Requirement Analysis of Some Blockchain-based E-voting Schemes

Latif Anıl Büyükbaskın[1] and Isa Sertkaya[1,2]

[1]Cybersecurity Engineering, Graduate School of Natural and Applied Sciences, Istanbul Sehir University, Turkey.
[2]MCS Labs & BCLabs, TÜBİTAK BİLGEM UEKAE, Kocaeli, Turkey,
e-mail: latifbuyukbaskin@std.sehir.edu.tr, isa.sertkaya@tubitak.gov.tr

ORCID iD: 0000-0003-3895-5307, 0000-0002-4739-0515

Research Paper Received: 08.01.2020 Revised: 18.08.2020 Accepted: 19.12.2020

Abstract—Today, developing technology is one of the most effective tools to make our lives easier. One of these developing technologies is blockchain that enables securely transferring digital assets between peers without requiring a trusted third party. In particular, blockchain poses new opportunities to effectively satisfy transparency, verifiability and anonymity for e-voting schemes. Based on recent proposals, it can be easily seen that applicability of blockchain technology for e-voting systems is actively researched. In this paper, we first summarized the set of e-voting requirements based on studies by Popoveniuc et al., Fujioka et al., Cranor et al., Benaloh et al., Juels et al. and etinkaya et al. In the light of these studies and requirement set, we analyzed recently proposed blockchain-based e-voting systems. As a result of these analyzes, one can determine that a mature blockchain based e-voting system that can meet all criteria has not been proposed yet. Particularly, we show that either the proposed schemes misses the basic requirements or does not fulfill these while claiming otherwise. Additionally, by simulating a large-scale election, we show that time complexity of e-voting schemes utilizing cryptocurrency blockchain such Bitcoin or Ethereum is impractical. Besides, we also emphasize new risks of utilizing public cryptocurrency blockchains for e-voting schemes. Accordingly, the readiness of blockchain-based e-voting has been discussed, from which it can be deduced that it would be more advantageous to research for e-voting specific blockchain technologies instead of utilizing existing cryptocurrency blockchains.

Keywords—Electronic voting, e-voting, blockchain, cryptocurrencies, security, privacy

1. Introduction

Traditional paper-based voting mechanisms are the most widely used election method which is developed over time, based on experience gained to provide certain properties. In the essence of a democratic electoral system, the choice of people should be based on free will. In order to do so, it is essential for an election system to ensure certain properties, such as anonymity, authentication, verifiability and robustness. The main advantage of a paper-based voting system is providing anonymity of the voters, along with their authentication. This is accomplished with the help of election booths, which is an efficient way of disabling all public and secret communication channels of the voter in both ways, [1]. With use of identical voting equipment, such as ballots and the markers, it is hard to determine voters' choice once its cast into the ballot box. Therefore, the choice of the voter

left to be completely their own free will.

Beside providing such crucial properties, paper-based voting systems has their own downsides. The cost efficiency and natural resource consumption are the first things that spring to mind. In a nation-wide election, these problems are burden on both nature and economy, which in the long run is seen as obstacles to a more direct democracy. Along with these problems, there are lack of functional properties, such as preserving integrity of the ballots, verification of the election result by public, auditing the transparency of the procedures. That is why so much effort on e-voting systems are widely proposed. Designs on electronic voting systems are already heavily researched subject of study for decades[2] and still goes underway.

Since e-voting schemes cannot simulate voting booths, a number of new problems comes into play, such as coercion and vote buying, that may undermine voter privacy or confidence in the election, [1]. An e-voting scheme both satisfy the current election systems features and requirements and also resist these new attack types such as vote buying and coercion.

There are e-voting system developed and used in some countries such as Brazil [3], Estonia [4], Switzerland [5], USA [6], Norway [7] and Australia [8]. Meanwhile, the adequacy of the security criteria claimed by these e-voting systems are discussed in the literature. Recently Switzerland held a competition where a flaw is found that allows an attacker to change cast votes, [9]. In parallel with these developments and studies, the design of e-voting systems on relatively newly developed blockchain technologies is one of the questions.

Especially, [10], [11] defines end to end verifiability for secure e-voting schemes. End to end verifiable election techniques enable individual voters to check crucial ingredients of election result without trusting the election software, hardware, election officials and procedures, [11].

After the first announcement of Bitcoin cryptocurrency in 2008 by Satoshi Nakamoto [12] , studies were started on different uses of cryptocurrencies and blockchain technology. Blockchain is a distributed ledger system that can store immutable blocks enabling single history of all peer to peer transactions. That is why it is widely believed that e-voting schemes can be one of the areas that benefit from blockchain. As summarized in the sequel, blockchain based e-voting schemes actively studied.

Related Work.

Zhao et al. [13] proposed the first e-voting scheme on a blockchain. The proposed system is an electral system with only two candidate where voters deposit extra BTC into the system to be refunded if they follow the protocol. In addition, because it is implemented with Bitcoin scripts, the protocol does not have flexibility property in changing the rules of the election. Ayed et al. [14] proposes a conceptual e-voting system where a blockchain is produced for each candidate in a single election, unlike other blockchain based e-voting approaches. However, we cannot analyze the system due to lack of detailed information. The e-voting system proposed by Hardwick et al. [15] claims to meet the criteria of fairness, eligibility, privacy and verifiability on smart contracts using permissioned blockchain infrastructure. However, it is stated that the electoral authority can be linked to the voter identity and the cast votes, [16]. The system proposed by Hjalmarsson et al. [17] obtains the result of the election with smart contracts. It is stated that it is still working and has not yet been tested for the alleged criteria, [18]. It is also noted that voters have access to partial results during the election period, [19]. In the system proposed by Khoury et al. [20], votes are sent publicly along with voter identification, [20]. In

the system proposed by Adiputra et al. [21], votes are encrypted with the common election key and the election results are obtained by issuing the key. As stated in the study, the election committee can establish a link between all voters and their votes.

Our Contributions.

In this study, here we first revisit the e-voting requirements defined in [22], [1], [23], [24], [25], [10] then we show that:

- [26] does not provide fairness and privacy requirements and does not consider forgiveness, coercion resistance, and receipt-freeness,
- [27] provides voter privacy without changing the Zcash protocol but cannot satisfy eligibility, uniqueness, robustness and end-to-end verifiability requirements,
- [28] has claimed robustness, voter privacy and coercion resistance when number of voters are high enough. Nevertheless, the system cannot provide robustness, privacy and coercion resistance, and does not consider forgiveness requirement. In addition, keeping actual records of the election in a system database creates a single point of failure.
- [29] does not provide the fairness and privacy and does not mention forgiveness, coercion resistance and receipt-freeness requirements.
- [30] It is stated that the system design causes a robustness issue where a single voter is able to cancel the election and coercion resistance and receipt-freeness requirements cannot be satisfied. Forgiveness isn't mentioned.
- [31] claims coercion resistance and receipt-freeness. Privacy on the system relies on generated common private key which should be destroyed after the election. Otherwise, cast votes is under risk of divulge. Besides, there isn't enough info to check recorded as cast and tallied as recorded verification.

We further discuss the applicability of Bitcoin and Ethereum based e-voting schemes for a nation-wide election and deduce that time complexity makes them impractical, even if the these schemes would satisfy the necessary security and e-voting requirements.

In our work, we are not discuss the problems inherited from blockchain infrastructure such as tolerance of dishonest node behaviours, transaction registration issues, cost of transaction fees, limitations of consensus rules. These issues are discussed in another study published by [32].

Organization. In Section 2, we are going to overview the requirements of an e-voting systems, then we are going to present the security flaws of the previously proposed blockchain based e-voting systems in Section 3. Next, we are going to discuss the limitations that a posed by the blockchain technologies in Section 4 and finally Section 5 concludes the paper.

2. Requirements of an E-voting System

In this section, the definitions of requirements that an election system should fulfill are given. Requirements are previously defined by [22], [1], [23], [24], [25], [10]

- *Eligibility*: Only authorized voters who are registered with their ID should be allowed to cast their vote, [22].
- *Uniqueness*: A voter should be able to cast ballot only one that will be counted in the final tally. It is important to notice that uniqueness does not mean unreusability, where voters should not vote more than once, [25].
- *Forgiveness*: The ability of a voter to alter their vote after it has been cast. This property links to coercion resistance, because it provides a coerced voter the option to change their own

cast vote at a later in order to reflect their true opinion, [25].

