On Secure E-Voting over Blockchain

PATRICK MCCORRY, Pisa Research Ltd, United Kingdoms
MARYAM MEHRNEZHAD, Newcastle University, United Kingdoms
EHSAN TOREINI, University of Durham, United Kingdoms
SIAMAK F. SHAHANDASHTI, University of York, United Kingdoms
FENG HAO, University of Warwick, United Kingdoms

This paper discusses secure methods to conduct e-voting over a blockchain in three different settings: decentralized voting, centralized remote voting and centralized polling station voting. These settings cover almost all voting scenarios that occur in practice. A proof-of-concept implementation for decentralized voting over Ethereum's blockchain is presented. This work demonstrates the suitable use of a blockchain not just as a public bulletin board, but more importantly, as a trustworthy computing platform that enforces the correct execution of the voting protocol in a publicly verifiable manner. We also discuss scaling up a blockchain-based voting application for national elections. We show that for national-scale elections the major verifiability problems can be addressed without having to depend on any blockchain. However, a blockchain remains a viable option to realize a public bulletin board, which has the advantage of being a "preventive" measure to stop retrospective changes on previously published records as opposed to a "detective" measure like the use of mirror websites.

CCS Concepts: • **Security and privacy**;

Additional Key Words and Phrases: e-voting, blockchain, Ethereum, boardroom voting, national elections

ACM Reference Format:
Patrick McCorry, Maryam Mehrnezhad, Ehsan Toreini, Siamak F. Shahandashti, and Feng Hao. 2021. On Secure E-Voting over Blockchain. *Digit. Threat. Res. Pract.* 1, 1, Article 1 (January 2021), 13 pages. https://doi.org/10.1145/3461461

1 INTRODUCTION

Secure voting methods underpin the integrity of democracy. In paper-based voting, tallying is a critical process where the winner is determined. Yet it is also the most vulnerable stage whereby a lack of transparency will allow tallying authorities (TAs) to (whether mistakenly or maliciously) modify, miscount or exclude a voter's ballot without their knowledge. A common countermeasure is to arrange public supervision attended by independent observers in the physical tallying procedure in the hope that any wrongdoing by the TAs would be observed and recorded.

In the digital era, there has been a trend of moving voting towards using electronic means, e.g., by using a touch-screen Direct Recording Electronic (DRE) machine at polling stations or casting votes from home through the Internet. However, when tallying is done electronically, hacking an election becomes much easier since it only takes flipping a few bits in the electronic tally to alter an election outcome. The traditional countermeasure through public supervision is no longer applicable when all votes are tallied in the computer memory. If there is

Authors' addresses: Patrick McCorry, Pisa Research Ltd, United Kingdoms, stonecoldpat@gmail.com; Maryam Mehrnezhad, Newcastle University, United Kingdoms, maryam.mehrnezhad@ncl.ac.uk; Ehsan Toreini, University of Durham, United Kingdoms, ehsan.toreini@durham.ac.uk; Siamak F. Shahandashti, University of York, United Kingdoms, siamak.shahandashti@york.ac.uk; Feng Hao, University of Warwick, United Kingdoms, feng.hao@warwick.ac.uk.

2576-5337/2021/1-ART1
https://doi.org/10.1145/3461461

a bug in the tallying software, or a hacker who has access to modify the software, the electronic tally or ballots can be changed without anyone knowing.

But digital technology is not all doom and gloom; it also brings new opportunities to secure an election in ways that were not possible with traditional paper ballots. For example, the use of digital technology, combined with cryptography, facilitates the design of voting systems that are publicly verifiable. Verifiability is typically defined as follows:

(1) **Cast-as-intended**: a voter is able to verify that their intended candidate is correctly captured in the cast vote.
(2) **Recorded-as-cast**: a voter is able to verify that their cast vote is correctly recorded by the system.
(3) **Tallied-as-recorded**: a voter (or any observer) is able to verify that all recorded votes are tallied correctly.

Voting systems that satisfy the above three properties are called end-to-end (E2E) verifiable [7]. In an E2E voting system, public verifiability is realized by the use of a receipt, which is provided to the voter after a voting session is finished. The receipt enables a voter to verify the integrity of the entire voting process, but does not allow the voter to prove to third parties how they have voted [10, 24]. Hence, voter privacy is also preserved in an E2E system. By comparison, traditional paper voting can only satisfy the first property – a voter marks a chosen candidate on a physical ballot with confidence that the ballot will be *cast-as-intended*. However, they cannot verify *by themselves* whether their cast ballots will be correctly recorded and tallied. Due to human error, malice or mischance, cast ballots may be lost or miscounted as reported before [14].

E-voting protocols that support public verifiability normally assume the existence of a public bulletin board that provides a consistent view to all voters. In practice, an example of implementing the public bulletin board can be seen in the yearly elections of the International Association of Cryptologic Research (IACR) [9]. They used the Helios voting system [1] whose bulletin board is implemented as a single web server. This server is trusted to provide a consistent view to all voters, however the server may misbehave if it is dishonest or compromised.

Many believe that blockchain, a public ledger that underpins the popular cryptocurrecies such as Bitcoin and Ethereum, holds the key to solving some key problems in e-voting. In practice, companies such as VoteWatcher (votewatcher.com), FollowMyVote (followmyvote.com) have proposed solutions that publish votes on a blockchain in plaintext. However, these solutions trivially reveal interim results before the end of voting, which can have an undesirable impact on voting behaviour.

