encountered, and the open questions that remain unanswered. In doing so, we hope to share the valuable lessons learned and foster a deeper understanding of the complex process that underlies the creation of the DecentraVote protocol.

The Process

The approach to developing the DecentraVote protocol was characterized by two distinct yet complementary aspects: Research and designing an example implementation. These two components were instrumental in refining our understanding of the system and its technical requirements, as well as in testing the feasibility of the unique approach.

Research The research journey involved a meticulous exploration of academic articles and resources related to decentralized voting systems. We approached this task in three distinct phases, each serving a specific purpose in the development of the DecentraVote protocol.

1. ***Fundamentals:*** Initially, we focused on comprehending the core technical components that underpin decentralized voting systems, such as smart contracts, blockchains, and ZK-Proofs. This foundational knowledge was crucial in establishing the groundwork for the design of the DecentraVote protocol.
2. ***Design Variations:*** Next, we delved into the diverse array of design elements featured in existing literature. By examining the nuances of various technical implementations, we were able to appreciate the impact of these differences on the overall system. This comparative analysis informed our decisions regarding the unique design elements of the DecentraVote protocol, such as:
    - Employing two smart contracts instead of one or other configurations found in the literature
    - Opting for a private blockchain over a public one
    - Utilizing ZK-Proofs for voter eligibility rather than tokens
    - Implementing a web-based user interface as opposed to an application
3. ***Cross-Disciplinary Implications:*** Lastly, we sought to understand the broader implications of decentralized voting systems beyond their technical aspects. This phase of research encompassed the socio-political benefits and challenges of implementing such systems. Our findings greatly influenced the design of the DecentraVote protocol, as we aspired to create a solution that was both technically sound and responsive to the contemporary political and cultural climate. The decision to use ZK-Proofs and a private blockchain, for example, was informed by our commitment to ensuring accessibility, security, and government oversight to prevent potential interference.

Methodically exploring the various dimensions of decentralized voting systems allowed the research process to develop a well-informed and carefully considered design for the DecentraVote protocol. This resulted in a system that is uniquely poised to address the complex challenges of modern-day elections.

Example Implementation: In order to assess the feasibility and effectiveness of the DecentraVote protocol, we created an example implementation that closely emulated the DecentraVote protocol. This prototype served as a valuable tool in refining our understanding of the system's functionality and identifying potential areas for improvement.

The example implementation primarily consisted of developing and deploying the smart contracts, which govern the operations of the system. We tested their functionality on a test network, thus gaining important insights into the performance of the DecentraVote protocol.

Smart Contract Components The smart contract components that make up the DecentraVote protocol relies on two primary smart contracts: VoterVerification and Voting. The VoterVerification contract is designed to verify the identity of voters and ensure

eligibility, while the Voting contract is responsible for recording and tallying votes accurately and transparently on the blockchain.

VoterVerification Contract: In the example implementation, the VoterVerification contract consists of three main methods: verifyVoter, generateZkProof, and verifyZkProof.

1. ***verifyVoter***: This method takes in personal information and, if approved to vote, calls the generateZkProof method with the same parameters. This method is a simplification of the actual verification process, as the actual implementation involves cross-referencing multiple sources of data, including government records and user inputs. For our sample implementation, we simplified this method to only take a voter's name, driver's liscence number, and SSN.
2. ***generateZkProof***: This method takes the inputs provided by verifyVoter and uses the abi library's encodePacked function to generate a valid zk-proof. The generated zk-proof is then returned in byte form.
3. ***verifyZkProof***: This method accepts a byte zk-proof as input and verifies its validity using a global mapping variable called verifiedProofs. This mapping maps the zk-proof byte value to a boolean (true), indicating successful verification.

Upon successful verification, the verifyVoter method emits the zk-proof via the VerificationCompleted event.

Voting Contract: In the example implementation, the Voting contract contains four primary methods, a constructor, and a Candidate struct. The Candidate struct encapsulates the data necessary for each candidate, including their name and vote count. A global mapping variable maps each Candidate to a uint256 (an array) to maintain information about each candidate. The smart contract methods are summarized in Table 5.

