On Trade-offs of Applying Block Chains for Electronic Voting Bulletin Boards

Sven Heiberg[1], Ivo Kubjas[1], Janno Siim[3,4], and Jan Willemson[2,3]

[1] Smartmatic-Cybernetica Centre of Excellence for Internet Voting
Ülikooli 2, 51003 Tartu, Estonia
{sven,ivo}@ivotingcentre.ee
[2] Cybernetica AS, Ülikooli 2, 51003 Tartu, Estonia
janwil@cyber.ee
[3] Software Technology and Applications Competence Center
Ülikooli 2, 51003 Tartu, Estonia
[4] Institute of Computer Science, University of Tartu
Ülikooli 18, 50090 Tartu, Estonia
jannosiim@gmail.com

Abstract. This paper takes a critical look at the recent trend of building electronic voting systems on top of block chain technology. Even though being very appealing from the election integrity perspective, block chains have numerous technical, economical and even political drawbacks that need to be taken into account. Selecting a good trade-off between desirable properties and restrictions imposed by different block chain implementations is a highly non-trivial task. This paper aims at bringing some clarity into performing this task. We will mostly be concentrating on public permissionless block chains and their applications as bulletin board implementations as these are the favourite choices in majority of the recent block chain based voting protocol proposals.

1 Introduction

In virtually all of the modern democratic societies, democracy (translated from Greek roughly as *rule of the people*) is implemented via some sort of public opinion polling e.g. voting on the elections of representative bodies.

Regrettably, the process of public polling is very fragile. Many things can go wrong, and have gone wrong in the history of elections. To address the historically experienced problems, many requirements have been put forward and many measures have been developed to meet them. Contemporary democratic nations have thick rule books describing how to run elections so that the number of problems would not rise above the threshold where general public would start questioning legitimacy of the elected bodies.

On the other hand, extensiveness of the rule book is actually a problem of its own. Having many (possibly even contradictory) requirements makes it difficult to make sure all of them are followed, which in turn translates to decreased transparency of the whole process.

The main measure to improve the transparency of election processes is to make them more publicly auditable. In case of electronic voting, this can only be achieved if as much data about the voting as possible can be accessed by public observers. In the end of the auditing procedures, the observers should agree on the outcome, which presumes that they also must be given the same input to start with.

Presenting a uniform view on some digital assets to several independent parties is a surprisingly hard task known as the problem of setting up a *bulletin board*. The first theoretically sound proposals to solve this problem for electronic voting have emerged only in the recent years [11,10,19], and they are relatively complex to implement. For example, an election organiser using protocol by Culnane and Schneider [11] needs to find four independent participants to achieve adversarial tolerance against one dishonest party. With public permissionless financially incentivised block chains the distributed ledger infrastructure already exists and the required organisational and technical effort to use it is intuitively less than setting up a new election specific bulletin board.

Block chain technology has been identified as a useful tool in order to address various auditing challenges. Already in 2007, Sandler and Wallach described the idea of hash linking of votes to guarantee their integrity [35], applying it later in the VoteBox system [34]. In 2011, Benaloh and Lazarus proposed a similar approach to mitigate their Trash attack [6], and in 2013 Bell *et al.* used the same idea as a part of STAR-Vote system [5].

In recent years, the number of similar proposals has risen considerably, and block chains have been pushed as an almost miracle solution to integrity problems. There are numerous academic [41,23,26,29,7,42,20,8,33,12,21,39] and market-oriented[5] proposals aiming at bringing block chains into voting processes. Unfortunately, the level of information provided about these initiatives (especially market-oriented ones) varies a lot and is often limited.

The current paper aims at putting together a higher-level view on different aspects of using block chain technology for electronic voting. Note, however, that we are not targeting a fully systematic treatment of the topic and keep the approach somewhat informal.

The paper is organized as follows. Section 2 makes a short introduction to block chains and their proposed usages for electronic voting. Section 3 points out the main shortcomings not very well addressed by the current proposals. Next, Section 4 points out the main tradeoffs to decide upon while building a block chain based voting system. Finally, we give our conclusions in Section 5.

[5] Some examples active at the time of this writing, summer 2018, include Follow My Vote `https://followmyvote.com/`, Polys `https://polys.me/`, SecureVote `https://secure.vote/`, VoteWatcher `http://votewatcher.com/`, Agora `https://agora.vote`, e-Vox `http://e-vox.org/`, TIVI `https://tivi.io/`, Boulé `https://www.boule.one/`, Democracy.Earth `https://www.democracy.earth/`, Voatz `https://voatz.com/`, Coalichain `https://www.coalichain.io/`, etc.

2 Block chains

The concept of a block chain does not have a single, universally agreed upon mathematical definition. However, different implementations seem to have a few common points.

- Data storage occurs in *blocks*, where the exact content of a block or its semantics may vary (e.g. it may contain transactions for cryptocurrency applications).
- The blocks are linked into a sequence (also called a *ledger*) using a cryptographic hash function.

The idea of hash linking data items is not at all new, going back to at least early 1990s to the works of Haber, Stornetta *et al.* on digital time stamping [16,4]. However, it seems to be exactly this idea of hash linking that gives block chains the attractive property of integrity assurance, since cryptographic hash functions are supposedly hard to invert, making it difficult to revert the linking once it has been performed.