- *Robustness*: Dishonest participants should not be able to disrupt an election, [22].
- *Fairness*: Nothing must affect the voting, [22]. Partial results should not be obtainable before the end of the vote tallying to ensure that the remaining voters who have not voted yet would not be influenced by the early results.
- *Privacy*: The relation between voter identity and his/her vote should not be revealed to anyone. Privacy in an e-voting is defined in terms of an adversary that cannot interact with voters during the voting stage, [24]. Even if the administrator and the counter conspire, they cannot detect the relation between voter and her cast vote, [22]. Voter privacy must be preserved during the tallying as well as after the publishing the election results for a long time.
- *Coercion Resistance*: Coercion resistance is a strong form of privacy in which it is assumed that the adversary may interact with voters, [24]. The adversary may instruct an individual voter or a group of targeted voters to divulge their private keys subsequent to registration, or may specify that these voters cast ballots for a particular candidate. If the adversary can determine whether or not coerced voters behaved as instructed, then the adversary is capable of blackmail or otherwise exercising undue influence over the election process.

  A coercion-resistant voting system is one in which the voter can deceive the adversary into thinking that they have behaved as instructed, when the voter has in fact cast a ballot according to their own intentions, [24]. A coercer should not have ability to distinguish whether a coerced voter cast their ballot the way they were instructed to.

  Up to our knowledge coercion resistant voting systems assumes each voter has a honest moment in which can cast vote by her will truly. Therefore, in order to make one-voting scheme coercion resistant, allowing the cast vote more than once is a necessary requirement, [1].
- *Receipt-freeness*: Voters must neither be able to obtain nor construct a receipt which can prove the content of their vote to a third party both during the election and after the election ends, [1]. Receipt-freeness does not necessarily provide coercion resistance, [24].
- *End-to-End Verifiability [10]*: End to end verifiable election techniques enable individual voters to check crucial ingredients of election result without trusting the election software, hardware, election officials and procedures, [11]. The verifiability of an election can be divided into two parts according to its verifier.
  - *Individual Verifiability*: The voter should be able to verify its ballot is counted correctly in the final tally. Principal requirements of the individual verifiability in E2E verifiable election system is defined by [10] as follow:
    - *Presented Ballots are well-formed*: The ballot should be interpreted in the same way by the voter and the voting system. The representation of the voter's choice on ballot should be same with the representation that will be used by rest of the election system.
    - *Recorded as Cast*: Voter's choice should be recorded correctly by the election system.
    - *Voting system follows the protocol*: Whenever there is some part of the voting protocol which must be followed by the voting system in order to ensure the integrity of the election, there must be some verification which can detect and provide publicly acceptable irrefutable proof

when voting system doesn't follow the protocol.

- *Universal Verifiability*: Anyone should be able to verify election outcome is correctly tallied. Principal requirements of the universal verifiability in E2E verifiable election system is defined by [10] as follow:
  - *Cast Ballots are well-formed*: Voter's choice should correctly impact on the final tally. Thus, ballots cannot contain over-votes or negative-votes.
  - *Tallied as recorded*: Anyone should be able to verify that announced tally has been constructed from all the recorded ballots.
  - *Consistency*: The set of ballots in "Recorded as cast" should be the same as the set of ballots in "Tallied as recorded".
  - *Each recorded ballot is subject to the "recorded as cast" check*: Anyone should be able to detect if any cast ballot does not have a unique voter who is able to check "recorded as cast" verification.

When a fault is detected in a verification, availability of a proof of fault is highly desirable. All the requirements defined above should have publicly acceptable proofs. Only the 'recorded as cast' requirement does not require its proof to be irrefutable. The same set of requirements without existence of proofs called *weak version of the requirement*, [10].

3. Security Analysis of E-voting Schemes

Based on the requirements given in Section 2, now we are going to analyze the blockchain based e-voting systems.

3.1. E-voting System Based on the Bitcoin Protocol and Blind Signatures [26]

Cruz et al. [26] have proposed a system that relies on prepaid bitcoin cards (PBC) and blind signatures [2] over Bitcoin cryptocurrency [12] as a public bulletin board. The idea is breaking linkability between identities of voter and their ballots with usage of the PBC while preserving the transparency of the system with blind signatures and Bitcoin protocol.

In general, there are 3 main actors in the system that take actions over 3 stages. At the registration stage, Administrator publishes a list of valid candidates for the election in progress. In order to register, each voter should encrypt their choice with a randomly generated key k, such as $x_i = E(choice, k)$. In the next step, voters should make an application to administrator with their identity and blinded version of their encrypted vote x_i in face to face. Administrator is responsible to sign the message x_i, if the voter is eligible to vote. A PBC has been issued to each eligible voter by the administrator, along with the signature. While the signature y_i indicates that owner of the message is a legit voter, administrator does not gain any knowledge over the actual message x_i or the choice of the voters, thanks to property of blind signature schemes. The privacy of the voter relies on the anonymity of the PBC key to ensure that it cannot be traced back to voter. At the end of the registration stage, administrator publishes a list of eligible voters as L_{voters} who requested a signature with their ID and their blinded messages as $< ID, x_i' >$.

Each voter can transfer the coins from PBC to their own Bitcoin accounts, to ensure their own privacy and anonymity. To cast a ballot, voter should create a transaction from their own Bitcoin address to administrator's address with an OP_RETURN that contains the pair of signature and the unblinded

message $< x_i, y_i >$. This transaction indicates that voter committed his choice with the information of eligibility for anyone. Authors suggest that usage of PBCs would increase the security and reliability of the election, where A publishes all the public address of issued PBCs.

At the end of voting stage, the administrator can verify the signatures in the transactions and publishes a list of valid ballots as $L_{ballots} = \{< V_i.BA, x_i, y_i >, ...\}$ that will count in the tally stage. The number of the entries in both list L_{voters} and $L_{ballots}$ should be equal.

Each voter creates a transaction from their own address to counter's address with an OP_RETURN that contains the encryption key of their votes. Counter checks the list of valid ballots and verifies the entries on the blockchain, ensure that the ballot is valid. For each successful check, counter decrypts the message x_i using key k_i and adds 1 to counter of the respective choice. Finally, the counter announces the results.

- *Eligibility*: Assuming that signature scheme used by administrator is secure, nobody except the administrator cannot create a valid signature y_i. Since the eligibility depends on the signature y_i.
- *Uniqueness*: Uniqueness depends on $< x_i, y_i >$ pair. Since the $< ID_i, x_i' >$ list is published at the registration stage, a corrupt administrator cannot give multiple signatures to the same voter without being noticed. Therefore, a voter can cast only one valid ballot.
- *Forgiveness*: A voter registers his/her choice in the registration stage and it must be signed by the administrator. Due to the design of the system, if the administrator creates more than one signature for the same voter, there is a situation that contradicts the requirement of uniqueness. Therefore, it is not possible to change the preferred choice for the voter. Thus, forgiveness property is not provided.
- *Robustness*: A voter can send the key from another Bitcoin address, if the voter loses or compromises the private key of the given Bitcoin address before sending the k value to counter. This might introduce some flaws, such as someone else can send an incorrect key k' that would invalidate the vote for x_i. This can be safely assumed to be valid if there is a validation check to verify the key actually is the correct one. Because, only the eligible voter possesses the correct key k and x_i cannot be changed. This system provides robustness because no one can change the course of the election.
- *Fairness*: As mentioned in the published article [26], a voter can intentionally disrupt the fairness by sending k during the voting stage. To avoid the problem, authors suggest that counter's public address can be kept secret until voting stage ends. Nevertheless, voters have the possibility to publish their key values k to get a partial result before the voting stage ends. This would violates the fairness property. Hence, the property is not satisfied.
- *Privacy*: Voter privacy depends on blind signature scheme, the anonymity of both the PBC's public bitcoin address and the voter's public bitcoin address. Since the transactions on the Bitcoin blockchain can be traced back to its origin [33], one can easily link the cast vote and the voter if one of two addresses is compromised. As mentioned by authors, voter should use bitcoin address solely for the voting process. Besides this, voter should make sure that PBC's public address is not logged. Authors suggest that PBCs can be given inside of an envelope. The problem might still exists, unless the envelopes are randomly chosen or shuffled by voter. Despite the suggested mitigation, the

underlying problem still exists.