1. ***Constructor***: The constructor takes in the block address of the verification contract, so that the Voting contract can ensure the user has been verified using the VoterVerification contract.
2. ***setCandidate***: This method accepts two strings, initializes Candidate structs for them, and adds the structs to the mapping (array) of candidates. Currently, the contract initializes a field of two candidates; however, additional candidates can be added using the addCandidate method.
3. ***addCandidate***: This method adds additional candidates to the most recent index in the candidates mapping.
4. ***vote***: This method takes in a uint256 that maps to a candidate and a zk-proof. The zk-proof is verified using the verifyZkProof method of VoterVerification, and a single vote is tallied to the candidate index's corresponding voteCount field in the Candidate struct.
5. ***getCandidateName***: This method returns the candidate's name via the mapping of candidates.
6. ***getCandidateVote***: This method returns the candidate's vote count via the mapping of candidates.

Limitations and Simplifications: The DecentraVote protocol employs several components that were impossible to represent in their entirety through the example implementation. Rather, the example implementation is designed to provide a practical example of how the protocol functions. However, it is important to emphasize that the primary focus of this paper is the protocol itself. The current implementation serves as a means to help the reader understand the protocol's potential and its key components, while acknowledging its limitations and simplifications:

Voter Verification: In the current implementation, the verifyVoter method is a simplification of the actual verification process. In actuality, this method involves cross-referencing biometric data, technological components, government records, and user data. However, for demonstration purposes, we have simplified it to accept a user's name, driver's license number, and social security number as inputs. It emulated the functionality of zk-proof generation and verification process in the example implementation with the following equation:

$$\textit{zk-proof} = \textit{Hash}(\textit{name}, \textit{driversLicense}, \textit{ssn}, ...) \quad (1)$$

Since the entirety of the voter verification method is outside the scope of this project, Equation 1 displays a simplification of the process to generate the ZK proof used in the example implementation.

Private Blockchain: The sample implementation emulated a private blockchain environment by deploying the smart contracts on a local, testing blockchain through Ganache. The process of initializing the blockchain rules, managing contributors, and controlling the operations of the private blockchain was outside the scope of the sample implementation.

Scalability: The current implementation does not fully address the scalability concerns that may arise when the system is deployed on a large scale, such as during a national election. [31] The actual implementation is optimized and tested to ensure that it can handle a high volume of voters and candidates efficiently.

Security: While the example implementation leverages zk-proofs to protect voter privacy, additional

Method	Description
verifyVoter	Accepts a name, driver's license, and SSN as arguments, and verifies the voter's identity
generateZkProof	Generates a valid zk-proof given the provided inputs
verifyZkProof	Verifies the validity of a byte zk-proof using the verifiedProofs mapping
Constructor	Accepts a verification contract address for user verification
setCandidate	Initializes Candidate structs and adds them to the candidates mapping
addCandidate	Adds additional candidates to the most recent index in the candidates mapping
vote	Accepts a uint256 that maps to a candidate and a zk-proof, and tallies a vote for the candidate after verifying the zk-proof
getCandidateName	Returns the candidate's name via the mapping of candidates
getCandidateVote	Returns the candidate's vote count via the mapping of candidates

Table 5: *Voting Contract Methods*

security measures are enacted for a real-world deployment. This includes implementing additional cryptographic techniques and ensuring the robustness of the underlying blockchain infrastructure.

The current implementation serves as a foundation for understanding the key components of the DecentraVote protocol. It is crucial to recognize that the current implementation is a simplified version of the protocol, and further development would be required to address the limitations and optimize the system for real-world deployment.

Challenges

Developing the DecentraVote protocol presented several challenges that tested our resolve and forced us to adapt and evolve our understanding of decentralized voting systems.

- Lack of Information: As we embarked on this endeavor, we found that sourcing pertinent research and technical assistance for our project was often an arduous task. The nascent nature of decentralized voting systems meant that relevant information was sometimes scarce, making it challenging to assemble the pieces of the protocol puzzle.
- Designing the System: With a plethora of diverse implementations available, determining the most suitable components for the protocol proved to be a complex undertaking. We had to consider the various elements of existing systems and evaluate their potential impact on the overall performance and security of our protocol, all while striving to create a novel and efficient solution.
- Addressing Social Concerns: In designing the DecentraVote protocol, we aimed to ensure that the system effectively addressed the social concerns unearthed during our research. Identifying the most appropriate configuration for addressing these concerns, while balancing the technical requirements of the protocol, presented a unique challenge. By incorporating elements such as ZK-proofs and private blockchains, we were able to create a system that is not only secure but also accessible and adaptable to the current political and cultural landscape.