The real renaissance of block chains happened in late 2008, when a researcher (or a group of researchers) hiding behind the pseudonym Satoshi Nakamoto published what is nowadays known as Bitcoin white paper [28]. Essentially, Nakamoto showed how to use available cryptographic and networking tools to achieve a new type of decentralized consensus protocol.[6]

The core innovation of Nakamoto's proposal is introducing computationally difficult puzzle solving (proof of work) together with financial incentives to consensus building. Whoever solves the puzzle first can create the next ledger block and is rewarded with a certain amount of bitcoins. Due to some similarity with gold mining, the participants in this joint effort are called *miners* or *mining nodes.*

Nakamoto's original motivation was to build a monetary system and there the need for consensus is clear – value exchange can only function correctly when there is a universally accepted way of deciding who has how much money.

However, the problem of obtaining a coherent view on the system in a distributed manner is more general, and this is why the original Bitcoin protocol and infrastructure have been used for a myriad of alternative applications, including voting.

It is worth noting that the original Bitcoin white paper does not present any formal definitions of targeted properties, and contains only a simplified security analysis. Follow-up work by Garay *et al.* [14] and Pass *et al.* [31] have formalized several aspects of block chains and clarified the necessary assumptions to prove the security of Bitcoin protocol.

[6] The origin of the term "block chain" is somewhat unclear. It seems to haver been used in some cryptography-related mailing lists in mid 1990-s, but the first occurrence is hard to track. It is interesting to note that Nakamoto's white paper only uses the term "chain of blocks" and not "block chain".

Another functionality making block chains appealing for legal applications is the ability to run smart contracts. Originally proposed already in mid-1990-s by Nick Szabo [37,38], smart contracts can be though of as a scripting layer on top of a block chain, allowing to check fulfillment of certain conditions, and enforcing predefined actions in the respective cases. There are several block chain frameworks that offer this functionality in a form of a programmable execution environment, including Ethereum Solidity[7], Hyperledger Fabric[8] and Cardano Plutus[9].

In principle, it is possible to formulate any set of rules (say, defining correctness of voting or tallying) in the language of smart contracts. In practice, however, the performance requirements needed to actually run them may become prohibitive. We will come back to this issue in Section 3.5.

Block chains come in several flavours. Bitcoin block chain is an extreme example of a distributed ledger where there is no single trusted entity to coordinate the work, nor to decide which blocks to accept from whom, etc. In this case we speak of a *permissionless ledger*.

However, this is not the only option. It is also possible to set up a block chain where data commitments are only accepted from a predetermined set of nodes, and there may even be an authority deciding that some of the blocks will not be admitted. Such a ledger is called *permissioned*. Block chains built within the Hyperledger framework are examples of such a paradigm.

Similarly, it is not necessarily the case that anyone is given access to the block chain for reading. Depending on whether or not general access is allowed, we speak of *public* or *private* block chains, respectively.

For the most part of this paper we will be treating public permissionless ledgers, and there are several reasons for that. First, such ledgers aim at building a fully distributed consensus mechanism which seems to be an attractive property for electronic voting systems. Second (and probably implied by the first reason), majority of the proposals we have studied in course of this research build on top of public permissionless ledgers (mostly Bitcoin or Ethereum). However, several of our observations hold for other kinds of block chains as well.

2.1 How to use block chain for electronic voting?

The obvious application for a block chain in electronic voting is to use it as a bulletin board for committing the state of an electronic ballot box. Dedicated bulletin board protocols for electronic voting do exist [11,10,19], but the assumptions made for achieving the security target are expensive to fulfil. For example, the protocol presented by Culnane and Schneider [11] requires correct behaviour by strictly more than 2/3 of the peers. Hence, in order to tolerate one malicious party, at least three honest peers are required. An election organiser willing to apply such a bulletin board protocol for integrity and transparency

[7] `https://ethereum.org/`

[8] `https://www.hyperledger.org/projects/fabric/`

[9] `https://cardanodocs.com/technical/plutus/introduction/`

reasons would need to find a significant number of independent participants to provide adversarial tolerance. Note that the situation is different when setting up a fault tolerant distributed storage where all nodes actually could be hosted by a single organization and same personnel. With bulletin board we must take into account that some nodes may be malicious. If a single entity is running the bulletin board, then we get no adversarial tolerance against this entity.

In case of a public permissionless financially incentivised block chain there is no problem of finding independent participants – the distributed ledger infrastructure exists and its security is maintained by a number of parties. It is only a matter of finding good use of this infrastructure for election purposes. There is a remarkable number of proposals suggesting variety of approaches involving also (but not only) block chain as a bulletin board.

- It is possible to utilise smart contracts to enforce voting rules [25,33,21,2].
- Several proposals implement vote casting via Bitcoin cryptocurrency transfer [7,42,41,29,26,23,8]. However, they differ a lot in implementation details, e.g. eligibility verification and voter authentication (see Section 3.1).
- To enhance voter anonymity, Takabatake *et al.* propose using Zerocoin instead of Bitcoin [39].
- Block chain based bulletin board can be used to directly commit votes, like e.g. in the schemes of Polys [3] and Agora [13].
- A related approach is taken by the TIVI framework where block chain is used for digital time stamping of certain integrity-critical events, e.g. vote submission [1].
- A few US-based initiatives like VoteWatcher and Votebook are still paper voting systems at their core, but use either a public or private block chain for committing certain data required for later verification (e.g. scanned paper ballots) [30].
- A weaker version of the last approach was used by the Agora team in Sierra Leone where they typed in the votes read out loud by the election officials, and used block chain to verify the count.[10]

These approaches can also be combined. For example one may commit all the votes to a private block chain and make commitments to a public ledger like Bitcoin from time to time; such a solution is implemented e.g. in Agora and VoteWatcher.

