Blockchain-Based Voting System for Secure Distributed Elections

Waseem Admad
Harvard College
waseemahmad@college.harvard.edu

Max Peng
Harvard College
mpeng@college.harvard.edu

05/04/25

Abstract

2nd draft Internal notes: add references

1 Introduction

In an era where digital technology has taken over aspects of daily life, the persistence of largely physical and centralized voting systems stands in stark contrast. Most modern democracies continue to rely on in-person polling places and rudimentary practices. While these traditional approaches offer familiarity, they also present persistent concerns regarding accessibility, scalability, transparency, and public trust in electoral outcomes.

Recent debates surrounding the integrity and inclusivity of elections (ie. the past few presidential elections) demonstrate the need to reimagine how voting systems can leverage modern computational advances. Distributed systems, and blockchain technologies in particular, offer the potential to fundamentally reshape how elections are conducted by introducing tamper-resistance, decentralization, and verifiable auditability.

1.1 Motivation

The primary motivation behind our project is to demonstrate the feasibility and value of a blockchain-based voting system that addresses long-standing challenges in traditional election infrastructure. We thoroughly enjoyed the Blockchain lecture in CS 2620 class, and ultimately wanted to build a difficult but meaningful distributed system project revolving around blockchain transactions.

1.2 Problem Formulation

For this project, we aim to design and implement a secure, transparent, and distributed election platform that:

1. **Ensures vote integrity**: Each vote, once cast, is immutable and tamper-proof through decentralized ledger storage.
2. **Promotes transparency**: All recorded votes are publicly verifiable, while preserving voter anonymity.
3. **Eliminates central dependence**: The system does not rely on a single trusted authority for validation, reducing the risk of corruption or failure.

Our approach centers on building a prototype that mirrors the core components of a real-world election system. This includes a web-based voting client, a distributed backend built on the Ethereum blockchain, a consensus protocol for validating votes, IPFS-based storage for decentralized candidate data, and an audit mechanism to verify election results independently. Through this project, we overall hope to explore how the foundational principles of distributed computing we've learned about throughout the semester can be applied to one of society's most foundational processes: voting.

2 Methodology

We will provide a very brief overview of our system, then discuss specific mechanisms of certain meaningful aspects, namely the, consensus algorithm, use of IPFS, transactions of ethereum blockchain, and audit mechanism.

2.1 Brief Overview of System

system overview here:

2.2 Consensus Algorithm

note: need to un-LLM this To coordinate agreement among validator nodes and maintain the integrity of the blockchain ledger, we implement a simplified Proof-of-Authority (PoA) consensus algorithm. Unlike Proof-of-Work (PoW) or Proof-of-Stake (PoS), which require either significant computational effort or financial collateral, PoA relies on a set of pre-designated authority nodes to validate and produce blocks. This approach is particularly well-suited for our use case, where scalability, speed, and fault tolerance are essential, and the threat model assumes partially trusted validators (e.g., campus servers or verified election officials).

In our implementation, a validator is deterministically selected in round-robin fashion from a predefined list of authority keys. Each block includes up to ten transactions, with each transaction containing the voter's election ID, vote data, and a digital signature. The consensus algorithm first verifies each transaction by checking that the voter's signature matches the content of their vote. Once validated, the block is accepted only if its previous hash matches the last block in the chain, its own hash is consistent with its contents, and it was produced by an authorized validator. This lightweight, deterministic approach ensures that vote inclusion is both efficient and tamper-resistant, without incurring the high resource costs of traditional blockchain consensus mechanisms.

2.3 Decentralized Storage with IPFS

In our blockchain-based voting system, we employ the InterPlanetary File System (IPFS) to achieve decentralized and immutable storage of election-related data. IPFS is a peer-to-peer distributed file system that allows for the storage and sharing of data in a decentralized manner, eliminating reliance on centralized servers. This approach enhances the system's resilience against data tampering and single points of failure.
Specifically, IPFS is utilized to store static election data such as candidate information, election metadata, and other relevant documents. By uploading these files to IPFS, each piece of data is assigned a unique content identifier (CID) based on its cryptographic hash. These CIDs are then referenced within our Ethereum smart contracts, ensuring that the data remains tamper-proof and verifiable. This integration allows voters and auditors to retrieve and verify election data directly from the IPFS network, promoting transparency and trust in the electoral process.
The use of IPFS in our system not only decentralizes data storage but also complements the blockchain's immutable ledger by providing a scalable solution for storing larger files that are impractical to store directly on-chain. This hybrid approach leverages the strengths of both technologies to create a robust and secure voting infrastructure.

2.4 Ethereum blockchain

-talk about ledger, reference consensus mechanism -discuss how this is not real mining (off-chain?)

2.5 Audit Mechanism

-not sure what to write here

3 Journal

2-3 pages go through 1 week process

3.1 Challenges

4 Results

4.1 General System Design

2 pages
-walk through what the system looks like

4.2 System Design Choices

-justify system design

4.3 Example Election walkthrough

-walk through setting up an example election

5 Conclusion

1 page

5.1 Future Work

References