

A Decentralized Voting System on the Ethereum Blockchain – A Solution to the Lack of Trust in Elections?

Shyam Shankar Hemachandran Rani, Dario Nalis
Ludwig Maximilians University Munich
shyam.rani@campus.lmu.de, dario.nalis@campus.lmu.de

Abstract

In recent US elections, the legitimacy of the outcomes has been questioned, leaving millions of Americans in doubt about the integrity of the voting process. In this report we examine the potential of blockchain-based elections on Ethereum as a solution to build trust in the election process. The main objections from recent elections are identified, and a toy example of an election on the Ethereum blockchain is implemented. The blockchain-based system is then evaluated for its merits in resolving trust and security issues. Our conclusion is that while blockchain technology has potential, it is likely not suitable for increasing trust in elections at this time.

1. Introduction

The two recent elections in the USA 2016 and 2020 were overshadowed by claims of the respective runners up that the elections were fraudulent in some way. Hillary Clinton claimed in 2016 that Donald Trump colluded with Russian actors to manipulate the results of the elections. Donald Trump claimed in 2020 that there was an organized attempt by Democratic Party officials to manipulate the election into Joe Biden's favor, after losing the elections despite what seemed to be a significant lead at the end of election night. The claims of Trump-Russia collusion culminated in the impeachment of Donald Trump, while Trump's claims of election fraud inspired protestors to storm the capitol. Investigations by the department of justice neither found sufficient evidence for the Trump-Russia collusion allegation nor for the election manipulation by the Democrats. Irrespective of the merits of the two claims of fraud, trust in US- elections appears to be seriously harmed as a poll conducted prior to the 2022 midterm-elections found out that 40% of Americans believe that the 2020-elections were fraudulent [1]. As trust in the election-system is crucial for a functioning democracy, methods to remedy trust-issues are important to investigate.

2. Blockchain and Trust

Blockchain technology is often associated with the term "trustless". This means that the architecture of blockchains is designed to eliminate the need for trust in any central

actor. Transactions on the blockchain are considered secure and tamper-proof through decentralized consensus mechanisms that make malicious behavior either too expensive or easily detectable.

Since its creation in 2009, the Bitcoin [2] blockchain has not experienced any significant successful attacks, and the Ethereum [3] blockchain has not seen any successful attacks on its consensus mechanism. However, some smart contracts deployed on the Ethereum blockchain have suffered from security vulnerabilities, such as exemplified in the DAO hack [4], where an attacker exploited a code error and stole millions worth of ether. When well-programmed, however, the Ethereum blockchain can effectively execute almost perfectly secure transactions. The success of blockchain technology has inspired proposals for using it in elections, some even published in scientific journals [4] [5]. However, it's important to note that the success of blockchains has mostly been limited to the financial sector and it's uncertain whether the same features that make blockchains effective for financial interactions can be transferred to elections.

In this paper, we describe a specific implementation of a published blockchain-based election system [7] and explain the modifications we made to the original design. We also examine specific trust issues raised in the aftermath of the 2016 and 2020 elections and explore how blockchain technology can address these concerns, and some additional problems facing elections on Ethereum. Finally, we present our view on the potential future of blockchain in elections.

3. An Implementation of an Election System on the Ethereum Blockchain

We deployed our election system on the Ethereum blockchain, which offers an advantage over Bitcoin in that it enables the deployment and execution of smart contracts. A smart contract is a piece of code that is stored on the Ethereum blockchain and executes automatically in accordance with its programmed rules. The Ethereum network utilizes a proof-of-stake (PoS) [8] [9] consensus mechanism to enforce the code of these smart contracts and reach consensus. In a PoS system, a group of validators, who hold a stake of the Ethereum cryptocurrency, are responsible for verifying transactions and maintaining the blockchain's integrity. These validators act as the

"enforcers" of the code in smart contracts and receive rewards for following the protocol and maintaining the network's security. However, if a validator is found to be violating the protocol or engaging in malicious activity, their stake is penalized, also known as "slashing". This provides a strong incentive for validators to act honestly and maintain the network's security, fulfilling the principles of trustlessness and decentralization.