There are also other voting systems that could potentially make use of controlled bulletin boards. For example, there is a Vote Registration Service compo-

[10] `https://medium.com/agorablockchain/agora-official-statement-regarding-sierra-leone-election-7730d2d9de4e`. The 2018 Sierra Leone event was advertised by the Agora team as the “world’s first ever blockchain elections” in their press release `https://agora.vote/pdf/Agora_Press-release_SL2018.pdf`. However, the mode that the block chain was eventually used in offered little to no advantages over a simple independent Excel-aided recount. After this disclosure, the Agora team took down their press release, but one can still find a copy of it cached by Google.

nent in the Estonian IVXV scheme [18] which can in principle be implemented on top of a private or public block chain.

In the next Section, we will discuss some of the common concerns not very well addressed in various proposals.

3 Shortcomings of block chain based voting systems

We argue in the following that there are several problems and limitations with using block chain in electronic voting.

3.1 Eligibility verification

Even though the overall target of block chain based voting systems is increasing transparency and public verifiability, there are several aspects of elections, correctness of which can not be established on the block chain. One of the prominent examples is deciding the eligibility of voters.

This problem manifests itself clearly in case of the proposals where Bitcoin transactions are used for vote casting [7,42,41,29,26,23,8,39]. The original design goal of Bitcoin as a cryptocurrency was to provide anonymous transfers, and this is something that contradicts the needs of voting. Even though we typically want the votes to be secret, the voters should still be uniquely identifiable in order to determine eligibility and provide uniformity (so that no-one would get more than one vote).

Hence an identity provider is required one way or another. Note that whoever that provider is, it has the ability to flag ineligible persons as eligible or vice versa, or even define new virtual voters who do not have a corresponding physical person [39]. The only setup where this problem can be ignored are small-scale boardroom type of elections where all the voters know each other. But for even a moderate size elections this can lead to a ballot box stuffing attack that is undetectable by any verification mechanism that may run on top of the block chain. Hence block chain does not remove the need for external trust anchors.

An interesting approach to identity validation in a distributed manners has been taken by the Democracy Earth Foundation. Their manifesto [2] proposes participant registration via creating a video where enough personal details are stated, and seeking acceptance to this video from the community. While this approach may follow the spirit of decentralized governance, it is hard to imagine such an identity creation mechanism to be accepted for official national elections any time soon. Crowd-sourcing-based identity providers are only applicable in small-scale community settings.

Even if we accept the need for external trust dependency for eligibility verification, the problem of implementing the link from identity provider to the block chain still remains. Different frameworks have different approaches to tackle this issue.

Wu proposes using ring signatures to provide anonymity [41], but this introduces a non-trivial setup procedure and a significant performance penalty, making the solution unusable even for moderately-sized elections.

Noizat requires each candidate to issue a key pair to every voter, organizing the public parts into a Merkle tree. Two more keys per voter are generated by the other components of the system, and to cast a vote, a 2-of-3 multisignature scheme is used [29]. However, this non-trivial cryptographic machinery still does not resolve the problem of eligibility verification, but actually makes it worse, forcing all the candidates to manage voter lists.

Lee *et al.* acknowledge the need to have Trusted Third Party (TTP) for identity confirmation, but to implement this procedure they only propose a password-based registration and authentication mechanism [23]. As a result of authentication, the user gets a confirmation that her asymmetric key pair is declared eligible, but the overall system is only as secure as the original, user-created password. Also, there is a strong reliance on the honest behavior of TTP. Dishonest TTP could easily manipulate the result, for example, by claiming that votes for a candidate whom the TTP does not like came from unregistered accounts.

Bistarelli *et al.* propose using Anonymous Kerberos protocol for voter authentication [7]. This approach has the benefit of relying on a relatively standard authentication mechanism that does not impose overly restrictive performance limitations. However, voter anonymity can be broken when Authentication Server and Token Distribution Server collude. More servers can be added to address this issue, but this would make the protocol more complex. Still, out of all the proposals we have studied, the one by Bistarelli *et al.* seems the most viable.

Zhao *et al.* [42] have developed a group incentive mechanism to motivate a group of voters to participate in tallying, but they completely ignore the problem of eligibility verification. Similarly, the voter identification problem is ignored by the considered frameworks that use block chain as a generic bulletin board implementation, including Polys [3] and Agora [13].

There is also a more general problem common to several of the approaches described above.

Namely, elections may last over a longer period of time (say, a week). Eligibility status of potential voters may change during this period (e.g. someone may turn 18 or die). It is, of course, possible to ignore this problem and only let people eligible at a certain point of time to vote [39]. However, another conceivable viewpoint is that it would be more fair to update the list of voters as the changes in eligibility occur.

For small-scale elections with a public voter list this issue can, in principle, be solved by committing the full voter list together with the potential updates to the ledger. But for larger events with potentially non-public sets of voters, only the schemes of Lee *et al.* [23] and Bistarelli *et al.* [7] have some potential to address this issue as they are using an online protocol with the identity provider as a backend service.

3.2 Ensuring ballot secrecy

In addition to integrity, voting protocols also have confidentiality requirements. Voting in majority of the elections is carried out by secret ballot, whereas the verifiability of the correctness of the final tally is still a desired property. Additionally – the tally must not be available before a fixed moment in time. Election result is public information, once it has been released, at the time of the voting the partial results are considered confidential, because of the potential impact on the voter behaviour.

Public block chain contents are public by definition, anybody who hosts a full Bitcoin or Ethereum node has access to all the data published there. The first implication of the fact is that unless the ballots on the block chain are obfuscated, anybody has access to the partial results, thus violating the common requirement of not releasing partial results too early.

