Blockchains and Voting: Somewhere between hype and a panacea
(A Position Paper)

Yomna Nasser[1,2], Chidinma Okoye[3], Jeremy Clark[3], and Peter Y A Ryan[4]

[1] EFF
[2] University of Waterloo
[3] Concordia University
[4] University of Luxembourg

Abstract. With the current attention (and some would say "hype") on new blockchain applications, it is unsurprising that a number of researchers, developers, journalists, and start-ups have posited blockchain technology as the missing link for transparent, verifiable election systems. Are they really? We approach this question in a systematic way from both directions: (i) beginning with a blockchain, what challenges arise from layering a voting system on top, and (ii) examining existing verifiable voting systems, how might a blockchain augment their properties? Our position is that blockchains are a useful augmentation to verifiable voting in some circumstances and may introduce interesting ways of voting in non-traditional settings, but blockchains are not a panacea. For current public sector elections in particular, any holistic approach to security will at most use a blockchain as a supporting component in a much larger system, if at all.

1 Introductory remarks

The blockchain—a novel data structure and consensus mechanism designed for the decentralized currency Bitcoin—is currently being considered, exuberantly, for new applications in the financial and technology sectors. A steady stream of whitepapers from well-established banks (*JPMorgan Chase, Deutsche, Barclays, Citi*), professional service networks (*PwC, Deloitte, EY, KPMG*), and technology companies (*IBM, Microsoft*) explore how blockchains and distributed ledgers can create new digital assets, smart contracts, and provenance systems that enable direct interactions between participants, eschewing traditional intermediaries.

In this work, we do not propose a new system. Rather we systematized existing knowledge on blockchain technology and verifiable voting technology and showcase what the landscape looks like for their composition. It is not a survey, per se, because we are commenting on future work. We are informed by past work but not summarizing it. Our goal is to provide a justified research agenda by bringing into focus where there is potential, and pruning out what seem to be dead-ends.

1.1 An abbreviated history of the blockchain

In late 2008, Satoshi Nakamoto (the pseudonym of a not-yet-identified developer) posted a whitepaper to a cryptography mailing list outlining a new digital cash system called Bitcoin [21]. By this time, many attempts at commercializing some form of cryptographic digital cash had been made, none reaching commercial success. The design of Bitcoin is quite different in several regards from previous work, but one vital design decision that distinguishes it from nearly all previous proposals is having a publicly available ledger that records every transaction. A recent article shows how Bitcoin's blockchain is a novel combination of existing ideas in the academic literature [22]. In particular, it uses a time-stamping data-structure for appending a series of temporally-ordered messages to each other [15,2] and computational puzzles [10,1] to prevent Sybil attacks during a consensus protocol run between nodes on an open peer-to-peer network. The nodes validate transactions, agree on a valid ordering, and make it infeasible to modify past transactions without a computational majority [14] (a smaller-than-majority is sufficient to attack subtler properties [12]).

Many of the companies exploring new blockchain applications consider how trusted entities can be added into the system to avoid computational puzzles, shorten block update times from 10 minutes (in Bitcoin) to seconds, and govern network join/write/read privileges. We use the term distributed ledger to refer to the family of technologies that includes both Bitcoin's blockchain and these relaxed variants.[5]

1.2 Verifiable voting and digital cash are old siblings

Both digital currency systems and electronic voting systems have a long intertwined history in the cryptography literature. In fact, both originate with the same person: David Chaum. Chaum proposed the first cryptographic voting system in 1981, based on his mix network protocol [7]. The next year, he proposed the first cryptographic payment system based on his blind signature primitive [5]. Blind signatures were later studied extensively for voting (the hallmark paper being [13]), and cryptographic mixing has been used in digital cash, recently to anonymize Bitcoin transactions [4], [23]. The shared history does not end there, with many cryptographic primitives and protocols being utilized in both payments and voting: zero knowledge proofs, ring signatures, digital credentials, homomorphic encryption, algebraic MACs, and on and on. We can now add blockchains to the list.

A cryptographic or end-to-end verifiable (E2E) voting system is one that provides a proof to each voter that her ballot was included, unmodified, in the final tally (even in the face of a corrupt election authority and malicious vote capture equipment). This proof generally consists of an obfuscated form of the voter's ballot choices, called a receipt, as well as public data about all recorded

[5] Ironically, most distributed ledger technology strips away all of Nakamoto's novel insights, and could have been deployed using 1990s' state-of-the-art knowledge [22].

receipts and the tallying process that can be validated by anyone who wants (even those who did not vote). It is a strict condition of cryptographic voting systems that this proof leaks no information about how the voter voted, beyond what can be inferred from the tally alone, including the case where the voter deliberately crafts her ballot to leak how she voted. For in-person voting, we assume the voter's actions cannot be observed in the voting booth. For online or remote voting, no such assumption can be made and it becomes quite challenging (but theoretically possible [17]) to provide voter privacy when anyone can look over her shoulder and coerce/pay her to make certain choices.