- *Coercion Resistance*: An adversary can instruct a voter to cast his vote with the key or simply give a x_i to register which both are generated by the attacker. Since the attacker can determine whether the given value is stored in the blockchain, the system does not provide coercion resistance.
- *Receipt-freeness*: The choice of the voter will be revealed to public at the counting stage in plain text. A voter can create a receipt with the private key of the PBC, the private key of the voter's Bitcoin address or possession of key k to create a receipt for proving the cast vote recorded on the blockchain. As it can be seen easily, receipt freeness property is not satisfied by the system.
- *Individual Verifiability*:
  - *Presented ballots are well-formed*: Presented ballots are published by administrator at the beginning of the election. Therefore, voter can verify that whether his ballot is well-formed or not by checking the published list. Even if authors explicitly states, it is a better approach for candidates to sign the values that represent them on the ballot.
  - *Recorded as cast*: Since the Bitcoin has a permissionless public blockchain infrastructure, voter can check whether his ballot and their keys are recorded correctly or not. Also, a voter can check the verified lists against the blockchain which are published publicly. Therefore, *Recorded as cast* check is provided by the scheme.
- *Universal Verifiability*:
  - *Cast ballots are well-formed*: The encrypted ballot on blockchain cannot be checked until their corresponding key values are revealed. Once the key values have been published, any one can decrypt the encrypted ballot to examine whether it contains a value from the published list of candidates. If any ballot contains invalid value, anyone will be able to detect it. In addition, there is no way to generate excessive or negative votes using values from the published list. Therefore, the election system ensures the correctness of the format of the votes cast.
  - *Tallied as recorded*: Since the Bitcoin has a permissionless public blockchain infrastructure, election data is publicly available. Anyone can count and verify that announced results are correct. Thus, *Tallied as recorded* check is provided by the scheme.
  - *Consistency*: Anyone can compare two public list. It is mentioned that a mismatch or overflowing of the lists can only happen because of a corrupt administrator which can be detected at the counting stage. Since this can be clearly detected at the counting stage, the system provides consistency check.
  - *Each recorded ballot is subject to "recorded as cast" check*: Because the identity of the voters and their encrypted ballots are published at end of the registration stage, administrator cannot create valid ballots instead of voters who did not vote.

3.2. Internet Voting Using Zcash [27]

This scheme utilizes Zcash [34] as underlying protocol without any change in its protocol. The protocol assumes that the confirmed identity of a voter can be verified to check legal rights to submit a vote. The voting protocol has four distinct stages that follows each other. These stages are registration, invitation, voting and tally.

Registration is the first step to create an eligible set of voters. It is required as part of identity

verification. Also, it is needed to keep track of which voters have cast a ballot. Registration is done via an online registration web page. After a successful registration, the organizer should have stored the voters email address. Invitation is the stage where system sends an one-time unique link to the voter's email address to redirect the voter to the unique ballot assigned to them. The link is used for voters to specify their Zcash t-addresses to receive ZEC token. This is the address used to make sure that voter receives the vote token which will be send to a candidate later. Each ZEC token given is counted by the system. Voter creates a z-address and transfers the vote token to his/her protected z-address. Then, the voter sends the token from his/her z-address to address of the candidate that he/she chooses.

There are two variants of the system which can be differentiated by receiving address type of candidates. If candidates use a z-address, transactions, the voter may not be validate their own vote for tally when candidates empties their wallet into ZEC pool. This variant requires more trust in the system while guarantees the privacy of the system. If candidates use a t-address, token balances of candidates can be observable by anyone in real time. The linkability of the tokens would also be preserved. This means voters can validate if their votes has been counted or not in the tally. Regardless of the variant used, the JoinSplit transfers of vote tokens from voters to candidates are stored on the blockchain.

The last step of the election is counting votes and auditing. It is done by candidates send all their acquired ZEC tokens to t-address of the ZEC pool using their t-addresses. End of the tally stage, total issued ZECs by the trusted ZEC pool must be equal to total votes from the sum of the candidate received votes.

- *Eligibility*: Voters are registered with their ID at the start of an election. If the voter is eligible to vote, voting system gives a token. On the other hand, since a vote token represented as exactly 1 ZEC, anyone that has more than 1 ZEC can send a transaction to a candidate which means casting a vote. Even the system checks the voter's public key has voted before, a ZEC can be send via a wallet. Since the transaction would be shielded, system cannot relate the origin t-address. This violates the eligibility property.
- *Uniqueness*: Uniqueness on voting depends on issued vote token. The same issue in eligibility can be applied. Even if administrators have made a transaction that can be clearly displayed to the voter's t-address from their own t-address, voters can cast ballot more than once by performing a transfer from their wallet, since ZEC can be supplied from outside of e-voting system.
- *Forgiveness*: Since Zcash prevent double spending, each honest voter has only one chance to send their issued vote token. There is no way provided in the scheme that allows to change the cast ballot. Thus, forgiveness property is not satisfied by the scheme.
- *Robustness*: In case of using z-addresses for candidates, it is stated that losing candidates may not send all of their received tokens. Since the system checks the integrity of the election by matching the total issued ZEC with sum of all votes, authors suggest that implement a counter that increases when the candidate received a new transaction. It cannot be achieved as publicly verifiable while using receiving z-addresses. Beside this, it is not possible to determine which candidate behaved honestly. Also, it should be mentioned that the token delivery depends on the client side scripts. A voter may request a token, but not send the candidate. This cannot be guaranteed with client side scripts that are running on an uncontrolled

environment.

- *Fairness*: If t-addresses are used for candidates' addresses, each candidate's vote count will be public. Therefore the system does not satisfy fairness property. In case of using z-address, only candidate's themselves have an information of their vote count. This is against fairness property. No one should be able to obtain full or partial results before tallying, including candidates.
- *Privacy*: Vote privacy depends on transaction linkability of Zcash join-split operations. Since Zcash join-splits cannot be related by public and the receiver of the transaction, vote privacy is satisfied.
- *Coercion Resistance*: Since the ownership of t-address used for the delivery of the tokens cannot be validated by the voting system, a coercer can instruct a voter to use a known t-address into ballot which belongs to coercer himself. Also, it is mentioned that a coercer could get access to the voter's email first and attempt to cast a vote on their behalf. Hence, the voting system is not coercion-resistant. Authors stated that coercion resistance is out of scope in their study.
- *Receipt-Freeness*: As stated in the ZCash documentation [35], users might want to give third-parties view access to their shielded addresses without also handing over spending capabilities for accounting or auditing purposes. Since the Zcash API provides a way to export viewing key, voter is able to create a receipt for their vote.
- *Individual Verifiability*:
  - *Presented ballots are well-formed*: Ballots are sent by voting system and should have been able to verified their integrity by voter. Since there is no way to validate candidates addresses on the ballots in the system, the system does not satisfy the property.
  - *Recorded as cast*: Voter can check whether his vote recorded correctly or not by validating the transaction which is occurred on the blockchain. Hence the voting system provides the 'recorded as cast' check.
- *Universal Verifiability*:
  - *Cast ballots are well-formed*: If t-addresses used for candidates, the amount received end is publicly visible. On the other hand, in case of using z-addresses for candidates, the the amounts in the transactions are not publicly visible. Sending a negative vote is prohibited by the Zcash protocol, but overvotes cannot be detected publicly in the system.
  - *Tallied as recorded*: If candidates are using t-addresses to receive vote tokens, voter can check transactions on blockchain to validate their vote tokens counted correctly. On the other hand, if candidates use z-addresses to receive tokens, voter cannot link their vote token at the tally stage. In both variation, nobody is able to verify that vote tokens are sent from legit voter.
  - *Consistency*: The system ensures that there are no extra votes with the system count and ZEC balance of candidates' wallet. If some of voters receive a vote token into their t-address, but they did not use it to cast a vote, then the counters cannot be used to detect extra votes. Also, it is stated that a possible attack is that if the losing candidate does not submit all of the received votes, integrity of the election might not be verified. Therefore, consistency property is not satisfied by the scheme.
  - *Each recorded ballot is subject to "recorded as cast" check*: Zcash protocol breaks linkability between ballots and the voters along

with the information that indicates eligibility. Anyone should be able to detect that the cast ballot has an unique corresponding voter to make sure that voting system cannot trick for a voter to check another voter's ballot. Hence, the property cannot be satisfied.

3.3. An E-voting System Based on Blockchain and Ring Signatures [28]

The proposed system is an e-voting protocol based on Bitcoin [12] with usage of the ring signature algorithm, [36]. There are three distinct roles in the system: Voters (V_i), Registration Authority (RA), Election Authority (EA). The protocol assumes that hashing algorithm $sha256$ is secure, RA and EA will not correspond and every actor follow the phases to enroll the voting process. The system consist of three phases.

The preparation and registration phase includes procedures where the authenticated voters and candidates registers into election system while EA collects their public keys to generate the key of the ring signature. Registration of voters and candidates are done in person. To register for casting a ballot in the election, a voter should generated a key pair for the ring signature, and submit the public key via random registration links that are generated and sent from RA via e-mail. The private key should be kept in secret. At the end of voter registration phase, the set of voters should be a fixed number of n. Once the stage is over, RA cannot accept any new registrations due to nature of ring signature.

Before the voting stage, EA generates a number of k BTC in their blockchain account to pay the transaction fees to voters. Transaction fees are delivered to voters by handing over the private key of the BTC addresses that are generated by the EA at the end of registration phase. To create a valid ballot, the voter requests the set of public keys from RA. Using the public key set and their own private key, voter signs their preferred candidate ID as σ. The voting system saves the pair $(\sigma, sha256(\sigma))$ at the same time. Then, the voter picks a bitcoin address from the address pool and request its private key from the EA if the link has not been used to retrieve an address from the pool. The voter casts their ballot by creating a transaction from the retrieved address to EA's bitcoin address with an OP_RETURN consists of the commitment c_i.

$$c_i = commitment(sha256(\sigma(C_i, SK_i, (PK_1, ..., PK_n))), C_i, L_i)$$

After the voting stage has been ended, the system returns the set of $(\sigma, sha256(\sigma))$ pairs and the set of public keys PK. All of the OP_RETURN data from transactions in EA Bitcoin address are collected and decoded to σ and C_i. The system compares the signature values and checks their verification. On each correct validation, the system adds 1 to the respective candidate C_i. If the transaction history for the same address is used more than twice, system should counts the first and ignores others.