Obfuscated ballots (via public-key encryption or some commitment scheme) on the block chain introduce the question of verifiable de-obfuscation. The block chain is of no use if we cannot verify the consistency of the tally. Note that in case of encryption, we cannot store the private key on the block chain, and some external tallying authority will be needed at least for the key management if not for the verifiable tally.

Self-tallying voting protocol Open Vote Network [17] (OVN) has been implemented as an Ethereum smart contract [25]. This protocol does not rely on any third parties for the tally, ballot secrecy is ensured jointly by all voters who have to participate in both obfuscation and de-obfuscation in order for the tally to be successful. The protocol provides privacy for the voters and benefits from the block chain as a public broadcast channel making it an excellent protocol for smart contracts. The downside of the OVN is its fragility – even one voter can prevent tallying simply by being absent. The properties of the protocol make it usable for small scale elections in a boardroom environment, but not for any large scale elections.

3.3 Consistency verification

The common characteristic of all the proposed block chain voting protocols is committing votes (in a plain or encrypted form) to the ledger. Committing just for the sake of it does not make any sense, hence there should exist a routine of checking certain claims about the commitments.

What these claims exactly are depends largely on the scheme. They can include zero-knowledge proofs of correct vote formatting, correspondence to some eligibility criteria, signature validity, block chain integrity, etc. Hopefully there is a relatively short list of them so that the corresponding checks can be implemented with a reasonable amount of code. On the other hand, none of the block chain voting scheme proposals we considered has claimed a full list of checks needed to establish internal consistency of the ledger. The most detailed attempt to describe consistency rules was made for Votebook by Kirby *et al.*, but even they only state that there should be sufficient information released to the public so that "blockchain can be counted correctly" [20].

There are several aspects to consider when defining internal consistency rules of a ledger and determining the final tally outcome.

One problem is dealing with simple, honest (or dishonest) mistakes. For example, due to a programming error, network delay or any other stochastic issue some of the data items required for later auditing may be missing or malformed. The originally desired property of ledger immutability suddenly becomes an issue, since one can not simply replace malformed items or add missing ones to where they should have been. Hence there must be an option of adding data blocks with exception handling and overriding semantics to the ledger, and the auditing logic must be capable of dealing with them. Taking all kinds of potential options of malformedness into account, consistency verification may become complicated beyond what one feels comfortable with.

The problem of unforeseen situations is clearly acknowledged by the Agora team [13]. As a resolution, they suggest using predetermined human auditors who have a power of any ruling they see fit to settle the matter. The ruling should be written down, signed and committed to the ledger. As the unforeseen situations do not follow any patterns, the signed statement is probably also human text that is then in turn open to misinterpretations, defying the purpose of block chain transparency. An interesting open question is whether such statements could be issued in the form of smart contracts.[11]

Another problem is managing repeated vote submissions. Depending on the setup, this may be either a necessary anti-coercion measure (see e.g. [18]), or an unwanted side effect of remote voting. Either way, some rules are necessary to determine which one of the submitted votes will be counted [41,23].

Smart contracts could be used to efficiently describe audit logic and prevent stochastic errors as the ones described above, but they come with a price. Namely, one has to spend a certain amount of resource (e.g. *gas* in case of Ethereum [40]) for the transactions, where the exact amount of the resource depends on the size of transaction. As a result, the authors of OVN estimate that their framework can only reasonably accommodate about 50 voters [25], whereas Ramachandran and Kantarcioglu limit their proposal to 100 voters [33].

We conclude that accounting for all the special cases that might occur is far from being trivial. The whole idea of using block chain as a ledger is to make its consistency independently verifiable. The definition of necessary and sufficient conditions for independent consistency verification must form an integral part of any proposal for a block chain based voting system.

3.4 Transaction registration issues

A common problem of public permissionless ledgers like Bitcoin and Ethereum is that the blocks in the chain are limited both in size and frequency, leading to a very low amount of transactions per second that the ledger can process (e.g.

[11] It is interesting to note that even though Agora is claimed to have an elaborate consistency verification mechanism, none of it was used in the Agora Sierra Leone event, where even the most basic block chain explorer tool was not provided for auditing `http://en.rfi.fr/africa/20180319-sierra-leones-electoral-commission-distances-itself-use-blockchain-during-polls`.

7 in case of Bitcoin and 15 in case of Ethereum)[12]. As this resource is shared worldwide, committing one transaction per vote to such a ledger is not realistic even for moderately-sized elections.

For electronic voting we need to provide voters and election officials with precise upper bounds on the transaction confirmation time. For the sake of this paper we assume that transaction is confirmed if it is included in any valid block, although more realistic requirement would be that several blocks (say, 6 in case of Bitcoin) extend the block containing the transaction. The time required to mine a new block in Bitcoin is 10 minutes on average, but the worst-case time can be much longer e.g. on the 1st of April, 2018, time between blocks number 516036 and 516037 is roughly 54 minutes.

Block generation time follows roughly the exponential distribution (although folklore, Bowden *et al.* [9] argue that this is not precisely correct if taken into account changing proof of work difficulty and network delays) where the cumulative distribution function $F(x) = 1 - e^{-x/\lambda}$ expresses the probability that a block is generated in x minutes, given the average rate of block generation λ (for Bitcoin $\lambda = 10$).