Central building blocks of smart contracts are state-variables which store the current state of the smart contract on the blockchain. An example of a state-variable is the current vote-count of a particular candidate in an election. Interactions with smart contracts usually fall in one of two categories: Reading state-variables and changing state variables. Reading is free of charge, while changing state variables costs transactions fees as changes to the state variables of smart contracts must be verified by the validators and are recorded on the blockchain, making them immutable by any other means than calling functions of the smart-contracts. The consensus reached through the PoS mechanism guarantees the truth and security of the data recorded on the blockchain.

The election system that we developed consists of two smart contracts: a voter registration contract and a voting contract. Voters interact with the system through their own Ethereum wallet, which is essentially a pair of private and public keys used for signing transactions and sending them to the Ethereum blockchain for verification. To cast a vote, voters sign a transaction through their wallet and send it to the Ethereum blockchain, where the vote count is recorded in a state variable for the selected candidate. This ensures that all votes are recorded and counted in a secure and transparent manner, leveraging the benefits of blockchain technology and the PoS consensus mechanism.

3.1. Threat of a 51% Attack

Elections on Ethereum face potential threats from 51% attacks, where a malicious actor holding more than half of the staked ether could alter entries in the blockchain for their benefit. At the time of writing, the total value of staked ether was 28 billion USD [10], making it possible for a malicious actor to acquire such an amount. In PoS consensus is reached through voting on the accuracy of blocks. An attacker with over half of the staked ether can vote in favor of their own blocks, manipulating the vote-counts in favor of their preferred candidate. If the attack is successful, the other validators will be slashed as their version of the history will no longer match the majority. It's worth noting that although the attack could disrupt the election, it would not go unnoticed. The slashing of validators is a rare occurrence and the attack would be discovered, resulting in the attacker losing all their staked ether. A hard-fork would then be performed, creating a new branch of the Ethereum network where the attacker would get slashed instead of the validators. In conclusion, while a 51% attack may seem catastrophic, it is unlikely to actually influence the election result.

3.2. Voter Registration Contract

In the original publication [7], voters were identified via their phone number from their SIM card. Each SIM could only be registered once. The voters received a random code via SMS upon registration. The voter registry contract then stored the voter's phone number and code, and the voter called the registry contract with the code. If the code matched, the Ethereum address associated with the transaction was added to the list of eligible addresses, and the phone number was marked as used.

For our project, we implemented a different registration process. Eligible voters included all students and three instructors of the course. The main instructor sent an email to all eligible voters with a unique registration code. He then added the SHA-256 hash of the codes to a map on the registration contract. Participants then called the verification function of the contract with their code, and the function executed a SHA-256 encryption. If the hash matched one in the map, the Ethereum address associated with the transaction was added to the list of registered voters. The associated hash was added to the map of already registered voters, prohibiting reuse of the code.

3.3. Voting Contract

The voting contract only works in combination with the voter registration contract. On deployment, the address of the already deployed registration contract and a question for the election (e.g. "What is superior: chocolate or vanilla?"), the options, and start and end times must be provided. Voters cast their vote by calling the `voteFor` function of the voting contract. The registration contract is then called to verify that the voter's Ethereum address is listed among the registered addresses and to ensure that the voter has not already voted and that the current block time falls within the start and end times of the election. If all criteria are met, the state variable of the vote count for the selected candidate is updated by 1.

Like in most elections, the vote count is not disclosed until the election is over to prevent influencing the voters' decisions. We implemented this by making the vote count state variable for each candidate private. A private variable can only be accessed by a function within the contract. We added a getter function to the election contract to retrieve the vote count, that can only be successfully called once the election has ended. As we will discuss later in this paper, private state variables are vulnerable to attacks.