- *Eligibility*: Eligibility property depends on the creation of the ring signature in the system, which must use a fixed set of public keys. Since the public keys are collected at the registration stage, an eligible voter should be able to vote if their public keys present in the set. Therefore, the scheme satisfies the eligibility property.
- *Uniqueness*: A voter can create multiple ring signatures with the same private key of theirs over different candidates. After that, they should create a transaction that pays to EA with an OP_RETURN that includes their commitment of choice and signature. In the protocol, it is specified that EA checks the voter has requested an address before or not. It is mentioned that the

protocol has a method to ignore extra votes from the same voter if they use the same pool address A_i. Therefore, a voter cannot use another valid vote. The system satisfies the uniqueness property.

- *Forgiveness*: It is stated that the system verifies the validity of the ring signature and only count the first and legal vote. Therefore, a voter cannot change their choice after the first legal vote is cast. The system does not provide the forgiveness property.
- *Robustness*: Due to bitcoin data storage limitations does not match the size of the ring signatures, authors suggested that only the hash values of the signatures stored in the bitcoin blockchain. The actual signature values that are created by voters are stored in the system database. If the system database is compromised with a data loss, the signature values must be submitted again by their owners. Otherwise, the election result might not be verified. Therefore, robustness property is not satisfied by the protocol.
- *Fairness*: According to authors, because of the tallying system is in the real time, this property cannot be guaranteed.
- *Privacy*: It is mentioned in the implementation section that commitments in the OP_RETURN fields can be decoded by anyone in the tally stage. Since the voter requests the private key of an address in the address pool, EA can relate the voter V_i and their cast ballot in the Bitcoin address A_i if the link LK_i is not randomized which cannot be verified to be randomized in the system. The protocol does not provide the vote privacy requirement.
- *Coercion Resistance*: Since the signatures on the blockchain cannot be identified by signer, a voter can deceive the coercer about how they cast their ballot. It can be ensured only the number of the voters is high enough, [28]. On the other hand, because the voter creates their own key pairs after the registration phase, a coercer can force the voter to use a given key pair which will be included in the public key of the ring signature. If the coercer knows the key pair, they can brute force to find the signature that would match the keys from all signatures published in the tally phase. Since a valid cast ballot cannot be changed by voter, the system is not coercion resistant.
- *Receipt-freeness*: It was stated that the voter obtains a transaction ID after creating a ring signature during the voting phase. In the event that the voter wants to prove their ballot, it cannot be understood whether the elector is acting honestly as he can say any transaction ID from the blockchain. However, due to the plain text recording of votes, a receipt can be generated by proving that the private key of the Bitcoin address for the referral transaction is known. For this reason, the system does not provide receipt feature.
- *Individual Verifiability*:
  - *Presented ballots are well-formed*: Although candidate IDs are given to voter through the public API, the protocol does not provide a method that verifies to the given data preserves its integrity. This can be easily implemented as an addition, such as candidates signature over their IDs are provided by the public API or recording their IDs into blockchain transaction before the voting stage begins. Since such a mechanism is missing in the protocol, the check cannot be verified by voter.
  - *Recorded as cast*: Hash value of the ring signature $H(\sigma)$ and the vote value C_i is recorded on the blockchain and a pair $(\sigma, H(\sigma))$ over C_i saved into system

database. A voter can verify that their vote recorded on the chain or not. Beside this, the ring signature values is recorded into system database. A voter should be able to check that their signature is recorded correctly into system database, since their ballot cannot be verified in the tally, if their signature is missing in the system. Since the system provides a public API of matching the ring signature through the hash value, the protocol is satisfies the recorded as cast check.

- *Universal Verifiability*:
  - *Cast ballots are well-formed*: There is no possible way to create a vote that contains negative vote or over-vote in the protocol. Besides the ballots are cast in plain text form, anyone can check whether the ballots are well-formed or not.
  - *Tallied as recorded*: A voter can verify whether their ballot has been counted by calculating the election result, since the recorded hash of the voting value is recorded in the blockchain and the signatures are publicly provided. Since the voting preference is also written to the blockchain with the hash of the signature, the choice cannot be changed even if the ring signature is signed with a known private key. However, in the event that the system database is compromised, if the voter's signature is deleted or changed, the voter's vote may be canceled as it will not match the values in the blockchain. In such a case, the invalidity of a voter's vote by any one may be determined with the transaction of his/her commitment to the blockchain.
  - *Consistency*: In case of compromised system database that has been mentioned tallied as recorded check, anyone can compare the list of pairs recorded in the database with the ones on the blockchain. Since blockchain transactions cannot be changed, there should be a difference that suggests one of the list are compromised. Therefore, detection of consistency check is satisfied.
  - *Each recorded ballot is subject to "recorded as cast" check*: To create a valid ballot, one needs to create a transaction on the blockchain and save their signature into system database. The election authority can create a transaction instead of voters who did not use their vote, if one of the private keys of the ring signature is compromised. In such a case, nobody, including the voter who did not vote, cannot realize that there has been a ballot cast in their behalf. Also, the statement is true for if anyone injects their own public key into compromised database in the registration procedure. Hence the protocol does not satisfy Each recorded ballot is subject to "recorded as cast" check.

3.4. An E-Voting Protocol Based on Blockchain [29]

The proposed system aims to provide anonymity with blind signatures [2] on custom blockchain [12]. The protocol distributes the responsibility of the a trusted third party (TTP) over multiple authorized entities. There are three types of participant who play roles on the protocol. These are voters, organizers and inspectors. Each participant should have a pair of asymmetric keys for addresses on blockchain and signing transactions. Organizers and inspectors have a pair of asymmetric keys for signing. Each voter has an extra pair of asymmetric keys for anonymously casting their ballot and a pair of function to use as masking and unmasking operation

on their ballot.

Voter creates two pair of asymmetric keys, $(pk_{voter_i}, sk_{voter_i})$ and (pk'_i, sk'_i) to create transactions on blockchain. After that, the voter submits his/her ID and a public key pk_{voter_i} while keeping sk_{voter_i}, pk'_i, sk'_i in secret. Organizer adds information of the voter along with his/her public key into a list, if voter is eligible to vote. After the registration stage is over, organizer publishes a list of eligible voter. In order to cast a ballot, the voter creates a vote string V and computes its hash value $hash(V)$. The voter applies the blinding function to hash value $c' = blind(hash(V))$ and creates a transaction from pk_{voter_i} to $pk_{organizer}$ with the information of c'. Organizer signs the message c', if the public key of the sender is in the eligible voter list, signature of the sender can be verified and they have not voted yet. Otherwise, organizer ignores the message. Organizer creates a transaction from $pk_{organizer}$ to pk_{voter_i} for sending his signature over c'. Voter repeats same steps for each inspector to get their signature. Inspectors follows the same procedure with the organizer.

After the required signatures is collected, the voter removes the blinding factor over the signatures that is sent by organizer and inspectors. Voter creates a transaction from pk' to $pk_{organizer}$ to send the ballot which contains V, $signature_{organizer}(hash(V))$, $signature_{inspector_j}(hash(V))$ as casting a vote.

In the tally stage, the organizer collects all ballots from blockchain and check their validity. The ballot is added into the valid ballot set, if the ballot is well-formed. After the tally, organizer publishes the results with valid ballot set as election result.

- *Eligibility*: Eligibility of the voter relies on the signatures of the organizer and inspectors. It is stated that corruption may happen if organizer and inspectors conspire together. To avoid dishonest behaviors, authors suggest that the number of organizer and inspectors should be increased. The disadvantage of the mitigation is increased number of transactions required to cast a vote per voter. If any of the organizer or observers rejects sign the message, it can be seen on the blockchain.
- *Uniqueness*: Uniqueness is provided with the components of the ballot, random bit string and the choice value. It is stated that only one of the ballots is valid from the set of ballots with the same random bit string. Therefore, creating a valid ballot requires change of the random bit string. That would break the signatures of the organizer and inspectors. Since they will ignore the messages from the same voter, second valid ballot should not be exists in the protocol.
- *Forgiveness*: Voters cannot create a second ballot with the required signatures. Since there is not any other way provided to change the cast ballot values by the protocol, forgiveness property is not satisfied.
- *Robustness*: Since any of the organizer or inspectors refuses the sign a vote, it can be easily observed on the blockchain. In addition, voters cannot take any action that causes disruption of the election.
- *Fairness*: As it can be seen easily, the election results are public on the blockchain to anyone in voting stage without any extension. Authors mentioned the problem and suggests two possible solution. The first one is the usage of permissioned blockchain, which leads to loss of the transparency of the blockchain as they have indicated. The other suggestion is usage of the public key encryption where the public key is provided by the organizer. The corresponding private key is kept in secret before the tally stage. However, the key holder gains authority not to publish the private key to decrypt the result which leads to a robustness issue. Besides,

organizer still can access intermediary results which is against the fairness property. Hence, the fairness property is not satisfied by default.