This allows us to make some rough estimates of block generation times. Probability that block generation takes more than 10 minutes is $1 - F(10) = e^{-1} \approx 0.37$, i.e. roughly third of the blocks take more than 10 minutes to generate. Probability that a block generation takes more than 50 minutes is $1 - F(10) = e^{-5} \approx 0.007$, hence we would expect to see 2 or 3 such blocks per day meaning that voters would have to wait more than 50 minutes to have any assurance about their vote actually being recorded.

Another serious drawback of public permissionless block chains is that the success of actually accepting a transaction into the ledger depends on financial incentives rather than the legislative need to create an immutable audit trail.

Garay *et al.* [14] prove the *liveness* property for Bitcoin block chain which informally says that if a transaction is broadcasted to honest nodes for a certain number of consecutive rounds, then the transaction will eventually be included in the block chain. However, this property relies on the assumption of synchronous network (see Pass et al. [31] for asynchronous networks) and that majority of hashing power is controlled by honest miners.

In particular, the last assumption is not quite true in real life – most miners behave rationally and will give preference to transactions with the highest fees. Transaction with no or little transaction fees might get completely neglected. As a result, successful data commitment can not be guaranteed in these types of block chains.

The main approach to speed up the process is to increase transaction rewards for the miners, rising the overall cost of the elections (but still not achieving a 100% guarantee due to the e.g. block-size limits). Note that transaction issuer will find out that the fee he offered for to the miners was too low only after some time has passed.

[12] `https://github.com/ethereum/wiki/wiki/Sharding-FAQ`

This problem is discussed most deeply by Noizat [29] who proposes various methods including using relatively high transaction fees to increase transaction priority, and election organizer becoming one of the miners. Even with using these methods, transaction confirmation may take several days [29,7,8]. Noizat and Bogucki [29,8] argue that this may be fine for at least some types of elections, but in our opinion, such a restriction limits applicability of block chain based voting remarkably.

The result of transaction confirmation being delayed (or even "forgotten" if it stays in the pool of pending transactions for too long) is devastating. The publicly verifiable audit trail will have blocks missing or occurring in the ledger in a wrong order. This will further complicate the consistency verification logic (see Section 3.3).

3.5 Performance issues

In order to incentivise block chain node contributions to the chain creation, transactions cost (crypto)money. On the other hand, cryptocurrencies like Bitcoin are very volatile. For example, in their paper published in April 2017, Bistarelli *et al.* [7] used the estimate 1₿=547€ to compute the overall cost of running elections. However, by December 2017 the value of Bitcoin had reached over 16000€, making voting over 30 times more expensive. It is large monetary risk for election organiser. In those protocols, where the voter must initiate the transaction it effectively introduces a fee for voting.

Another performance problem specific to Bitcoin comes from the fact that it was designed primarily for monetary transactions and its ledger's ability to accommodate other types of data is rather limited. The most well known method, is to use the `OP_RETURN` field for free-form input, and the length of that field is only 80 bytes. This in turn means that instead of full data blobs, only their hashes can be committed to the Bitcoin block chains. If this is done carelessly, it can lead to new vulnerabilities.

A good example of this type of misdesign can be observed in the scheme proposed by Wu [41]. The scheme relies on ring signatures, but they can unfortunately be rather large. So, instead of signature σ, the hash $h(\sigma)$ is committed to the Bitcoin block chain. The original σ is kept by the Election Authority without any integrity protection. This means that an independent auditor has no way of resolving the dispute between a voter and the Election Authority if the latter e.g. claims that it has not received the signature σ from the voter in the first place. A potential solution is using a local block chain to store all the data required for auditing, and only committing snapshot hashes of the local block chain to the Bitcoin ledger. Such a solution is deployed e.g. by Agora and VoteWatcher [13,30].

Additional options for storing arbitrary data in Bitcoin block chain have been proposed in [36]. It is possible to store nearly 100kB in a single transaction. Content insertion services that fragment data over multiple transactions have been used to store files as large as 310.72kB in Bitcoin block chain [24]. It is, however, worth noting that usage of the `OP_RETURN` field or any other mechanism for storing non-currency data is discouraged by the Bitcoin development team.[13]

General purpose block chains (e.g. Ethereum) are better suited for arbitrary data storage. In principle, the storage available to Ethereum smart contracts is limited to 2^{256} words of 32 bytes, but it must be noted that any write operation requires 20k gas per 32 byte word [40].

The block chain storage is more general issue. A full copy of Bitcoin ledger has surpassed 150 GB [13], which makes it unreasonable to store it at every node. Also, due to objectionable content being committed to the Bitcoin block chain, storing certain parts of it can even be considered illegal [24].

As a possible solution to transaction cost, latency and storage problems, Agora uses several layers of block chains. On a lower level, a dedicated ledger is run, and only periodic aggregated snapshots are then committed to the Bitcoin block chain [13]. A similar approach is deployed by VoteWatcher. This can indeed reduce the Bitcoin transaction costs, but such an architecture complicates auditing, so essentially we get a trade-off between direct cost and complexity. This approach does not solve any transaction registration issues, meaning that some snapshots will be published in average time, some snapshots may take hours to publish and there are no guarantees.

3.6 Centralization of mining power

Even though one of the main targeted properties of permissionless public block chains is decentralization, this property does not necessarily hold in practice. For example, the most popular block chain implementations Bitcoin and Ethereum (unintentionally) incentivize the miners to group into centrally managed mining pools to allow for a constant stream of rewards to the pool participants. Additionally, being connected into geographically close nodes allows for faster broadcast of the mined blocks into pool members and thus gives an advantage in starting mining a new block earlier than other pools could [27].