In overcoming these challenges, we attempted to refine our understanding of decentralized voting systems and honed the DecentraVote protocol to meet the demands of modern democratic societies.

Open Questions

As we reflect upon the development of the DecentraVote protocol, several open questions remain that warrant further exploration and investigation.

One critical inquiry pertains to the barriers that still need to be overcome to facilitate the nationwide implementation of our system. Deploying a decentralized voting system on a large scale poses significant challenges, both technical and logistical, and understanding these obstacles will be crucial for the successful integration of our protocol into the democratic process.

Another open question revolves around optimizing the system for ease of use. While we have made strides in creating an accessible voting system, there is always room for improvement. Exploring methods to streamline the user experience and minimize technical hurdles will contribute to increased adoption and engagement from eligible voters, ultimately enhancing the democratic process.

Lastly, the issue of security remains a paramount concern. As with any system that deals with sensitive information and processes, it is essential to assess and address potential vulnerabilities. Identifying the most pressing security concerns for the DecentraVote protocol and developing effective countermeasures will ensure the continued integrity and credibility of our system in the eyes of the voting public.

These open questions offer a roadmap for future research, development, and refinement of the DecentraVote protocol.

VIII - Conclusion

As we stand at the precipice of technological innovation revolutionizing our democratic systems, the DecentraVote protocol serves as a model to reshape the way we understand elections. [32] This paper has sought to provide a glimpse into a future where voting systems transcend the limitations of the past, embracing a new era of security, transparency, and accessibility. The unification of blockchain technology, smart contracts, zk-proofs, and private blockchain infrastructure within the DecentraVote protocol embodies a radical departure from conventional voting methods, paving the way for a more resilient and equitable democratic landscape [33]. In this way, we dare to dream of a world where technology and democracy walk hand in hand, empowering individuals to participate in shaping their own destinies.

References

[1] T. Kohno et al. "Analysis of an electronic voting system". In: *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*. 2004, pp. 27–40. DOI: 10.1109/SECPRI.2004.1301313.

[2] Robin Singh Bhadoria et al. "Implementing Blockchain-Based Traceable Certificates as Sustainable Technology in Democratic Elections". In: *Electronics* 11.20 (2022), p. 3359.

[3] Friðrik Þ Hjálmarsson et al. "Blockchain-based e-voting system". In: *2018 IEEE 11th international conference on cloud computing (CLOUD)*. IEEE. 2018, pp. 983–986.

[4] Claudia Daniela Pop et al. "Blockchain and demand response: Zero-knowledge proofs for energy transactions privacy". In: *Sensors* 20.19 (2020), p. 5678.

[5] Jun Huang et al. "The application of the blockchain technology in voting systems: A review". In: *ACM Computing Surveys (CSUR)* 54.3 (2021), pp. 1–28.

[6] Ruhi Taş and Ömer Özgür Tanrıöver. "A manipulation prevention model for blockchain-based e-voting systems". In: *Security and communication networks* 2021 (2021), pp. 1–16.

[7] Bhabendu Kumar Mohanta, Soumyashree S Panda, and Debasish Jena. "An Overview of Smart Contract and Use Cases in Blockchain Technology". In: *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 2018, pp. 1–4. DOI: 10.1109/ICCCNT.2018.8494045.

[8] Hoil Ryu, Dongwoo Kang, and Dongho Won. "On a Partially Verifiable Multi-party Multi-argument Zero-knowledge Proof". In: *2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM)*. 2021, pp. 1–8. DOI: 10.1109/IMCOM51814.2021.9377407.

[9] Sara Rouhani and Ralph Deters. "Security, Performance, and Applications of Smart Contracts: A Systematic Survey". In: *IEEE Access* 7 (2019), pp. 50759–50779. DOI: 10.1109/ACCESS.2019.2911031.

[10] Lasse Herskind, Panagiota Katsikouli, and Nicola Dragoni. "Privacy and Cryptocurrencies—A Systematic Literature Review". In: *IEEE Access* 8 (2020), pp. 54044–54059. DOI: 10.1109/ACCESS.2020.2980950.

[11] Eli Ben-Sasson et al. *Scalable, transparent, and post-quantum secure computational integrity*. Cryptology ePrint Archive, Paper 2018/046. https://eprint.iacr.org/2018/046. 2018. URL: https://eprint.iacr.org/2018/046.

[12] Tommy Koens, Coen Ramaekers, and Cees Van Wijk. "Efficient zero-knowledge range proofs in ethereum". In: *ING, blockchain@ ing. com* (2018).