In this paper, we investigate secure methods of running an election over a blockchain. Our contributions are as follows. First, we provide the first comprehensive analysis of secure e-voting over a blockchain for all three types of e-voting systems that cover decentralized boardroom voting, centralized internet voting and centralized polling station voting. Second, we present a proof-of-concept implementation of decentralized boardroom voting using Ethereum's blockchain to demonstrate the feasibility. Third, we discuss scaling up the implementation for national elections and highlight open research problems.

A preliminary conference version of the paper that reports a smart-contract implementation of decentralized boardroom voting over a blockchain is presented in [20]. This journal paper extends the previous conference paper by covering centralized remote voting and centralized polling station voting in addition to decentralized voting. The content of this paper forms the basis of a technical report that was submitted to the Economist Cybersecurity Challenge jointly organized by the Economist and Kaspersky Lab, and was ranked third place [15].

2 BACKGROUND

This section reviews the state-of-the-art in e-voting research and discusses the potential combination with a blockchain such as one that is currently being used by Ethereum.

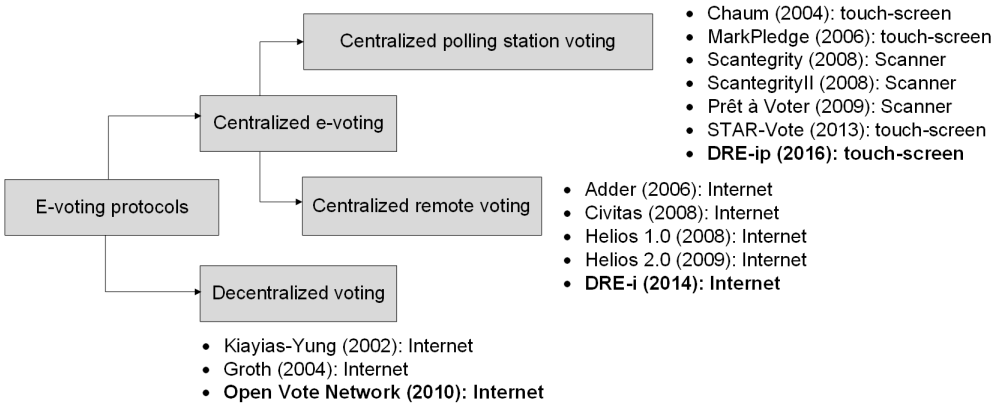

Fig. 1. Categorization of e-voting systems

2.1 Categorization of e-voting

Depending on how an election is organized, an e-voting system can be categorized into two types (see Figure 1). The first type is *decentralized voting*. In this setting, the election is essentially run by voters themselves. Votes are cast over a distributed network in several rounds before the final tally can be computed by every voter. Due to requiring multiple rounds of interaction, such systems are only suitable for small-scale elections, e.g., boardroom voting. Examples of existing voting systems include Kiayias-Yung [19], Groth [8] and Open Vote Network (OV-net) of Hao, Ryan and Zielínski [11]. Among the three, OV-net is the most efficient in terms of the number of rounds, computational efficiency and bandwidth usage. It only requires 2 rounds, fewer than other schemes. Within each round, each voter only needs to broadcast one group element, together with one zero-knowledge proof (ZKP) to prove that the published group element is well-formed. This is close to the best efficiency that one may hope for. We will discuss more details on OV-net in Section 4.

The second type is *centralized voting*. This corresponds to almost all practical large-scale elections where central administrations are involved. Depending on the implementation, a central facility is used to collect votes: it may be DRE machines in polling stations or a web server in Internet voting. The state-of-the-art research in the centralized setting includes voting systems that are End-to-End (E2E) verifiable. These systems can be further divided into centralized *remote* voting and centralized *polling station* voting. Examples of existing E2E voting systems include Adder [18], Civitas [6], Helios [2] and DRE-i [10] for centralized remote voting, and Votegrity [4], MarkPledge [22], Scantegrity [5], Prêt à Voter [23], STAR-vote [3] and DRE-ip [24] for centralized polling station voting. Among these systems, DRE-i and DRE-ip are the only two systems that are E2E verifiable without any TAs; they are also known as *self-enforcing e-voting* (SEEV) systems [10, 24]. An internet voting system using DRE-i has been used for classroom voting and student prize competitions [9]. A touch-screen based polling station voting system using DRE-ip was trialed in Gateshead, United Kingdom, on 2 May 2019 during the UK local elections with positive voter feedback [12].

It is worth nothing that in the centralized setting, whether an E2E voting system is used for local polling station voting or remote internet voting is mostly an implementation detail. Many E2E systems can be deployed for both with some adaptions. For example, although Helios was proposed and implemented as an Internet voting system, it is also possible to use it for polling station voting, e.g., by installing a browser-like voting client in the voting booth. Similarly, although the DRE-ip protocol has been implemented and trialled for polling station voting [12], the same protocol can be implemented for Internet voting as well.

2.2 Public bulletin board

All the protocols in Figure 1 require a public bulletin board that provides a consistent view to all voters. The publication of data on the bulletin board should be append-only; if published data are retrospectively changed, assurance on the integrity of the tallying results would be lost. However, a secure implementation of the bulletin board is a non-trivial challenge. The use of a single server for the bulletin board, as in the implementation of Helios [1], DRE-i [9] and DRE-ip [24], has the risk that if the server is dishonest or compromised, it might retrospectively modify the published data without being noticed. To address this issue, the published data should also be replicated in mirrored websites (which may be maintained by volunteers in distributed geographic locations). Thus, besides changing data on its own site, a dishonest server also needs to modify copies of data in mirrored sites. This is a lot harder, and can easily be detected by the public.