- *Privacy*: Vote privacy relies on hash function and the blinding operation and the pseudo addresses on blockchain network. Authors mentioned that, blockchain network may reveal the IP address of voters via network analysis which makes possible to create a link between voters and their ballots. They suggests usage of anonymity services to voters. Since the vote is sent to blockchain in plain text, the protocol does not satisfy the vote privacy by its own.
- *Coercion Resistance*: An attacker can make a voter to use the key which is given by himself/herself. Since, the actions of the voter can be observed on the blockchain by anyone, a voter cannot trick the attacker. Also, in order to make one e-voting system coercion resistant, allowing the cast vote more than once is a necessary requirement, [1]. Hence the protocol does not satisfied the forgiveness property, coercion resistance cannot be satisfied.
- *Receipt-freeness*: The voter is able to create a receipt by proving the knowledge of the private key of the address is used to send the vote string. Since the vote string is in plain text and the blockchain data is available to public, receipt-freeness requirement cannot be satisfied.
- *Individual Verifiability*:
  - *Presented ballots are well-formed*: The protocol does not mention how the representation of the candidates are published.
  - *Recorded as cast*: Since voters can check transactions created by themselves to verify that their ballot is correctly recorded on the blockchain.
- *Universal Verifiability*:
  - *Cast ballots are well-formed*: Ballots are published plain text form, as it can be seen easily whether they are well-formed or not.
  - *Tallied as recorded*: Ballots are collected from the blockchain for tally. Since the content of transactions on the blockchain cannot be changed, it can be checked whether ballots are tallied as recorded or not by anyone.
  - *Consistency*: Since ballots are recorded and collected from the blockchain, anyone can compare two sets by calculating the results and compares with the announced result.
  - *Each recorded ballot is subject to "recorded as cast" check*: Each voter creates an unique ballot for casting their vote. Since the signatures of the organizer and inspectors can be checked by voter and their ballot is recorded on a public blockchain, the authority cannot trick the voters by giving the same signatures.

3.5. A Smart Contract for Boardroom Voting With Maximum Voter Privacy [30]

The Open Vote Network is a distributed two-step blockchain based e-voting protocol designed by McCory et al. [30] for use in small scale, low-coercion elections with maximum voter privacy. There is no need for a counting authority because the protocol has self-tallying property. Anyone who has voted in the election can conduct the counting process without violating other's privacy. Designed as a smart contract in the Ethereum blockchain. Therefore, voters can rely on the agreement on the blockchain that the protocol is executed correctly.

It is stated that the commonly used blockchain implementations cannot be used in a national election due to the limitations of data storage and transaction operations, so the protocol presented is designed for small-scale elections such as board

meetings. It is assumed that all voters have a secure communication channel. In addition, it is considered to be only yes/no election, but the protocol can be extended as multiple options, [37].

The election administrator publishes the eligible voter list. The administrator adds the account information of each eligible voter to the whitelist included in the voting contract. Voters in the published list are required to register their voting keys at the registration stage to agree on a G value. Each voter entitled to vote sets a random x_i secret voting key. Each voter publishes a zero knowledge proof $ZKP(x_i)$ of g^{x_i} that the value x_i is known. At the end of the key registration phase, the key values generated, after each voter has verified the validity of the proofs published by the other voters.

$$g^{y_i} = Y_i = \prod_{j=1}^{i-1} g^{x_j} / \prod_{j=i+1}^{n} g^{x_j}$$

In order to cast their ballot, each voter publishes their choice of v_i as $g^{x_i y_i} g^{v_i}$ with the zero knowledge proof that $v_i \in 0, 1)$.

All published zero knowledge proofs are verified to check that the votes are in the correct format. At this stage, all voters must have cast their votes. All registered voters can calculate the tally as $\prod_i g^{x_i y_i} g^{v_i}$ and find the result as $g^{\sum_i v_i}$.

Since the number of voters is considered to be low, v_i would be a relatively small value, so it can be found with a brute search.

- *Eligibility*: Due to the inclusion of the voter list as a whitelist in a contract controlled by the administrator, a person who does not have the right to vote cannot vote in the election because the smart contract will ignore this transaction. For voters who have the right to vote, the procedure is clearly stated. Apart from this, it is mentioned that the smart contract should determine the identity of the voter with 'msg.sender' instead of 'tx.origin'. This avoids the use of another contract to emulate a voter.
- *Uniqueness*: When a vote is requested from another address, the contract will be ignored as it cannot be verified the address from the whitelist. A voter can vote at most one valid vote, as the voting value used by the voter is identified by 'msg.sender' on the smart contract and only one vote key can be generated during the key registration stage.
- *Forgiveness*: Forgiveness property is not mentioned in the system.
- *Robustness*: As a common problem of self-tally e-voting systems, one of the problems is mentioned as abortive issue. As stated, the voter who uses the last vote can calculate the result before everyone else. In such a case, if a dissatisfied voter does not complete the voting process, no one will be able to calculate the election result. As noted, the election result can be recalculated with full co-operation of all the remaining voters by adding an additional recovery step, [38], [39]. However, the same situation is encountered in this mitigation. The authors proposed a deposit/refund solution that would create an incentive for voter participation through smart contracting. However, the mitigation does not solve the problem, when the voter will benefit more if the election is canceled.
- *Fairness*: The last voter who cast their vote can calculate the result by simulating the system with the values stored in the blockchain as if they had used the game before everyone else. An optional additional step has been proposed to avoid influencing the voter's decision. In this step, all voters commits the hash value of their vote before submission. In this case, even if the last voter can calculate the result in advance,

the voting process is not affected as the voter cannot change their vote.

- *Privacy*: The privacy of a voter is based on the confidentiality of their ballot. Since ballot secrecy is based on discrete logarithm problem, it is not applicable to calculate the vote value. As stated in the proposal, a voter's ballot can only be found via full collision of all of other voters. Therefore, the system meets the privacy requirement.
- *Coercion resistance*: It is stated in the proposal, the protocol cannot provide coercion resistance for an election in an unattended environment. Even if a voter who had to use their vote under pressure can change their vote, the voter always can prove his/her vote by revealing the vote key x_i, so that the coercion resistance cannot be achieved.
- *Receipt-freeness*: It is stated in the proposal, a voter can create a receipt with the values stored explicitly on the blockchain by revealing the vote key x_i.
- *Individual Verifiability*:
  - *Presented ballots are well-formed*: The ballot format is clearly stated as 1 as yes and 0 as no.
  - *Recorded as cast*: Since the encrypted vote values are written into the blockchain, a voter can verify whether their ballot is recorded correctly or not by following the blockchain.
- *Universal Verifiability*:
  - *Cast ballots are well-formed*: The correctness of the cast votes is based on the zero knowledge proof submitted with the vote. A voter attempting to cast excessive or negative votes will not be able to produce the proof. Therefore, whether the cast ballots are well-formed or not can be check by everyone.
  - *Tallied as recorded*: Since the proposed system is a self-tallied election system, any one can calculate the election result using the values on the blockchain. The calculated result can be compared with the announced result.
  - *Consistency*: This can be checked by anyone because the election data is stored on a public blockchain and the result cannot be calculated if one of the cast votes is missing.
  - *Each recorded ballot is subject to "recorded as cast" check*: All votes used in encrypted form must be cast by a voter in the list of voters published by the administrator. Otherwise, the election result cannot be calculated and the list and the votes counted can be compared to determine the votes that have not been cast by a voter.

3.6. Polys Online Voting

Polys Online Voting [31] is a commercially available web-based e-voting service on Ethereum blockchain which breaks the link between voter ID and their votes with public ElGamal public key encryption, while ensuring the validity of the vote format with zero knowledge proof. It distributes the private key over trusted election representatives.

The most critical role in the preparation phase is the trusted representatives. They are responsible for signing the blocks in the blockchain, generating the public key that voters use to encrypt their votes, and decrypting the election result in the tally stage. Trusted delegates generate the public key pair among themselves using the Shamir secret share, [40]. Since each trusted agent has a part of the secret value of the generated private key, a greater number of trusted agents must be brought their part of secret together to reconstruct the key and obtain

the election result.