Even more, the hash rate distribution within the pools is highly concentrated to a few nodes. In [27], Miller *et al.* have found that in 2015, only 2% of the Bitcoin mining nodes held three quarters of the mining power.

The state of centralization has not improved significantly since then. In a recent work [15], Gencer *et al.* have shown that in Bitcoin, four of the biggest pools hold 53% of average mining power. Ethereum shows a similar tendency towards centralization – three top Ethereum pools hold 61% of the mining power.

Centralization introduces a clear political risk. Several estimates indicate that the top miners are exclusively located in China, with around 80% of mining power of the Bitcoin network belonging to Chinese pools[14][15]. This makes it in principle possible to influence majority of the miners from one central political

[13] `https://bitcoin.org/en/release/v0.9.0#opreturn-and-data-in-the-block-chain`

[14] `https://www.buybitcoinworldwide.com/mining/pools/`

[15] `https://altcointoday.com/ethereum-mining-hashrate-distribution-issues/`

authority, thus violating the assumptions required for immutability of the block chain [28]. This might not be a serious concern for smaller elections (say, electing a rector of a university), but is clearly a worry for governmental elections.

4 Trade-offs

Block chain is not a miracle solution for all the voting-related problems. While it has strong integrity properties definitely required by election systems, a block chain ledger is non-trivial to set up and interface with other components. Several trade-offs need to be made and the interplay of these trade-offs may have unwanted consequences.

4.1 Expressive power *vs.* complexity

The first trade-off one needs to consider is between the expressiveness of the claims the block chain commitments have, versus the complexity that one needs to accept while verifying them.

For example, we may want to automate exception handling to the level where the correctness of possible error fixes is certified by smart contracts, but this implies the need for complicated certification logic which is accompanied by performance penalties of running the smart contracts.

Alternatively, we may want to make use of commitments to a public ledger like Bitcoin to utilize the whole power of global trust. However, due to Bitcoin's limited capacity of handling external inputs, we need an extended mechanism of integrity certification (essentially, another layer of block chain ledger below Bitcoin). This in turn complicates the verification logic.

Smart contracts have more potential in terms of expressive power – there is support for arbitrary data structures and the validation logic itself is published to the chain and executed by the nodes. Of course, this has severe implications on performance.

It is a question if in case of general purpose ledgers it is enough to verify the consistency of the ledger with respect to one particular election or should the consistency of the ledger as a whole also be assured, so that the verification of full nodes becomes part of the election audit procedure.

4.2 Small scale *vs.* large scale

There are issues that are potentially easier to solve on small-scale elections.

For example, eligibility verification is much easier when everyone knows everyone else, but this can only be the case in very localized settings. For larger-scale events we have to accept reliance on an external identity provider who may be malicious without detection.

The OVN smart contract [25] is a great example how to ensure ballot secrecy in boardroom elections where we can somehow assure that everybody is going to participate in the whole process. The protocol is clearly not usable for large-scale elections.

Also, several techniques potentially useful for block chain voting (e.g. ring signatures) have a significant performance penalty. For large-scale elections one has to avoid them, also losing the benefits they provide.

4.3 Trust *vs.* cost

Using Bitcoin ledger for commitments is very appealing because of a lot of public trust in the integrity properties it provides and a large community relying on this trust already. However, the cost of Bitcoin transactions (and also transactions in other public ledgers) has been very volatile in the near past. For example, on December 22nd 2017, Bitcoin transaction fee spiked to $55.16.[16]

This means that the election organizers must have a lot of flexibility in budgeting. However, public election authorities tend to operate under budget constraints and prefer well-predictable costs. While one of the targets of block chain voting is reducing the overall price tag of elections [29,26,7], this effect may be reduced by the potential volatility of the costs.

Using local block chains mitigates this problem considerably, but this solution also deprives us of the benefit of public trust. Also, the costs of setting up and running a local block chain are non-negligible, although better-predictable.

4.4 Usability *vs.* individual verifiability

In both academic and industrial communities a general consensus is that the electronic voting schemes have to provide some kind of verifiability. The individual aspect of the verifiability, as defined in [22], should convince the voter that the vote has been stored correctly by the election provider.

To be able to provide individual verifiability, the election provider needs to store the ballot and construct a receipt which can be used for verifying the correct storage. The receipt could contain storage location identifier, proof of registering the ballot [18], block chain block identifier etc.

In case block chain is used for storing any aspect of the ballot, then independently of the underlying block chain technology used, the voter needs to wait until the information has been stored in the block chain. However, the latency of block chain storage can be rather long and the variance can be large. In extreme cases, the storage confirmation may even not arrive in reasonable time (see Section 3.4).

From the voter's perspective, this means that either receiving the receipt or verifying the ballot may take considerable time. The voter would then need to return later to complete verification and this decreases usability of the whole scheme. As a result, the number of vote verifications is expected to drop, together with the overall public confidence in election integrity.

[16] `https://bitinfocharts.com/comparison/bitcoin-transactionfees.html`

5 Conclusions

Even though applying block chain as an integrity assurance measure may seem straightforward for electronic voting, extra assumptions and trade-offs required make setting up such a system a non-trivial task.

Many of the proposals that we have considered try to make use of public permissionless economically incentivised block chains, mostly either Bitcoin or Ethereum. On one hand this makes a lot of sense, since they are widely used and trusted. However, these ledgers have major drawbacks like the tendency for centralisation, providing no guarantees of transaction acceptance and performance limitations. In our opinion, due to these drawbacks such block chains have very limited use for electronic voting. To be considered useful for voting, the block chain must accept authorized commitments immediately and unconditionally.