An alternative approach is by leveraging a blockchain. The purpose of a blockchain is to synchronize a set of replicated ledgers kept by mutually distrustful peers such that all peers reach consensus on the public ledger's contents. At first glance, Bitcoin's blockchain is a potentially attractive candidate for an e-voting public bulletin board as it is censorship-resistant with an immutability property that prevents data modification once it is stored. Unfortunately, Bitcoin's limited programming capability makes it difficult to enforce the voting protocol's correct execution. This critical limitation makes us turn to Ethereum.

2.3 Ethereum

Among several competing blockchain systems, Ethereum's blockchain seems the most promising as a platform for an e-voting bulletin board. Ethereum is the second most popular cryptocurrency with a $40.6 market capitalization as of November 2020. Its blockchain is considered an ordered transaction-based state machine. Programs ('smart contracts') can be stored and executed over the blockchain using user-authorized transactions.

Above all, smart contracts offer an expressive programming language that allows us to directly implement our e-voting protocol using cryptography. The execution of these contracts is enforced by *consensus computing* as every validating peer in Ethereum's underlying peer-to-peer network must repeat its execution to reach consensus on the blockchain's contents. Once data is published on a blockchain, the chance that it will be retrospectively modified decreases exponentially after more blocks are appended. As compared to the *detective* measure as seen in mirrored website, a blockchain potentially provides a *preventive* measure to stop data being modified in the first place. Furthermore, each peer in the Ethereum network has built-in computing facility to perform publicly verifiable operations. As such, Ethereum offers more than just a bulletin board as its consensus protocol can also enforce the correct execution of an e-voting protocol.

3 A HIGH-LEVEL VISION

The role of trustworthy authorities in ensuring the tallying integrity of an election exists in many voting systems. In traditional paper-based voting, after the voter puts the ballot into the box, they lose possession of the ballot and need to trust election authorities to faithfully record and tally votes. A chain of custody is required to ensure the integrity of the paper ballots during the collection, transportation and counting processes. Despite this, there are still many cases of missing or miscounted paper ballots reported in the past [14]. Today, e-voting products such as DREs used in USA, India, Brazil and many other countries mostly work like a blackbox and the public cannot independently verify the tallying results. This requires even stronger trust on authorities to assure the integrity of system, e.g., through the government certification process.

Over the past two decades, researchers have proposed to apply cryptography to build voting systems that are end-to-end verifiable. Mimicking the role of trusted authorities in paper-based counting, most of the proposed E2E voting systems assume tallying authorities (TAs), who are trustworthy individuals with computing and cryptography expertise to perform the tallying operation. However, voters must trust the TAs do not collude all

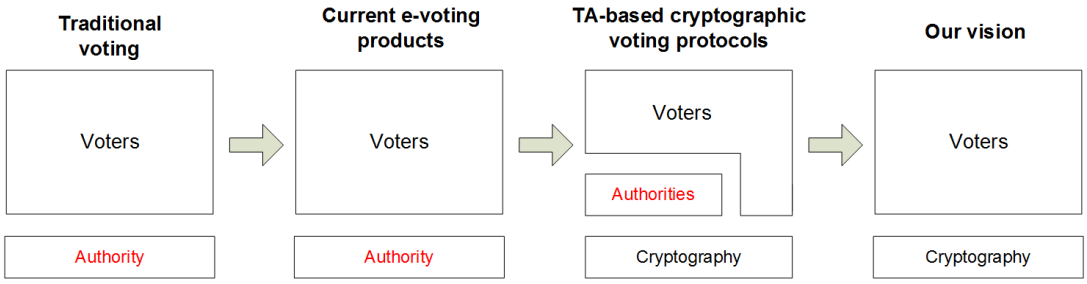

Fig. 2. A vision of removing trusted tallying authorities

together as a full collusion will allow TAs to trivially learn every individual vote. The fact that TAs have such power can present a deterring effect on some voters for the concern on privacy. In a real-world election, besides TAs, there are other authorities in various roles, e.g., those who are responsible for setting up ballot choices, registering voters and performing voter authentication. However, finding and managing tallying authorities in a secure and distributed manner has proved particularly difficult, as reported in real-life experience of trial elections [2].

The recent works on E2E e-voting systems [10, 24] show, counter-intuitively, that E2E verifiability is achievable without involving TAs; in other words, an e-voting system can be "self-enforcing". Such TA-free voting systems are unimaginable with the traditional paper-based methods, but become possible thanks to the digital technologies. We thus envision that "self-enforcing e-voting" systems will play an important role in future elections. This vision is highlighted in Figure 2.

One important factor for the popularization of cryptocurrencies such as Bitcoin is attributed to their decentralized blockchain, which as a public ledger is free from control by central banks or financial institutions. In other words, a blockchain is *self-enforcing*. The philosophy of removing trusted authorities in maintaining the public ledger in Bitcoin echoes that of removing trusted authorities in administrating the tallying process in self-enforcing e-voting systems. The two technologies appear to fit naturally.

Hence, we set out to explore the feasibility of running *self-enforcing* e-voting protocols over a *self-enforcing* blockchain. Running a self-enforcing e-voting system over a blockchain has several advantages. First, instead of simply publishing the voting data as a bulletin board, a blockchain provides a trustworthy computing platform to verify the data in real-time. Thus any malformed cryptographic data can be caught instantly, which effectively turns the traditional *detection* measure in E2E voting systems into a *prevention* measure. Second, blockchain makes it possible to automate the execution of a voting protocol in a timely and verifiable manner through *smart contracts*. Finally, blockchain supports a deposit-and-refund mechanism that can be built into the smart contract as an incentive to encourage honest behavoiur of all parties involved in the voting. In the next section, we will present a proof-of-concept implementation to further demonstrate these advantages.