It is mentioned that voter authentication may vary for each election. The system assumes that the authentication is provided by a third party. There are three types of registration methods. The first is by specifying the voter's e-mail addresses when creating the election. The system sends a link address to their e-mail address where they can cast their vote. Secondly, one unique code is generated for each voter by specifying the number of voters when creating the election. The generated codes are distributed to the voters and allowed to vote. The last method is provided for cases where voters are not known in advance, such as voting during a conference. At the election-specific link, voters may vote for the relevant election. It is stated that the registration method for open voting is presented experimentally and it is not suitable for large scale elections.

The most critical part of the voting process is the anonymity of voters who are eligible to vote, while ensuring that all the ballots are well-formed. When an eligible voter logs on to the system, the client application creates a key pair to create transactions on blockchain. The voter sends his/her preferred candidate ID Z_i to be recorded onto the blockchain by signing with $ZKP(\mathcal{V} \in Z_f)$ to prove that the ballot is encrypted with the common election key pk_{common} and is well-formed. Any trusted representative verifies the voter's signature upon receipt of the ballot. If the signature is not valid, the ballot is registered as invalid. The ElGamal algorithm, which is used by the voters to encrypt their preferences, has multiplicity homomorphic feature. The valid votes are multiplied after the voting process is completed and the result of the election is encrypted with the public election key pk_{common} generated by the representatives. Trusted representatives can decrypt the results by sequentially applying the parts of the common election secret key s they have. The result is divided into prime multipliers by the big step/small step algorithm and the votes of the candidates are obtained.

- *Eligibility*: Each voter receives an unique email is sent by the e-voting system. The unique token value generated by the KECCAK-256 hash algorithm for each voter in the system is added to the voting smart contract. If the voter is eligible to vote, it may be cast by a newly produced intermediary contract, or even with the one previously produced. If there is no token of the voter from the contract, the votes cast by the smart contract are ignored. Therefore, the system provides the eligibility requirement.
- *Uniqueness*: There is a special intermediary smart contract, which is named as an alias in the election created for each voter. The voting status of a voter can be monitored by this contract. Because the outputs of the smart contract are written on the blockchain, a link can be established to cancel the previously cast ballot when the voter wants to change their choice.
- *Forgiveness*: As stated in the uniqueness requirement, voters are allowed to change their cast ballot at any time during the voting stage.
- *Robustness*: The public key was generated by using the threshold scheme. In addition, assuming that the zero knowledge proof algorithm has the soundness property, it is not computationally feasible to create a proof that a voter can mislead the verifier. One of the situations that may prevent obtaining the election result is that the calculated result cannot be divided into prime factors, when the calculated result is too large. The authors stated that this situation can be calculated by bringing together smaller partial results. However, the risk increases as representatives have to reuse their keys for each

partial result account.

- *Fairness*: Assuming trusted representatives are honest, partial election results cannot be calculated by anyone. Because the public key of the ElGamal algorithm is not known during the voting period.
- *Privacy*: Voter privacy is protected by the secrecy of the corresponding private key of the common encryption key generated by trusted representatives. Therefore, the private key must be kept secret even after election is over. This puts privacy at risk in the long run. Because the disclosure of the private key can be used to decrypt all of ballots that are cast in the election.
- *Coercion Resistance*: A voter is allowed to cast more than one vote in order to prevent coercion resistance and vote trading. However, the provision of a receipt-freeness does not necessarily mean that coercion resistance is also ensured, [24]. An attacker could coerce a targeted voter by trying to obtain login information to the system. In addition, a voter who wishes to sell their vote may trade by selling login information or token sent to him/her by e-mail. Proposed mitigation cannot prevent these situations.
- *Receipt-freeness*: A voter can prove that he has sent the vote by proving that the private key of the signature on vote transaction is known. However, since the value of the vote will be encrypted, the voter must prove their vote. In the ElGamal algorithm, the voter can prove the exact voting value as he/she can generate the registered vote in the blockchain using the public key together with the ephemeral key which is generated by the voter.
- *Individual Verifiability*:
  - *Presented ballots are well-formed*: The presented ballot format is not mentioned in the system.
  - *Recorded as cast*: Since the voter's choice is recorded into the blockchain with smart contracts, each voter can follow the record of the vote or the output of the intermediary contract which he has signed with the public signature key.
- *Universal Verifiability*:
  - *Cast ballots are well-formed*: The format of the votes that cast is based on zero knowledge proof. Any one can verify the correctness of the ballot format by verifying the zero knowledge proof with the ciphertext published in the blockchain.
  - *Tallied as recorded*: As the result of smart contracts in the Ethereum blockchain is recorded to ledger by all of the miner nodes, it is ensured that each voter's vote cannot be changed as soon as it is recorded.
  - *Consistency*: Anyone can verify the format and signatures of the recorded votes to obtain the ciphertext as a result of the tally. Consistency check can be achieved if trusted representatives publish proof that the plaintext election results are decrypted using this ciphertext and encrypted with the public key.
  - *Each recorded ballot is subject to "recorded as cast" check*: If same unique code is given to different voters, it will not be detected by anyone as the system allows voters to change their voting addresses.

4. Discussion

In this section, we are going to discuss maturity of blockchain-based e-voting schemes and their usage in a nation-wide elections. In order to do that, we first summarize our requirement analysis given in Section 3 in the Table 1. Even if OVN, [30] scheme is designed to be used in small-scale elections, such as boardroom elections we still include in

the summary since this scheme seems more solid than the other schemes. As it can be deduced from the table, one can unfortunately conclude that these blockchain-based e-voting system proposals do not fully meet the e-voting requirements.

4.1. The readiness of cryptocurrency blockchains for e-voting

Blockchain technology was originally designed as a financial structure. The criteria that cryptocurrencies such as Bitcoin and Ethereum designed to provide differ from the requirements deemed necessary for e-voting systems. For instance, while cryptocurrencies are designed to prevent double-spending, a person who has right to vote in an election may allowed to be able to cast more than a ballot as long as only one of them is counted in the tally.

In order to ensure voter eligibility in national elections, features such as citizenship and age of the person should be determined by authentication. Verification of voter's identity is the first step in privacy in an e-voting system. With a digital identity infrastructure based on zero knowledge proofs, it will be very valuable for e-voting systems to prove that a voter has the right to vote without being identified. However, digital identity infrastructures that do not meet the necessary privacy criteria, and hence may stir up new problems or require extra processes. One of the challenges that can be encountered is to ensure that voter eligibility and uniqueness co-exist, without revealing voter privacy. Although it is assumed that the digital authentication infrastructure is already present in proposed systems, these systems are not yet suitable for practical application since they are still under development.

In most of the e-voting system proposals, registration is performed by an authority that is assumed to be honest. This may also create the risk that fraudulent identities may be created by the authority and that non-existent voters can be included in elections, this may counterfeit the results of the elections. In order for an e-voting system to be fully distributed, voter verification against fraudulent identities created by a dishonest authority at the registration stage is another open issue that needs to be explored for large-scale elections where voters do not know each other.

In an e-voting system, where the costs are intended to be reduced, use of existing cryptocurrency networks that are considered safe bring the transaction fee problem. Although transaction fees for blockchain structures such as Bitcoin and Ethereum are very reasonable and understandable when considered as a financial structure, the requirement that voters have a small amount of money to vote in an e-voting system is contrary to the Universal Declaration of Human Rights. If the transaction fees are provided by the authority, all transfer transactions made in blockchains such as Bitcoin and Ethereum always maintain their relations with their history, [33], [41]. Blockchain-based systems in which votes are sent in plain text or decrypted during the tally, voter privacy might not be protected due to the behavior of voters in before or after the election.

Besides, the fact that Bitcoin's value went above $19,000 in December 2017 and that the value of this work was around $10,000 at the time of the study, has a significant effect on the cost of transaction fees[2]. In a public permissionless blockchain network, which is managed by independent nodes, the miner nodes prefer transfers with higher transaction fee due to constraints accepted by the blockchain network. Even if the transactions initiation with the minimum transaction fee is technically feasible, the miner nodes would naturally prioritize transfers

2. see https://coinmarketcap.com/currencies/bitcoin/

TABLE 1
The summary of the requirement analysis of blockchain based e-voting systems

Eligibility	Uniqueness	Forgiveness	Robustness	Fairness	Privacy	Coercion Resistance	Receipt-freeness	Presented ballots are well formed	Recorded as cast	Cast ballots are well formed	Tallied as recorded	Consistency	ERB is subject to RaC[1] check	Scalability	
Cruz et al. [26]	✓	✓	·	✓	–	✗	·	·	✓	✓	✓	✓	✓	✓	✗
Tarasov et al. [27]	✗	✗	·	–	·	✓	·	·	·	✓	✗	✗	✗	✗	✓
Wu et al. [28]	✓	✓	·	✗	○	✗	✗	○	·	✓	✓	✓	✓	✗	✗
Liu et al. [29]	✓	✓	·	✓	–	–	·	·	·	✓	✓	✓	✓	✓	✓
McCorry et al. [30]	✓	✓	·	–	✓	✓	○	○	✓	✓	✓	✓	✓	✓	✗
Polys Voting [31]	✓	✓	✓	✓	✓	✓[1]	✗	✗	·	?	✓	?	✓	✓	✗

✓ Claimed and satisfied.
✗ Claimed, not satisfied.
○ Mentioned without mitigation.
– Mentioned with infeasible mitigation.
· Not mentioned, not satisfied.
? Not enough info.
[1] Each recorded ballot is subject to "recorded as cast" check.

with higher transaction fees. This would delay the finalization time of the voting transactions. Furthermore, also one can argue that spending some of the election budget for these transaction fees contradicts with election cost reduction goals.