3.4. Interacting with the Election-System through a Decentralized Application

In principle smart contracts are independent entities registered on the Ethereum blockchain, executable independently of any particular user interface. Once the two contracts are deployed and their addresses and code made public, anyone with the knowledge of how to interact with smart contracts can use them. However, to make the system

user-friendly, usually user-interfaces called decentralized applications (dapps) are created to interact with the contracts. It should be noted though, that dapps can in principle be developed independently of the smart contracts they interact with. There can also be a multitude of competing dapps that interact with the same contracts.

Fig. 1 shows the voter verification page of our dapp. The voter enters his passcode, whose hash is checked against the hashes of registered voters, and if found, his wallet address is made eligible to vote.

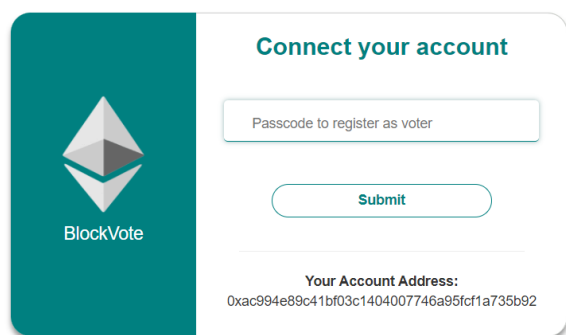


Fig 1: Voter verification page of the Dapp

4. Specific Trust issues raised by Hilary Clinton and Donald Trump

Most of the trust issues raised by Hillary Clinton and Donald Trump can be grouped into four categories: issues related to absentee or mail-in ballots, vote-counting procedures, voter identification, and technology related problems.

4.1. Absentee Ballots

Absentee or mail-in ballots allow voters to cast their votes remotely, rather than in person. To do this, the voter must typically request a mail-in ballot in advance and sign it with a registered signature that will be verified during the count. Mail-in ballots are collected in designated boxes. Donald Trump raised concerns about absentee ballots, as he was leading significantly on election night based on in-person votes. However, as the absentee ballots were continuously arriving and being counted, his lead slowly decreased and eventually vanished. Trump claimed that many of the votes that came in after election night were fraudulent.

4.2. Vote-Counting Procedures

Many of the fraud claims were a combination of concerns about vote-counting and absentee voting. Trump and his supporters were concerned about fraud at vote counting, as they accused officials of hindering Republican observers from observing the counts. The late arrival of mail-in ballots at counting venues also raised suspicions.

4.3. Voter Identification

Trump and his team claimed that many of the votes were not cast by eligible voters, either because the voters were allegedly deceased or because they were not citizens. The main point of mistrust was the process of voter registration, as some states allow for verification without a photo ID, such as signature identification.

4.4. Compromised Technology

Hillary Clinton's claim of election fraud falls into another category. She claimed that Trump colluded with Russian agents to interfere with the election in a digital form, by distributing false information or hacking servers of the Democratic Party and releasing compromising information (whether true or false). Similarly, Trump and others in the US and elsewhere raised concerns about the integrity of the voting machines used in some states, which could potentially be compromised.

5. How can Blockchain Address Trust Issues?

5.1. Absentee Ballots

One major concern about the integrity of mail-in voting arose from the observation that Trump's lead diminished when absentee ballots were counted. The proposed blockchain-based system addresses this issue by recording vote counts on the blockchain as soon as the votes are cast. In the smart contract, the vote counts for each candidate are recorded as a private state variable that is only accessible after the election has concluded. As in mail-in voting, there is no in-person voting in the blockchain-based system. The secrecy of the vote is not guaranteed in both mail-in-voting and blockchain voting when it comes to possible intimidation or selling of one's vote. People can be harassed into casting their votes for specific candidates or can expose their votes to a third party by choice. This issue can only be resolved by banning mail-in-voting and in the case of blockchain allowing the vote to take place only in an enclosed area. However, in a sense, this would go against the nature of blockchain voting.