4 BOARDROOM VOTING AND IMPLEMENTATION

We start by studying the simplest setting: decentralized voting for a small-scale boardroom election. In this setting, we implement the OV-net protocol over Ethereum as a smart contract. This is the first implementation of a boardroom voting protocol with maximum voter privacy over Ethereum's blockchain.

4.1 Open Vote Network Protocol

Open Vote Network (OV-net) is a decentralized two-round voting scheme [11]. It is self-tallying, meaning that every voter is able to tally votes by themselves. Voter privacy is protected at the maximum as each voter is limited to learn nothing more than their own input and the total tally. The protocol assumes a public authenticated channel for every voter to cast votes.

For a single candidate election with a Yes/No choice, the protocol can be described as follows (a single-candidate election can be easily generalized to support multiple candidates [11]). First, all n voters agree on (G, g) where G is a cyclic group of prime order q, and g is a generator in G. All modular operations are performed with respect to a large prime modulus p, where $q \mid p-1$. Each voter P_i chooses a secret value x_i uniformly at random from $[0, q-1]$.

Round 1: every voter P_i publishes g^{x_i} and a Schnorr Zero Knowledge Proof (ZKP) for proving the knowledge of x_i. At the end of this round, every voter validates all ZKPs, and computes: $g^{y_i} = \Pi_{j<i} g^{x_j} / \Pi_{j>i} g^{x_j}$.

Round 2: every voter P_i publishes $g^{x_i y_i} g^{v_i}$ and a one-out-of-two ZKP for proving that v_i is either 0 or 1 (for No and Yes respectively). At the end of this round, anyone who observes the protocol can tally the number of Yes votes by computing: $\Pi_i g^{x_i y_i} g^{v_i} = g^{\Sigma_i x_i y_i} g^{\Sigma_i v_i} = g^{\Sigma_i v_i}$.

The above protocol works based on the cancellation of random factors at the tallying phase, i.e., for any set of x_i, when y_i is defined as $y_i = \Sigma_{j<i} x_j - \Sigma_{j>i} x_j$ (as per Round 1), we have $\Sigma_i x_i y_i = 0$. From $g^{\Sigma_i v_i}$, anyone can compute the tally $\Sigma_i v$ by exhaustive search. No tallying authorities are required.

In this protocol, all communication is public. No secret channels between voters is required. This makes it suitable for implementation over Ethereum, as we explain below.

4.2 Structure of Implementation

The implementation of OV-net over Ethereum consists of two smart contracts, which are both written in Ethereum's Solidity language. The first contract is called the voting contract. It implements the voting protocol, controls the election process and verifies the two types of zero knowledge proofs. The second contract is called the cryptography contract. It distributes the code for creating the two types of zero knowledge proofs. This provides all voters with the same cryptography code that can be used locally without interacting with the Ethereum network. The code for both contracts is freely available as open source[1]. Three HTML5/JavaScript pages are developed in this implementation:

- *Election administrator* (via `admin.html`) administers the election. This includes establishing the list of eligible voters, setting the election question, and activating a list of timers to ensure the election progresses in a timely manner. The latter includes notifying Ethereum to begin/close registration, to begin/close voting and to compute the tally.
- *Voter* (via `vote.html`) can register for an election, and once registered can cast their vote.
- *Observer* (via `livefeed.html`) can watch the election's progress, including the election administrator starting and closing each stage and voters registering and casting votes. The running tally is not computable until the last vote is cast, due to the cryptographic design of the protocol.

It is assumed that voters and the election administrator have their own Ethereum accounts. The Web3 framework provided by the Ethereum Foundation is used to facilitate communication between a user's web browser and their Ethereum client. The user can unlock their Ethereum account (decrypt their Ethereum private key using a password) and authorize transactions directly from the web browser. There is no need for the user to interact with an Ethereum wallet and the Ethereum client can run in the background as a daemon.

[1] https://github.com/stonecoldpat/anonymousvoting

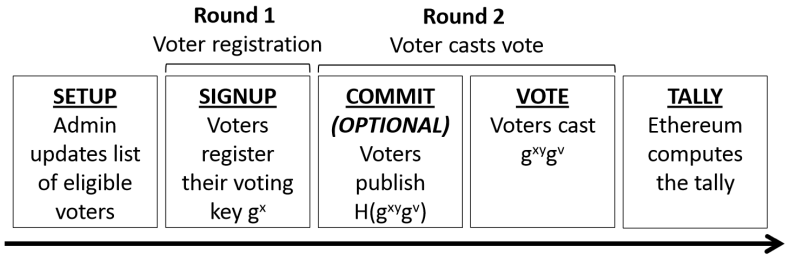

Fig. 3. Five election stages in the Open Vote Network

4.3 Election stages

The protocol is executed in five stages as shown in Figure 3. A list of timers is enforced by the smart contract to ensure that the election progresses in a timely manner. The contract only allows eligible voters to register for an election, and only registered voters can cast a vote. Furthermore, the contract can require each voter to deposit ether (the internal network currency in Ethereum) upon registration, and automatically refund the ether when their vote is accepted into the blockchain. Each stage of the election is described in detail below.

SETUP. The election administrator authenticates each voter with their user-controlled account and updates the voting contract to include a whitelist of accounts as eligible voters. The administrator then defines a list of timers to ensure that the election progresses in a timely manner in subsequent stages. Finally, the administrator sets the registration deposit, the voting question, whether the optional COMMIT stage should be enabled, and notifies Ethereum to transit to the next stage.