Currently, public permissionless blockchains suffers from scalability problem. The maximum number of transactions that can be processed is limited due to restrictions such as the frequency of block generation and the size of transfers recorded in the block. The required time of generating a block in Bitcoin is by design fixed to approximately 10 minutes so that the generated block can propagate between the nodes. In addition, again by design each block is limited to a maximum of 1 megabyte size, hence both time and space constraints limits the transaction per time performance. In a national-wide election, the measures to be taken against centralization should be taken into account within the scope of the system designed with blockchain, in order to ensure that the voter's ballot is recorded securely. For example, to verify a transaction that is securely recorded into blockchain, it is recommended to wait for at least 2 to 6 validations in the Bitcoin cryptocurrency. This amount is expected to be higher for an e-voting system. However, the disadvantage of the measure is that a vote registration process is extended as a period of time and its usability for voters is negatively affected.

Ethereum supports smart contracts which can be run with gas. The amount of gas required to process smart contracts in the Ethereum blockchain must be

supplied as a transaction fee. Since cryptographic functions require more computational calculation than regular operations, it would increase the gas cost considerably when used in a smart contract. In addition to the increase in cost, it should be also noted that some operations such as homomorphic or zero knowledge proof calculations cannot be performed due to the maximum gas limit in both a transfer operation and a block on smart contracts of the Ethereum blockchain, [30].

In blockchain technology, the immutability of the transactions allows reliable and secure value transfers between peers. This blockchain ledger generally behaves as a public bulletin board in context of e-voting systems to verify that votes have not been changed once they cast. However, this assumption is valid when the criteria of the consensus mechanism that is used in the blockchain are fulfilled. The tolerance of dishonest node can vary according to the consensus mechanism used. For example, if more than half of the computational power is honest in the proof of work consensus based on the processing power, the transactions occurred in the blockchain cannot be removed. The proof of work consensus which is based on the processing power assumes that it is secure as long as more than half of the computational power is honest in the blockchain network. It is assumed that the problem of centralization of computational power, also known as 51% attack, will not occur in the systems studied in general. However, according to a study, only 2% of Bitcoin miner nodes state that they constitute 3/4 of the total computational power, [42]. In blockchain based e-voting systems, it is technically possible to manipulate the election by entities that hold the majority of the resource that make the consensus secure. A centralized consensus mechanism for an e-voting system means that a recorded vote can be deleted from the ledger without being included in the tally.

User devices are considered to be secure in blockchain-based e-voting system proposals. However, the security of the devices used by voters who are intended to vote in a national election is another problem that remains open. In addition, the combination of paper-based election with blockchain-based e-voting system for those who do not prefer e-voting methods among the voters or who do not have the possibility is one of the generally ignored issues.

4.2. Applicability of the current proposals in nation-wide elections

In this section, the national-wide blockchain based e-voting scheme applicability will be examined by calculating the approximate cost and time based on this number. Recent national election in Turkey involved approximately 57 million voters, [43].

Bitcoin based E-voting Systems

The size of a Bitcoin transaction may vary depending on the number of inputs and outputs included. A simple transaction consisting of one input and two outputs contains approximately 250 bytes of data. Since the maximum block size is limited to 1 megabyte in size, a maximum of 4,000 transfers can be added to a block. Due to difficulty of the proof of work consensus, the required time for forming a block is approximately 10 minutes. Even in the best case where the transaction of voters is given higher priority by miners, a maximum of 24,000 transfers per hour can be recorded.

In order to process all of the transactions created by voters, approximately 14250 blocks per round should be created. Assuming a block is created approximately every 10 minutes, it will take a

minimum of 99 days to create such blocks. It should be noted that as the number of transactions required to complete voting procedure increases, this number increases cumulatively. For instance, in a system that requires one transaction for registration and one transaction for casting ballot, it would took approximately 198 days to record election data into blockchain. It should be noted that this is the best case that transactions do not contain any OPRETURN data and miners in the network prioritize e-voting transactions.

The block number of 1st confirmation	Estimated Time	Estimated Transaction Fee	Estimated Transaction Fee(USD)
1st block	10 mins	33 satoshi/byte	0.85$
3rd block	30 mins	32 satoshi/byte	0.82$
6th block	1 hour	9 satoshi/byte	0.25$

TABLE 2
Block time and approximate transactoin fees for first confirmation[44].

In Bitcoin, transaction fees are determined by the sender of the transaction. The transaction fee amount only affects how quickly the initial confirmation is obtained. This amount may vary according to the daily processing density. Transaction fees stated in the Table 2 are given on the basis of August, 2019.

According to Table 2, to obtain the first confirmation in 10 minutes for a transaction 250 byte long, at least 8250 satoshi (33 satoshi/byte) transaction fee should be paid. In addition, approximately 50 minutes are required to obtain 6th validation for a transaction that has been added to the block. A total cost of election with 57 million voters, approximately 4702.5 BTC ($ 47 million) must be allocated to receive sufficient verification for each voter in the blockchain within 50 minutes. This amount increases cumulatively with the number of transactions a voter has to make for voting. In addition, since the transfer fee will be sent to the miners, this amount has to be re-established for each election. This cost has been calculated approximately when 1 transaction for voting is performed.

Ethereum based E-voting Systems

Bitcoin has limited ability to store data, even though it provides the non-alterability, which is expected to provide an important feature for the public bulletin board in an e-voting system. Furthermore, Bitcoin's scripting language is not Turing complete, so there is limited scope for processing in transactions. More specifically, the halting problem in Bitcoin scripting language is mitigated with limited scripting size and preventing the branching capabilities. Although the analysis of this study assumes that the e-voting system is implemented honestly, the operation of the protocol should be verifiable. In the Ethereum blockchain, unlike Bitcoin, it is able to store data in smart contracts, so that it can be validated across the network without the need for third parties.

In Ethereum-based blockchains, each smart contract requires a certain amount of gas as payment to ensure that the running contract is terminated. This requirement is also used to calculate transaction fees. Because the Ethereum network limits the amount of gas that can be used by contracts, it is not possible to perform certain operations on the smart contract, which may require excessive calculations such as zero knowledge proof and homomorphic encryption, [30]. In this case, even if the OVN e-voting system is assumed to provide robustness, it is not possible to operate an e-voting protocol on Ethereum for use in nation-wide elections.

In the Ethereum blockchain, the block gas limit specifies the maximum amount of gas that can be used in a block. The block gas limit is used to

minimize both the spread time and block generation time of blocks by limiting the amount of transfers added to a block. In addition, they prevent the creation of effective infinite loops with contracts that call each other. Although the block gas limit changes in each block created, it approximately is about 8,000,000 gas. It is stated that up to 6 voter registration or one vote can be made in one block due to the cost of cryptographic calculations, [30]. Assuming that a block is created in the Ethereum blockchain in about 15 seconds, the registration of 57 million voters alone will take about 4.5 years.

Blockchain solutions designed for E-voting

Bitcoin and Ethereum have scalability problems. It does not allow an e-voting to be performed at the nation-wide due to limitations by design. Given that the constraints arise from the fact that the blockchain network to be used is designed as a financial structure, it is reasonable to create a specialized blockchain network for the e-voting system.

A flow chart to determine whether a blockchain is the appropriate technical solution to solve a problem is presented, [45]. Considering the desired features of e-voting systems according to the formula, there is the potential to use two different types of blockchains according to the selection of writers into ledger in the system. These are public permissioned blockchain and public permissionless blockchain.

The first is the use of a permissioned blockchain. If a permissioned blockchain is chosen, the selection of the nodes that are allowed to write to the blockchain is required. The nodes granted the right to write will need to be identified by a central authority. In a democratic election, although it is assumed that the task of selecting these nodes is fulfilled honestly, problems originated from centralization may arise. Although the centralized execution of elections is undesirable, e-voting with permissioned blockchain facilitate the use of more efficient distribution of trust to wide range of entities and consensus mechanisms.

We believe that permissioned public blockchains specifically built for and used only for e-voting schemes would be a solution. Nonetheless, authorized nodes should be carefully established, in order to convince the voters on transparency of the elections.