5.2. Vote-counting

The key advantage of blockchain-based voting over traditional voting is in vote counting. The smart contract deterministically increases the value of the state variable when a new vote is transacted to the Ethereum blockchain. With no human intervention in the counting process, the vote count is secure and tamper-proof. However, there is no possibility for a recount, as there are no physical ballots to be counted. As the consensus mechanism on the Ethereum blockchain appears to be safe, there is no necessity for a recount option.

5.3. Voter Identification

Voter registration and identification varies from state to state and country to country. Some states and countries require Photo-IDs, while others only have signatures, which are verified either in person at the voting venue or in the case of absentee voting by adding the signature to the envelope or verifying one's ID at a post office. In the original paper, the verification via the phone number, but that would not be implemented this way in a real-life application. In our project, the main-instructor acted as a central trusted authority. Of course a central authority, has power to corrupt the voter registration, by adding hashes they themselves can use to vote multiple times. In a real-life application ID verification systems can be integrated with blockchain systems. When a person registers in the registration contract, there could be a necessary step in which the ID is also verified. Once The ID is verified a password/code could be both sent to the blockchain and the voter. The person registering then will only be registered if the code he receives is registered on the blockchain - of course separately. The safety of blockchain registration depends on the quality and safety of the identification process, blockchain or not. We neither see a superiority nor inferiority of the blockchain-based system compared to conventional voting.

The requirements for voter registration and identification vary depending on the state or country. Some require photo IDs, while others only require signatures that are verified in person or through the mail. In the original paper, voter verification was implemented via the phone number, but that would hardly be the case in a real election. Identification verification systems can be integrated with blockchain systems. When a person registers to vote in the registration contract, their ID must be verified, and a password or code is sent to both the voter and the blockchain. The voter is only registered if the code they receive is recorded on the blockchain. The security of an election depends on the quality and security of the identification process, regardless of whether it is blockchain-based or not. We estimate that the blockchain-based system and conventional voting, especially through mail-in voting, can have equal security and trust issues in the way voter identification is conducted.

5.4. Compromised Technology

The biggest vulnerability in a blockchain-based voting system is not the blockchain itself, but the front-end technology. Transactions must be sent to the Ethereum network to interact with the smart contracts on the blockchain, and most users will require a GUI to do so. This opens the door for potential attackers to deliver fraudulent front ends and intercept votes. The elderly are particularly vulnerable to such attacks, as they are often targets of email fraud. To prevent this, an official source of reliable front ends may be necessary, which goes against the decentralized

nature of blockchain-based voting. For this reason, we do not recommend using blockchain-based voting.

6. Further Problems to be Addressed

In the previous sections, we evaluated the benefits of the blockchain system in addressing trust issues. However, during the implementation of our system, we encountered additional problems that need to be addressed in future blockchain-based election systems.

6.1. Privacy Concerns

Blockchains like Ethereum and Bitcoin are decentralized and public by design. While state variables can be declared as private and not be accessible through interaction with the smart contract code, the smart contract-code along with the current state are still stored in bytecode. Each variable has a reserved slot, which can be deduced from the smart-contract code itself as variables are stored sequentially. [10] In our case the current vote counts could be accessed even though the variables in which they are stored, are private. There have been attempts to preserve privacy in Ethereum smart contracts but have been evaluated as expensive and overly complex to implement [11] for less experienced developers. One of the essential aspects of a free election is the secrecy of the vote, and the government should not be capable of knowing who a person voted for. However, Ethereum is not completely anonymous, and IP addresses can be discovered by analyzing blockchain data [12]. Each interaction with the election system should be secured through a proper VPN to conceal the voters' actual IP address.