SIGNUP. All eligible voters can choose to register for the vote after reviewing the voting question and other parameters set by the election administrator. To sign up, the voter sends the first round OV-net data to Ethereum alongside a deposit. The election administrator is responsible for notifying Ethereum to transit to either the optional COMMIT or the VOTE stage. All g^{y_i} are computed by Ethereum during the transition.

COMMIT (optional). All voters publish a hash of the second round data of OV-net to Ethereum as a commitment. The contract automatically transits to the VOTE stage once the final commitment is accepted into the blockchain.

VOTE. All voters publish the second round OV-net data. The deposit is automatically refunded to the voter when their vote is accepted by Ethereum. The election administrator notifies Ethereum to compute the tally once the final vote is cast.

TALLY. Ethereum computes the tally and posts it on the blockchain.

Note that the last voter in round 2 of OV-net can compute the tally before others [11], and depending on the computed outcome, might change their vote. The optional COMMIT stage addresses this issue by requiring all voters to commit to their votes before revealing them in the VOTE stage. The deposit in this implementation provides a financial incentive for registered voters to vote and is returned to the voter upon successfully casting their vote. The timestamps defined by the election administrator determine if the voter's deposit is forfeited or refunded. Forfeited deposits may go directly to selected charities, which are specified in the smart contract.

4.4 Experiments on Ethereum's test network

The proof-of-concept implementation was deployed on Ethereum's official test network that mimics the production network. There were 126 transactions sent to simulate forty voters participating in this protocol. In the actual deployment, the implementation had to be split into two contracts as the code was too large to store in an Ethereum block. The voting contract VoteCon (80% of block capacity and $0.83 transaction fee) contains the protocol logic. The cryptography contract CryptoCon (52% of block capacity and $0.54 transaction fee) contains

the code to create and verify the two types of zero knowledge proofs in the protocol. Note that creating the ZKPs is run locally on the voter's machine.

4.5 Cost analysis

A breakdown of costs for 40 participants using OV-net is summarized in Table 1. The cost in USD ($) is approximated using the conversion rate of 1 ether = $11 and 1 gas = 0.00000002 ether (the rates when the experiments were conducted). Here a gas is a unit that measures the amount of computation required to execute certain operations over the Ethereum blockchain.

Overall, running the election with 40 voters costs the election administrator $2.74. The total election cost is $31.98, which breaks down to a cost of $0.73 per voter. As the number of voters increases, the election administrator's cost increases linearly, while the voter's cost remains constant. The total cost averaged over the number of voters stays approximately the same.

Entity: Transaction	Cost in Gas	Cost in $
A: VoteCon	3,779,963	0.83
A: CryptoCon	2,435,848	0.54
A: Eligible	2,153,461	0.47
A: Begin Signup	234,984	0.05
V: Register	763,118	0.17
A: Begin Election	3,085,449	0.68
V: Commit	70,112	0.02
V: Vote	2,490,412	0.55
A: Tally	746,485	0.16
Administrator Total	12,436,190	2.74
Voter Total	3,323,642	0.73
Election Total	**145,381,858**	**31.98**

Table 1. A breakdown of the costs for 40 participants using OV-net. The cost for the election administrator is identified as 'A' and the voter as 'V'.

4.6 Timing analysis

Table 2 outlines the timing measurements for operations in OV-net. All measurements were performed on a MacBook Pro running OS X with 4 cores, 2.3 GHz Intel Core i7 and 16 GB RAM, and are rounded up to the next whole millisecond. Computation times for all tasks are measured on the local daemon. Overall, ZKP operations are the most computationally expensive components in the protocol execution.

4.7 Limitations

While our proof-of-concept implementation shows that it is feasible to run small-scale boardroom voting in a decentralized setting over a blockchain, there are a few limitations. First, since each voter uses their own computer to generate secret keys, they naturally need to trust their own computing device. For example, if the device is compromised, the voter may be presented with a misleading view. Also, if the secret key is not generated uniformly at random, the guarantee on vote privacy will be lost. Second, if voters reveal the local secrets to a third party (say a coercer), they can prove how they have voted. Third, each voter needs to have an Ethereum account, and needs to pay a deposit (which will be refunded if the voter completes voting). However, ordinary

Action	Avg. Time (ms)
Create Schnorr ZKP	81
Register voting key	142
Begin election	277
Create 1-out-of-2 ZKP	461
Cast vote	573
Tally	132

Table 2. A time analysis for operations that run on the Ethereum daemon.

voters may not have an Ethereum account. Finally, if some voters refuse to send votes in the second round of OV-net, they will lose their deposits, but the tally will fail to work. It is still possible to compute the tally by adding a recovery round [16], however, a recovery operation requires all the remaining voters to cooperate all together, and will fail if some drop out halfway in the recovery process. These are inherent limitations in a totally decentralized voting system. They may be less an issue for small-scale voting in a boardroom, but can prove problematic for national scale elections, which we will discuss next.

5 NATIONAL ELECTIONS

A number of researchers have investigated running a national election over a blockchain [13, 17, 25, 27]. Obviously, running a national election is completely different from running decentralized boardroom voting since national elections are inherently centralized. Almost all of the proposed solutions so far involve using a blockchain as an immutable storage device to save digital ballots in plaintext. This is also what several companies (e.g., VoteWatcher, FollowMyVote) are proposing. However, as explained in a consensus study report published by National Academies of Sciences, Engineering, and Medicine [21], storing ballots on a blockchain in plaintext does not provide any ballot secrecy and as such the system is trivially subject to coercion and vote selling. The report concludes: "The use of blockchains in an election scenario would do little to address the major security requirements of voting, such as voter verifiability". However, this report does not elaborate how the major security requirements of voting can be met in a national election. Here we further investigate this subject from a system's perspective with the blockchain being one possible part of the system. This allows us to study how the verifiablity requirements can be fulfilled in a national election and the exact role that a blockchain might play in the overall system.