5. Conclusion

In this study, requirement analysis of e-voting systems designed with blockchain technology, which has remarkable features for e-voting, have been made. Blockchain technology is considered to be a new era in the field of e-voting, that has the potential to be an important step in direct democracy. As can be seen in the Table 1, although blockchain-based e-voting systems can provide the verifiability of an election, it has not yet matured in terms of e-voting requirements.

It is not possible to use public permissionless cryptocurrency blockchains with high node participation such as Bitcoin, Ethereum, Zcash, which are accepted as secure, for national scale e-voting systems. The main constraints to this situation are that they are designed as an unauthorized type of financial structure, as discussed in the Section 4. The main reason why e-voting requirements cannot be fully fulfilled is that they are designed for the criteria of an economic value.

Even if the issues listed in Section 4 are fixed, utilizing economically valuable assets in voting schemes would give rise to ethical discussion. Voting is a right for eligible voter, thus requir-

ing some monetary costs for voting, or binding votes with valuable cryptocurrencies would result in new problems. For example, in elections with an economically high-value cryptocurrency, there is a possibility that voters may choose to keep the money that is conveyed to them for voting or to use it for other purposes. This constitutes a situation which is contrary to the objective of increasing democratization and thus the nature of elections.

We believe that building an e-voting scheme on a permissioned blockchain rather than cryptocurrency ledgers is more promising. This will also create more scalable system and evade the ethical debates, provided with diverse entities permitted as authenticated nodes.

Acknowledgments

The authors would like to thank to the anonymous reviewers for their insightful comments and reviews.

References

[1] J. Benaloh and D. Tuinstra, "Receipt-free secret-ballot elections," in *STOC*, vol. 94, 1994, pp. 544–553.

[2] D. Chaum, "Blind signatures for untraceable payments," in *Advances in cryptology*. Springer, 1983, pp. 199–203.

[3] P. A. Rezende, "Electronic voting systems–is brazil ahead of its time," *RSA CryptoBytes*, vol. 7, no. 2, 2004.

[4] Ü. Madise and T. Martens, "E-voting in estonia 2005. the first practice of country-wide binding internet voting in the world," *Electronic voting*, vol. 86, no. 2006, 2006.

[5] J. Gerlach and U. Gasser, "Three case studies from switzerland: E-voting," *Berkman Center Research Publication No*, vol. 3, p. 2009, 2009.

[6] S. Wolchok, E. Wustrow, D. Isabel, and J. A. Halderman, "Attacking the washington, dc internet voting system," in *International Conference on Financial Cryptography and Data Security*. Springer, 2012, pp. 114–128.

[7] I. S. G. Stenerud and C. Bull, "When reality comes knocking norwegian experiences with verifiable electronic voting," in *5th International Conference on Electronic Voting 2012 (EVOTE2012)*. Gesellschaft für Informatik eV, 2012.

[8] J. A. Halderman and V. Teague, "The new south wales ivote system: Security failures and verification flaws in a live online election," in *International conference on e-voting and identity*. Springer, 2015, pp. 35–53.

[9] S. J. Lewis, O. Pereira, and V. Teague, "Ceci nest pas une preuve," 2019.

[10] S. Popoveniuc, J. Kelsey, A. Regenscheid, and P. Vora, "Performance requirements for end-to-end verifiable elections," in *Proceedings of the 2010 International Conference on Electronic Voting Technology/Workshop on Trustworthy Elections (EVT/WOTE'10)*. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–16. [Online]. Available: https://www.usenix.org/legacy/event/evtwote10/tech/full_papers/Popoveniuc.pdf

[11] J. Benaloh, R. Rivest, P. Ryan, P. Stark, V. Teague, and P. Vora, "End-to-end verifiability," arXiv preprint arXiv:1504.03778, 2015.

[12] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008, (Accessed: 2019-02-18).

[13] Z. Zhao and H. Chan, "How to vote privately using bitcoin," in *International Conference on Information and Communications Security*. Springer, 2015, pp. 82–96.

[14] A. B. Ayed, "A conceptual secure blockchain-based electronic voting system," *International Journal of Network Security & Its Applications*, vol. 9, no. 3, pp. 01–09, 2017.

[15] F. Sheer Hardwick, A. Gioulis, R. Naeem Akram, and K. Markantonakis, "E-voting with blockchain: An e-voting protocol with decentralisation and voter privacy," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 1561–1567.

[16] S. Gajek and M. Lewandowsky, "Trustless, censorship-resilient and scalable votings in the permission-based blockchain model."

[17] F. P. Hjálmarsson, G. Hreioarsson, M. Hamdaqa, and G. Hjálmtỳsson, "Blockchain-based e-voting system," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*. IEEE, 2018, pp. 983–986.

[18] B. Shahzad and J. Crowcroft, "Trustworthy electronic voting using adjusted blockchain technology," *IEEE Access*, vol. 7, pp. 24 477–24 488, 2019.

[19] R. Tso, Z. Liu, and J. Hsiao, "Distributed e-voting and e-bidding systems based on smart contract," *Electronics*, vol. 8, no. 4, p. 422, 2019.

[20] D. Khoury, E. F. Kfoury, A. Kassem, and H. Harb, "Decentralized voting platform based on ethereum blockchain," in *2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*. IEEE, 2018, pp. 1–6.

[21] C. K. Adiputra, R. Hjort, and H. Sato, "A proposal of blockchain-based electronic voting system," in *2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2018, pp. 22–27.

[22] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting

scheme for large scale elections," in *Advances in Cryptology — AUSCRYPT '92*, J. Seberry and Y. Zheng, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 244–251.

[23] L. F. Cranor and R. K. Cytron, "Sensus: a security-conscious electronic polling system for the internet," in *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, vol. 3, Jan 1997, pp. 561–570 vol.3.

[24] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, ser. WPES '05. New York, NY, USA: ACM, 2005, pp. 61–70.

[25] O. Cetinkaya and D. Cetinkaya, "Towards secure e-elections in Turkey: Requirements and principles," in *The Second International Conference on Availability, Reliability and Security (ARES'07)*, April 2007, pp. 903–907.

[26] J. Cruz and Y. Kaji, "E-voting system based on the bitcoin protocol and blind signatures," *IPSJ Transactions on Mathematical Modeling and Its Applications*, vol. 10, no. 1, pp. 14–22, 2017.

[27] P. Tarasov and H. Tewari, "Internet Voting Using Zcash," Cryptology ePrint Archive, Report 2017/585, 2017, https://eprint.iacr.org/2017/585.

[28] Y. Wu, "An e-voting system based on blockchain and ring signature," *Master. University of Birmingham*, 2017.

[29] Y. Liu and Q. Wang, "An e-voting protocol based on blockchain," Cryptology ePrint Archive, Report 2017/1043, 2017, https://eprint.iacr.org/2017/1043.

[30] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 357–375.

[31] R. Alyoshkin, "Polys online voting system. whitepaper." [Online]. Available: https://polys.me/assets/docs/Polys_whitepaper.pdf

[32] S. Heiberg, I. Kubjas, J. Siim, and J. Willemson, "On trade-offs of applying block chains for electronic voting bulletin boards," *E-Vote-ID 2018*, p. 259, 2018.

[33] F. Reid and M. Harrigan, "An analysis of anonymity in the bitcoin system," in *Security and privacy in social networks*. Springer, 2013, pp. 197–223.

[34] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE Symposium on Security and Privacy*, May 2014, pp. 459–474. [Online]. Available: https://doi.org/10.1109/SP.2014.36

[35] "Zcash user documentation, viewing key," 2018, (accessed 04.2018). [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/zcash/latest/zcash.pdf

[36] R. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in Cryptology — ASIACRYPT 2001*, C. Boyd, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 552–565.

[37] F. Hao, P. Ryan, and P. Zieliński, "Anonymous voting by two-round public discussion," *IET Information Security*, vol. 4, no. 2, pp. 62–67, 2010.

[38] A. Kiayias and M. Yung, "Self-tallying elections and perfect ballot secrecy," in *International Workshop on Public Key Cryptography*. Springer, 2002, pp. 141–158.

[39] D. Khader, B. Smyth, P. Ryan, and F. Hao, "A fair and robust voting system by broadcast," *Lecture Notes in Informatics (LNI), Proceedings-Series of the Gesellschaft fur Informatik (GI)*, pp. 285–299, 2012.

[40] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[41] M. Moser, "Anonymity of bitcoin transactions," 2013.

[42] A. E. Gencer, S. Basu, I. Eyal, R. Van Renesse, and E. G. Sirer, "Decentralization in bitcoin and ethereum networks," *arXiv preprint arXiv:1801.03998*, 2018.

[43] "T.C. Yksek Seim Kurulu, SSS," 2019, (accessed 05.2019). [Online]. Available: http://ysk.gov.tr/tr/sss/1523

[44] "Bitcoin Transaction Fees," 2019, (accessed 19.08.2019). [Online]. Available: https://bitcoinfees.info

[45] K. Wüst and A. Gervais, "Do you need a blockchain?" in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 45–54.