[13] Xiao-Jun Wen et al. "Blockchain consensus mechanism based on quantum zero-knowledge proof". In: *Optics & Laser Technology* 147 (2022), p. 107693.

[14] S Sridevi. "User interface design". In: *International Journal of Computer Science and Information Technology Research* 2.2 (2014), pp. 415–426.

[15] Ugochi Oluwatosin Nwokedi, Beverly Amunga Onyimbo, and Babak Bashari Rad. "Usability and security in user interface design: a systematic literature review". In: *International Journal of Information Technology and Computer Science (IJITCS)* 8.5 (2016), pp. 72–80.

[16] Dominique Guegan. "Public blockchain versus private blockhain". In: (2017).

[17] Suzan Almutairi, Nusaybah Alghanmi, and Muhammad Mostafa Monowar. "Survey of centralized and decentralized access control models in cloud computing". In: *International Journal of Advanced Computer Science and Applications* 12.2 (2021).

[18] Drew Ivan. "Moving toward a blockchain-based method for the secure storage of patient records". In: *ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, Maryland, United States: ONC/NIST*. sn. 2016, pp. 1–11.

[19] Wenting Li et al. "Towards Scalable and Private Industrial Blockchains". In: *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*. Abu Dhabi, United Arab Emirates: Association for Computing Machinery, 2017, pp. 9–14. URL: https://doi.org/10.1145/3055518.3055531.

[20] F Rizal Batubara, Jolien Ubacht, and Marijn Janssen. "Unraveling transparency and accountability in blockchain". In: *Proceedings of the 20th annual international conference on digital government research*. 2019, pp. 204–213.

[21] Andrea Barbon and Angelo Ranaldo. "On the quality of cryptocurrency markets: Centralized versus decentralized exchanges". In: *arXiv preprint arXiv:2112.07386* (2021).

[22] Tien Tuan Anh Dinh et al. "Blockbench: A framework for analyzing private blockchains". In: *Proceedings of the 2017 ACM international conference on management of data*. 2017, pp. 1085–1100.

[23] Jeffrey L Brudney and Nara Yoon. "Don't you want my help? Volunteer involvement and management in local government". In: *The American Review of Public Administration* 51.5 (2021), pp. 331–344.

[24] Miguel Ángel Prada-Delgado et al. "PUF-derived IoT identities in a zero-knowledge protocol for blockchain". In: *Internet of Things* 9 (2020), p. 100057.

[25] Hany F Atlam et al. "Blockchain with internet of things: Benefits, challenges, and future directions". In: *International Journal of Intelligent Systems and Applications* 10.6 (2018), pp. 40–48.

[26] Nir Kshetri and Jeffrey Voas. "Blockchain-enabled e-voting". In: *Ieee Software* 35.4 (2018), pp. 95–99.

[27] Jason Paul Cruz and Yuichi Kaji. "E-voting system based on the bitcoin protocol and blind signatures". In: *IPSJ Transactions on Mathematical Modeling and Its Applications* 10.1 (2017), pp. 14–22.

[28] Pavel Tarasov and Hitesh Tewari. "The future of e-voting." In: *IADIS International Journal on Computer Science & Information Systems* 12.2 (2017).

[29] Lasse Herskind, Panagiota Katsikouli, and Nicola Dragoni. "Privacy and Cryptocurrencies—A Systematic Literature Review". In: *IEEE Access* 8 (2020), pp. 54044–54059. DOI: 10.1109/ACCESS.2020.2980950.

[30] Ruizhong Du, Caixia Ma, and Mingyue Li. "Privacy-Preserving Searchable Encryption Scheme Based on Public and Private Blockchains". In: *Tsinghua Science and Technology* 28.1 (2022), pp. 13–26.

[31] Uzma Jafar et al. "A Systematic Literature Review and Meta-Analysis on Scalable Blockchain-Based Electronic Voting Systems". In: *Sensors* 22.19 (2022), p. 7585.

[32] Ghassan Z Qadah and Rani Taha. "Electronic voting systems: Requirements, design, and implementation". In: *Computer Standards & Interfaces* 29.3 (2007), pp. 376–386.

[33] Ahmed Ben Ayed. "A conceptual secure blockchain-based electronic voting system". In: *International Journal of Network Security & Its Applications* 9.3 (2017), pp. 01–09.