6.2. Cost Issues

When registering or casting votes, voters must send transactions that require paying transaction fees. If the Ethereum network is not congested, the fees remain manageable within single digits. However, on election day, when many transactions are sent, the fees can become substantial. Transactions are processed by validators, and the more complex the operations, the higher the transaction fees become. Smart contracts that guarantee the security and privacy demanded by an election, will likely be quite complex and require high transaction fees. Requiring voters to pay fees to participate in an election is inherently undemocratic if the fees are substantial. To address this, one option is to refund voters from the election contract. Another option is to fund addresses that pay the transaction fees on behalf of voters. Ethereum does not natively allow for transaction fees to be paid by other addresses, but projects like OpenZeppelin have implemented decentralized solutions such as the Gas Station Network [12]. In a public election, addresses could be funded by tax-money to pay for the transaction fees on behalf of the voters.

7. Conclusion

Despite the potential of blockchain technology in finance, it does not seem to be a suitable solution for elections in its current form. The issue of trust in the voter registration process remains a challenge, with a blockchain-based system presenting similar challenges to traditional elections. Although blockchain technology excels in the vote counting process, the same cannot be said for voter registration. A possible solution could be a hybrid system, where voters register in person and cast their votes directly on the blockchain using their own devices. However, this still leaves room for suspicion and doubt. Technologically savvy individuals may understand the tamper-proof nature of transactions on the Ethereum blockchain, but this does not mean that those who are skeptical of the current election system will not have similar reservations about a blockchain-based system. In fact, the mistrust of technology in some circles may even increase such doubts. Privacy is also a major concern, as anonymity is not guaranteed on the Ethereum blockchain. To address these issues, a blockchain-based election system would need to be implemented on another more anonymous blockchain or a blockchain would have to be created specifically for the purpose of hosting elections.

8. References

- [1] R. & W. Strategies, 2022. [Online]. Available: <https://redfieldandwilsonstrategies.com/media-polls/40-of-americans-think-2020-election-was-stolen-just-days-before-midterms/>. [Accessed 4. February 2023].
- [2] S. Nakamoto, „Bitcoin: A Peer-to-Peer Electronic Cash System,“ 2008.
- [3] V. Buterin, „Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform,“ 2014.
- [4] V. Dhillon, "The DAO hacked," *blockchain enabled applications: Understand the blockchain Ecosystem and How to Make it work for you*, pp. 67-68, 2017.
- [5] R. Taş and Ö. Ö. Tanrıöver, "A systematic review of challenges and opportunities of blockchain for E-voting," *Symmetry*, vol. 12, no. 8, p. 1328, 2020.
- [6] U. Jafar and M. J. Aziz, "Blockchain for electronic voting system—review and open research challenges," *Sensors*, vol. 21, no. 17, p. 5874, 2021.
- [7] D. Khoury, E. F. Kfoury, A. Kassem and H. Harb, "Decentralized voting platform based on ethereum blockchain," in *IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, 2018.
- [8] S. King und S. Nadal, „Ppcoin: Peer-to-peer cryptocurrency with proof-of-stake,“ *Self-published paper*, 2012.
- [9] C. Smith, "Proof-Of-Stake," 12 January 2023. [Online]. Available: <https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/>. [Accessed 4 February 2023].
- [10] "Stakingrewards," [Online]. Available: <https://www.stakingrewards.com/earn/ethereum-2-0/>. [Accessed 5 February 2023].
- [11] A. Yardi, "How to access private data from a smart contract," [Online]. Available: <https://bitsbyblocks.com/how-to-access-private-data-from-a-smart-contract/>. [Accessed 4 February 2023].
- [12] A. Unterweger, F. Knirsch, C. Leixnering und D. Engel, „Lessons learned from implementing a privacy-preserving smart contract in ethereum,“ in *9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2018.
- [13] A. Biryukov und S. Tikhomirov, „Deanonymization and linkability of cryptocurrency transactions based on network analysis,“ in *In 2019 IEEE European symposium on security and privacy (EuroS&P)*, 2019.
- [14] O. Zeppelin, "Sending gasless transactions," [Online]. Available: <https://docs.openzeppelin.com/learn/sending-gasless-transactions>. [Accessed 4 February 2023].