To start with, one may ask whether we can apply the proof-of-concept implementation in Section 4 to larger-scale elections such as national elections. Unfortunately, a straightforward application will not work for two reasons. First, the OV-net protocol that underpins our implementation is designed for decentralized voting, while national elections are inherently centralized. The protocol requires several rounds of interactions among voters, which may not be an issue in boardroom voting, but is not realistic in larger elections. In a national election, a voter should just be able to cast a vote without having to interact with other voters. Second, using Ethereum as deployed today, only one vote can be cast in each block due to reaching the block's gas limit. Given that each block is generated every 12 seconds, this means only five votes per minute can be cast over the blockchain. This may be sufficient for boardroom voting, but is inadequate for national elections. Take the 2016 UK referendum as an example. For 5.2 million postal votes in that election, it would require 722 days for all votes to be recorded into Ethereum's blockchain. Supporting a national election requires addressing these two major limitations accordingly.

To address the first problem, a centralized e-voting protocol should be used instead of OV-net. Depending on whether the voting environment is "supervised" or not, there are two settings: centralized remote voting

("unsupervised") and centralized polling station voting ("supervised"). In general, E2E verifiability is a standard requirement for secure voting protocols in both settings. We refer the reader to an edited book on " Real-World Electronic Voting" [7] for a review of the existing E2E e-voting protocols and testing in real-world elections. While many of the E2E protocols are applicable, we choose DRE-ip [24]. As compared with other related E2E protocols, DRE-ip does not involve any TAs, and hence is much simpler and more efficient to manage. As compared with DRE-i [10], DRE-ip computes encrypted ballots in real time rather than pre-computing ballots before an election as in DRE-i, and as such provides a higher privacy guarantee than DRE-i – in case the voting system is completely compromised, the information leakage will be limited to only the partial tally at the time of compromise, which is minimum. The same DRE-ip protocol can be implemented for both local polling station voting and remote Internet voting [24].

To overcome the second problem, a dedicated blockchain will likely be needed as the throughput of the existing Ethereum's blockchain is not scalable enough for a national election. As an example, if we choose DRE-ip to provide E2E verifiability in large-scale elections, the voting system will need to publish encrypted ballots on a public bulletin board to enable public verifiability. A blockchain is one way to implement such a public bulletin board and to ensure that published data cannot be retrospectively changed. Creating a dedicated blockchain with high throughput (e.g., larger blocks, faster rates) for a national election is theoretically possible but a harder problem is creating an ecosystem in which a sufficiently large number of distributed miners are incentivized to mine blocks and maintain the blockchain. An alternative way to implement a public bulletin board is by using mirrored websites, where encrypted ballots are published at an election website and mirrored on other websites in distributed places. This follows a similar practice of distributing software over the Internet by using mirror websites. The main difference between the two approaches is that a blockchain can be seen as a "preventive" measure to stop modification of data by design, while the use of mirrored websites can be seen as a "detective" measure.

6 COMPARISON

In this section, we compare our proposed voting methods with the real-world voting applications and related voting schemes in the literature. For fairness, we compare systems within their respective settings. Table 3 summarizes the comparison results in terms of decentralized voting, centralized remote voting and centralized polling station voting.

Decentralized voting. In this setting, voting is decentralized and there is no central facility to collect votes. A classic example is a simple *show of hands*. Everyone can tally the votes, but voters have no privacy, and they may be subject to coercion. Our proposal for this setting is to implement OV-net over blockchain, denoted as *OV-net/BC*. OV-net is an e-voting protocol executed over a distributed network. It provides the maximum protection on the voter's privacy under the condition that a voter uses a trustworthy voting device [11]. With a trustworthy voting device, a voter can verify that their vote is cast as intended and recorded as cast. The system is self-tallying, so anyone can compute the tally without needing any tallying authorities. However, OV-net does not offer coercion resistance. This is because voting is conducted in an unsupervised environment. A voter may vote while a coercer is standing over their shoulder. A voter can also prove how they have voted by revealing the ephemeral secrets cached on their computer. For this reason, OV-net/BC is only suitable for small-scale boardroom voting.

Centralized remote voting. In this setting, votes are cast remotely into a central facility, which may be a specified mailing address for postal voting, or a web server for centralized Internet voting. In postal voting, voter privacy is typically protected by using double envelopes, where the outer envelope displays voter identifying information for authentication while the inner one contains only the secret vote. In other words, the voter privacy is fulfilled

Scheme	Decentralized (boardroom) voting		Centralized remote voting			Centralized polling station voting				
Criteria	Show of hands	OV-net/BC*	Postal	TA-based E2E/BB	TA-free E2E/BB*	Paper	DRE	DRE+ VVPAT	TA-based E2E/BB	TA-free E2E/BB*
Voter privacy	✗	☑	☑	☑	☑	☑	☑	☑	☑	☑
Voter can check if their vote is cast as intended	✓	☑	✓	✓	✓	✓	✗	✓	✓	✓
Voter can check if their vote is recorded as cast	✓	☑	✗	✓	✓	✗	✗	✗	✓	✓
Anyone can check if all votes are tallied as recorded	✓	✓	✗	✓	✓	✗	✗	✗	✓	✓
Receipt does not reveal voter's choice	N/A	N/A	N/A	✓	✓	N/A	N/A	N/A	✓	✓
Coercion resistant	✗	✗	✗	✗	✗	☑	☑	☑	☑	☑
Free from TA	✓	✓	✗	✗	✓	✗	✗	✗	✗	✓
Suitable for large-scale voting	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓

Table 3. Comparison between voting systems within respective settings. BC: Blockchain. BB: Bulletin Board. TA: Tallying authority. *: Our proposed solution. Examples of TA-based E2E include Helios [2], Scantegrity [5], Votegrity [4], and Prêt à Voter [23]. Examples of TA-free E2E (or self-enforcing e-voting) schemes include DRE-i [10] and DRE-ip [24]. In this paper, we choose DRE-ip for both centralized remote voting and polling station voting. ✗: not fulfilled. ✓: fulfilled. ☑: fulfilled under conditions. N/A: not applicable.

under the condition that the voter's identity is not linked to the cast ballot (i.e., voting is anonymous). After the voter posts the ballot, they cannot check if their vote will be recorded and tallied correctly. They need to trust the integrity of the postal system and tallying authorities in the receiving end to correctly record and tally their votes. Our proposal for this setting is to use E2E verifiable e-voting systems over a secure bulletin board (BB). A BB may be implemented by a blockchain or mirror websites. Among E2E schemes, we choose DRE-ip as it is free from tallying authorities and is simpler to manage. An E2E voting system, such as DRE-ip, is designed to protect the tallying integrity even if the voting system is completely compromised. In an E2E voting system, the voter privacy is generally protected by ensuring that each authenticated voter obtains a random credential not linked to their real identity. In other words, the voter privacy is fulfilled under the condition that voting is anonymous. An E2E voting system, by definition, allows a voter to verify their vote is cast as intended (CAI), recorded as cast (RAC), and tallied as recorded (TAR). In DRE-ip (and other E2E systems), the E2E verifiability is realized by issuing each voter a receipt, which allows the voter to verify CAI and RAC but without being able to prove to any third party how they have voted. In practice, the voter receipt can be delivered by using SMS or email (or printed on physical paper in a polling station) as long as it cannot be retrospectively changed by the voting system. The verification on TAR can be performed by anyone who has access to the audit data published on the public bulletin board. As the voting environment is "unsupervised", all systems in this category are subject to coercion (e.g., a coercer watches over a voter's shoulder when they cast a vote), hence are only suitable for elections where the threat of coercion is low.

Centralized polling station voting. In this setting, votes are cast in central polling stations in a supervised environment. Traditional paper-based voting, though commonly perceived by the public as secure, do not allow

voters to verify *by themselves* that their votes are correctly recorded as cast and tallied as recorded. Voters need to trust third parties for the integrity of the recording and tallying process. A plain DRE-based voting system is much worse, as the machine may record a different candidate than the intended one without the voter's knowledge. This issue can be addressed by adding a Voter Verifiable Paper Audit Trail (VVPAT), but the resultant system still does not allow the voter to check by themselves if their vote is correctly recorded and tallied. Since VVPAT leaves physical audit trail for the cast ballots, this makes it possible in principle to verify the electronic tally by comparing it against the full audit trail or a subset of it (risk-limiting auditing [26]). This assumes that the integrity of VVPAT is protected through a chain of custody, and more importantly, that the paper records are actually checked. However, in practice, VVAPT is rarely checked. For example, states in the US only allow checking VVPAT when there is evidence of fraud, which is a high bar (see the unsuccessful petitions asking to verify VVPAT of certain e-voting machines following the 2016 US presidential election[2].

In a polling station, the voter privacy is protected by ensuring that voting is anonymous in a private voting booth. This requires appropriate physical and procedural measures in place to ensure the security of the polling station. The requirement of a secure polling station applies to all voting methods in this setting. Our proposal is to use DRE-ip over a secure bulletin board. As before, the use of a blockchain is one possible way to implement a bulletin board. As compared with alternative E2E schemes such as Votegrity [4] and Prêt à Voter [23], our proposal does not require any TAs, which contributes to greatly simplifying the election management. Same as other E2E voting systems, DRE-ip guarantees the tallying integrity even if the voting machines are completely compromised [24]. The E2E verifiability (CAI, RAC and TAR) is realized without having to trust the voting system.

7 CONCLUSION

In this paper, we systematically investigated running an election over a blockchain in three different settings: namely, decentralized voting, centralized remote voting and centralized polling station voting. We proposed solutions for each setting. In the decentralized voting setting, we presented a smart-contract implementation of OV-net over Ethereum's blockchain, and demonstrated the feasibility to run a small-scale boardroom election using an existing Ethereum blockchain. In the centralized remote/onsite voting settings, we proposed to use an E2E verifiable e-voting scheme, in particular DRE-ip, over a secure bulletin board. The use of a blockchain represents one possible way to implement such a bulletin board. However, we showed that there was an alternative way of using mirrored websites to implement a bulletin board. While a blockchain provides a "preventive" measure to stop retrospective modification of data by design, the use of mirrored website provides a "detective" measure with a different cost-benefit trade-off.

ACKNOWLEDGMENT

We thank anonymous reviewers for useful and constructive comments. This work is partly funded by ERC Starting Grant, No. 306994 and Royal Society grant, No. ICA/R1/180226.

REFERENCES

[1] Ben Adida. Helios: Web-based open-audit voting. In *USENIX security symposium*, volume 17, pages 335–348, 2008.