None of the proposals we studied had a complete description of conditions that need to be verified in order for the voting event to be considered right. Currently, using smart contracts seems to be the most systematic approach to deal with this issue, but systems using smart contracts so far imply a significant performance penalty, strongly limiting e.g. the number of voters.

Also, majority of the proposals ignored the need for exception handling. We conjecture that full consistency verification of block chain based voting systems is rather complex, defying the original target of transparency. It may be the case that simplicity of the verification routines needs to be recognised as a development requirement of its own right.

We would like to conclude the paper by citing Josh Benaloh [32]:

> *I find myself debunking a blockchain voting effort about every few weeks. It feels like a very good fit for voting, until you dig a couple millimeters below the surface.*

Even though such a statement may be a bit too categorical, we agree that in all of the proposals we considered, many of the shortcomings and trade-offs of block chains were addressed insufficiently. Considerably deeper research is required to settle a good design for a block chain based electronic voting system.

Acknowledgements

The research leading to these results has received funding from the Estonian Research Council under Institutional Research Grant IUT27-1, European Union's Horizon 2020 research and innovation programme under grant agreement No 780477 (project PRIViLEDGE), and the European Regional Development Fund through the Estonian Centre of Excellence in ICT Research (EXCITE) and the grant number EU48684.

References

1. Online Voting. Successfully Solving the Challenges. TIVI Whitepaper. `http://www.smartmatic.com/fileadmin/user_upload/Whitepaper_Online_Voting_Challenge_Considerations_TIVI.pdf`.

2. The Social Smart Contract, 2018. http://paper.democracy.earth/.
3. Roman Alyoshkin. Polys online voting system. Whitepaper. https://polys.me/assets/docs/Polys_whitepaper.pdf.
4. Dave Bayer, Stuart Haber, and W Scott Stornetta. Improving the efficiency and reliability of digital time-stamping. In *Sequences II*, pages 329–334. Springer, 1993.
5. Susan Bell, Josh Benaloh, Michael D Byrne, Dana DeBeauvoir, Bryce Eakin, Gail Fisher, Philip Kortum, Neal McBurnett, Julian Montoya, Michelle Parker, et al. STAR-Vote: A secure, transparent, auditable, and reliable voting system. *USENIX Journal of Election Technology and Systems (JETS)*, 1(1):18–37, 2013.
6. Josh Benaloh and Eric Lazarus. The trash attack: An attack on verifiable voting systems and a simple mitigation. Technical report, 2011. MSR-TR-2011-115, Microsoft.
7. Stefano Bistarelli, Marco Mantilacci, Paolo Santancini, and Francesco Santini. An End-to-end Voting-system Based on Bitcoin. In *Proceedings of the Symposium on Applied Computing*, SAC '17, pages 1836–1841. ACM, 2017.
8. Brianna Bogucki. Buying Votes in the 21st Century: The Potential Use of Bitcoins and Blockchain Technology in Electronic Voting Reform. *Asper Review of International Business and Trade Law*, 17:59–84, 2017.
9. Rory Bowden, Holger Paul Keeler, Anthony E. Krzesinski, and Peter G. Taylor. Block arrivals in the Bitcoin blockchain. *CoRR*, abs/1801.07447, 2018.
10. Nikos Chondros, Bingsheng Zhang, Thomas Zacharias, Panos Diamantopoulos, Stathis Maneas, Christos Patsonakis, Alex Delis, Aggelos Kiayias, and Mema Roussopoulos. D-DEMOS: A distributed, end-to-end verifiable, internet voting system. In *ICDCS 2016*, pages 711–720. IEEE Computer Society, 2016.
11. Chris Culnane and Steve A. Schneider. A peered bulletin board for robust use in verifiable voting systems. In *IEEE CSF 2014*, pages 169–183. IEEE Computer Society, 2014.
12. Nazim Faour. Transparent Voting Platform Based on Permissioned Blockchain. Master's thesis, Higher School of Economics, National Research University, Russia, 2018. https://arxiv.org/abs/1802.10134.
13. Leonardo Gammar, Bryan Ford, and Jaron Lukasiewicz. Agora. Bringing our voting systems into the 21st century. Whitepaper, version 0.1. https://agora.vote/Agora_Whitepaper_v0.1.pdf.
14. Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *EUROCRYPT 2015: Advances in Cryptology – EUROCRYPT 2015*, volume 9057 of *LNCS*, pages 281–310. Springer, 2015.
15. Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert van Renesse, and Emin Gün Sirer. Decentralization in Bitcoin and Ethereum Networks, 2018. https://arxiv.org/abs/1801.03998.
16. Stuart Haber and W. Scott Stornetta. How to Time-Stamp a Digital Document. In Alfred J. Menezes and Scott A. Vanstone, editors, *Advances in Cryptology-CRYPT0' 90*, volume 537 of *LNCS*, pages 437–455. Springer Berlin Heidelberg, 1991.
17. Feng Hao, Peter Y. A. Ryan, and Piotr Zielinski. Anonymous voting by two-round public discussion. *IET Information Security*, 4(2):62–67, 2010.
18. Sven Heiberg, Tarvi Martens, Priit Vinkel, and Jan Willemson. Improving the Verifiability of the Estonian Internet Voting Scheme. In Robert Krimmer, Melanie Volkamer, Jordi Barrat, Josh Benaloh, Nicole Goodman, Peter Y. A. Ryan, and Vanessa Teague, editors, *E-Vote-ID 2016: Electronic Voting*, volume 10141 of *LNCS*, pages 92–107. Springer, 2017.