[2] Ben Adida, Olivier De Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Electing a university president using open-audit voting: Analysis of real-world use of helios. In *Proceedings of the Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE)*. USENIX Association, 2009.

[2]https://www.washingtonpost.com/news/post-nation/wp/2016/11/22/the-department-of-justice-is-not-going-to-conduct-a-vote-audit-based-on-your-phoned-in-outrage/

[3] Susan Bell, Josh Benaloh, Michael D. Byrne, Dana DeBeauvoir, Bryce Eakin, Gail Fisher, Philip Kortum, Neal McBurnett, Julian Montoya, Michelle Parker, Olivier Pereira, Philip B. Stark, Dan S. Wallach, and Michael Winn. STAR-Vote: A secure, transparent, auditable, and reliable voting system. *USENIX Journal of Election Technology & Systems*, 1(1):18–37, 2013.

[4] David Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE security & privacy*, 2(1):38–47, 2004.

[5] David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L Rivest, Peter YA Ryan, Emily Shen, and Alan T Sherman. Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems using Invisible Ink Confirmation Codes. *EVT*, 8:1–13, 2008.

[6] Michael R Clarkson, Stephen Chong, and Andrew C Myers. Civitas: Toward a secure voting system. In *IEEE Symposium on Security and Privacy*, pages 354–368. IEEE, 2008.

[7] Feng Hao and Peter Y A Ryan (Eds). *Real-world Electronic Voting: Design, Analysis and Deployment.* Series in Security, Privacy and Trust. CRC Press, 2016.

[8] Jens Groth. Efficient maximal privacy in boardroom voting and anonymous broadcast. In *International Conference on Financial Cryptography*, pages 90–104. Springer, 2004.

[9] Feng Hao, Dylan Clarke, Brian Randell, and Siamak F Shahandashti. Verifiable classroom voting in practice. *IEEE Security & Privacy*, 16(1):72–81, 2018.

[10] Feng Hao, Matthew N Kreeger, Brian Randell, Dylan Clarke, Siamak F Shahandashti, and Peter Hyun-Jeen Lee. Every vote counts: Ensuring integrity in large-scale electronic voting. *The USENIX Journal of Election Technology and Systems*, 1, 2014.

[11] Feng Hao, Peter YA Ryan, and Piotr Zielinski. Anonymous voting by two-round public discussion. *IET Information Security*, 4(2):62–67, 2010.

[12] Feng Hao, Shen Wang, Samiran Bag, Rob Procter, Siamak F Shahandashti, Maryam Mehrnezhad, Ehsan Toreini, Roberto Metere, and Lana Liu. End-to-end verifiable e-voting trial for polling station voting at gateshead. *IEEE Security & Privacy*, 2020.

[13] Friðrik Þ Hjálmarsson, Gunnlaugur K Hreiðarsson, Mohammad Hamdaqa, and Gísli Hjálmtýsson. Blockchain-based e-voting system. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 983–986. IEEE, 2018.

[14] Douglas Jones and Barbara Simons. *Broken ballots: Will your vote count?* CSLI Publications Stanford, 2012.

[15] Kaspersky Lab. Can Blockchain Technology Secure Digital Voting Systems? https://www.kaspersky.co.uk/blog/cybersecurity-case-study/8067/. [Online; accessed 16-April-2021].

[16] Dalia Khader, Ben Smyth, Peter Ryan, and Feng Hao. A fair and robust voting system by broadcast. *Lecture Notes in Informatics (LNI), Proceedings-Series of the Gesellschaft fur Informatik (GI)*, pages 285–299, 2012.

[17] David Khoury, Elie F Kfoury, Ali Kassem, and Hamza Harb. Decentralized voting platform based on ethereum blockchain. In *2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, pages 1–6. IEEE, 2018.

[18] Aggelos Kiayias, Michael Korman, and David Walluck. An internet voting system supporting user privacy. In *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual*, pages 165–174. IEEE, 2006.

[19] Aggelos Kiayias and Moti Yung. Self-tallying elections and perfect ballot secrecy. In *International Workshop on Public Key Cryptography*, pages 141–158. Springer, 2002.

[20] Patrick McCorry, Siamak F Shahandashti, and Feng Hao. A smart contract for boardroom voting with maximum voter privacy. In *International Conference on Financial Cryptography and Data Security*, pages 357–375. Springer, 2017.

[21] National Academies of Sciences, Engineering, and Medicine and others. *Securing the Vote: Protecting American Democracy.* National Academies Press, 2018.

[22] C Andrew Neff. Practical high certainty intent verification for encrypted votes, 2004.

[23] Peter YA Ryan, David Bismark, James Heather, Steve Schneider, and Zhe Xia. Prêt à voter: a voter-verifiable voting system. *IEEE transactions on information forensics and security*, 4(4):662–673, 2009.

[24] Siamak F Shahandashti and Feng Hao. DRE-ip: a verifiable e-voting scheme without tallying authorities. In *European Symposium on Research in Computer Security*, pages 223–240. Springer, 2016.

[25] Shalini Shukla, AN Thasmiya, DO Shashank, and HR Mamatha. Online voting application using ethereum blockchain. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 873–880. IEEE, 2018.

[26] Philip B Stark. Risk-limiting vote-tabulation audits: The importance of cluster size. *Chance*, 23(3):9–12, 2010.

[27] Emre Yavuz, Ali Kaan Koç, Umut Can Çabuk, and Gökhan Dalkılıç. Towards secure e-voting using ethereum blockchain. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pages 1–7. IEEE, 2018.