19. Aggelos Kiayias, Annabell Kuldmaa, Helger Lipmaa, Janno Siim, and Thomas Zacharias. On the security properties of e-voting bulletin boards. In *Security and Cryptography for Networks - 11th International Conference, SCN 2018, Amalfi, Italy, September 5 - September 7, 2018, Proceedings*, 2018. To appear.
20. Kevin Kirby, Anthony Masi, and Fernando Maymi. Votebook. A proposal for a blockchain-based electronic voting system, September 2016. `http://www.economist.com/sites/default/files/nyu.pdf`.
21. Ali Kaan Koç, Emre Yavuz, Umut Can Çabuk, and Gökhan Dalkiliç. Towards Secure E-Voting Using Ethereum Blockchain, 2018. 6th International International Symposium on Digital Forensic and Security (ISDFS), Antalya, Turkey. `https://www.researchgate.net/publication/323318041`.
22. Steve Kremer, Mark Ryan, and Ben Smyth. Election verifiability in electronic voting protocols. In *Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings*, pages 389–404, 2010.
23. Kibin Lee, Joshua I James, Tekachew Gobena Ejeta, and Hyoung Joong Kim. Electronic voting service using block-chain. *The Journal of Digital Forensics, Security and Law: JDFSL*, 11(2):123–135, 2016. `https://commons.erau.edu/jdfsl/vol11/iss2/8/`.
24. Roman Matzutt, Jens Hiller, Martin Henze, Jan Henrik Ziegeldorf, Dirk Müllmann, Oliver Hohlfeld, and Klaus Wehrle. A Quantitative Analysis of the Impact of Arbitrary Blockchain Content on Bitcoin. In *Proceedings of the 22nd International Conference on Financial Cryptography and Data Security (FC). Springer*, 2018. `http://fc18.ifca.ai/preproceedings/6.pdf`.
25. Patrick McCorry, Siamak F. Shahandashti, and Feng Hao. A Smart Contract for Boardroom Voting with Maximum Voter Privacy. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security*, volume 10322 of *LNCS*, pages 357–375. Springer International Publishing, 2017.
26. Christian Meter. Design of Distributed Voting Systems. Master's thesis, Heinrich-Heine-Universität Düsseldorf, 2015. `https://arxiv.org/pdf/1702.02566.pdf`.
27. Andrew Miller, James Litton, Andrew Pachulski, Neal Gupta, Dave Levin, Neil Spring, and Bobby Bhattacharjee. Discovering bitcoin's public topology and influential nodes, May 2015. `http://cs.umd.edu/projects/coinscope/coinscope.pdf`.
28. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. `https://bitcoin.org/bitcoin.pdf`.
29. Pierre Noizat. Chapter 22 – blockchain electronic vote. In David Lee Kuo Chuen, editor, *Handbook of Digital Currency*, pages 453 – 461. Academic Press, San Diego, 2015.
30. Ryan Osgood. The Future of Democracy: Blockchain Voting. *COMP116: Information Security*, 2016. `http://www.cs.tufts.edu/comp/116/archive/fall2016/rosgood.pdf`.
31. Rafael Pass, Lior Seeman, and abhi shelat. Analysis of the Blockchain Protocol in Asynchronous Networks. In *Advances in Cryptology – EUROCRYPT 2017*, volume 10211 of *LNCS*, pages 643–673. Springer, 2017.
32. Morgen E. Peck. Blockchain world – Do you need a blockchain? This chart will tell you if the technology can solve your problem. *IEEE Spectrum*, 54(10):38–60, October 2017.
33. Aravind Ramachandran and Murat Kantarcioglu. Using Blockchain and smart contracts for secure data provenance management, 2017. arXiv preprint arXiv:1709.10000, `https://arxiv.org/abs/1709.10000`.

34. Daniel Sandler, Kyle Derr, and Dan S Wallach. VoteBox: A Tamper-evident, Verifiable Electronic Voting System. In *17th USENIX Security Symposium*, pages 349–360, 2008.
35. Daniel Sandler and Dan S Wallach. Casting Votes in the Auditorium. In *USENIX/ACCURATE Electronic Voting Technology Workshop*, 2007.
36. Andrew Sward, Vecna OP_0, and Forrest Stonedahl. Data Insertion in Bitcoin's Blockchain, 2017. Computer Science: Faculty Scholarship & Creative Works. `https://digitalcommons.augustana.edu/cscfaculty/1/`.
37. Nick Szabo. Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought, (16)*, 1996.
38. Nick Szabo. Formalizing and Securing Relationships on Public Networks. *First Monday*, 2(9), 1997.
39. Yu Takabatake, Daisuke Kotani, and Yasuo Okabe. An anonymous distributed electronic voting system using Zerocoin. *IEICE Technical Report*, 116(282), 11 2016. `http://hdl.handle.net/2433/217329`.
40. Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger EIP-150 REVISION (759dccd - 2017-08-07), 2017. Accessed: 2018-01-03.
41. Yifan Wu. An E-voting System based on Blockchain and Ring Signature. Master's thesis, University of Birmingham, 2017. `https://www.dgalindo.es/mscprojects/yifan.pdf`.
42. Zhichao Zhao and T.-H. Hubert Chan. How to Vote Privately Using Bitcoin. In Sihan Qing, Eiji Okamoto, Kwangjo Kim, and Dongmei Liu, editors, *Information and Communications Security: 17th International Conference, ICICS 2015, Beijing, China, December 9-11, 2015, Revised Selected Papers*, volume 9543 of *LNCS*, pages 82–96. Springer, 2016.