2 A toy blockchain voting system

We now describe a simple voting system, layered on top of Bitcoin, to stack it up against the properties of a true E2E voting system. This is not a proposal for a system. Rather it is a pedagogic device we will tweak to showcase both the strengths and weaknesses of blockchains. It is not purely pedagogical however, as it is the core protocol of a number of blockchain voting projects including Swarm, Ballotchain, BitCongress, and Blockchain Voting Machine.

Voters register a Bitcoin address with the election authority (EA). The EA publishes a list of addresses but does not list which address belongs to which voter. Each candidate also publishes an address. Voters then cast a ballot by sending a small payment to their selected candidate. Any deviation from the voting rules (e.g., one vote per voter) can be seen by inspecting the blockchain, and the tally is visible by inspecting the candidate's received payments.

This scheme has one major benefit and several drawbacks. The main benefit is that when voters cast their ballots, they are doing so to a decentralized global network of computers where at least some will have no affiliation to the election authority and will include the transaction in their blocks.

The first drawback of the scheme is that the registration authority knows the mapping between voter identities and keys, and the mapping between keys and candidates voted for is public, so the registration authority can break ballot secrecy. Known cryptographic techniques (from pre-blockchain work) can address this through one of at least two ways. In the first method, the voter list consists of real voter identities and each voter adds a public key encryption of her address to the list beside her name. A set of mutually distrustful parties takes each encrypted address, with the real identity left behind, and shuffles it into a new location in the list and then 'obfuscates' the value so that it cannot be recognized but still decrypts to the same value. Once the list is shuffled by each party, the complete set is decrypted with a decryption key shared amongst them. This system is essentially what Chaum originally proposed in 1981! In his scheme, voters chose random signing keys that are shuffled and revealed in the first phase, and used to sign ballots in the second phase [7]. A modern variation could use ECDSA signatures and provably correct shuffling [16].

A fairly equivalent approach could have voters approach the registration authority in order to obtain a signature on their address. By using a blind signature scheme, the voter can obfuscate (blind) the address such that the authority just sees random-looking data when it signs, and then de-obfuscate (unblind) the message and signature in tandem such that the signature is properly formed relative to the original address. This is the basis of many voting systems [13], and is largely equivalent to the voter list approach above with two drawbacks: (1) once a signature is issued, it becomes invisible and cannot be easily revoked (if a voter becomes ineligible); and (2) no one can enumerate how many signatures an EA has handed out.

2.1 Tokens and coins

Because bitcoin payments can be traced on the blockchain, it is possible to flag a small amount of bitcoin to serve as a digital token for some other kind of asset. These tokens can be transacted between parties in the same way as Bitcoin itself (and possibly in new ways). Colored Coins, Counterparty and MasterCoin rely on this idea. While a ballot can be tokenized, the difference is largely semantic. In the toy scheme, anyone can send bitcoins to the candidate addresses and the system will have to use a list of authorized voters to filter out spam transactions from the real ones. Tokens do not solve this issue as the process for checking a token's validity is also consulting a list of coins that have been tokenized. In a closed distributed ledger, submitting a transaction could require authorization. This would eliminate spam but would reintroduce trust in the network for censorship-resistance and consensus.

2.2 A running tally

Even with one of the address anonymization add-ons, our toy blockchain voting scheme has the property that votes can be seen as they come in, enabling a visible running tally for each candidate. From a transparency perspective, this can be either a feature or a bug. Some blockchain voting projects (*e.g.,* Follow My Vote) advocate this as a beneficial feature. They argue that when combined with the ability to change one's vote (this could be accomplished in the system above by giving the voter many tokens with the policy that the 'last vote' counts), a new type of scoring protocol emerges where voters can cast a ballot for their most favoured candidate and then manually intervene if it seems they are too far from winning midway through the voting period. In response, we would argue that ranked choice voting is a more direct way of achieving this—one that removes the voter's guesswork and manual intervention. Further, open tallies are a 'bug' in one major sense: they are illegal in essentially all governmental elections today. Finally we know from paper audit trails (added to electronic voting machines) that writing votes into a ledger in the same order as they are cast can reveal how voters voted through simple timing analysis.

2.3 Hiding the tally until the end

These drawbacks motivates us to consider how the tally might be kept confidential until voting closes. Once again, the E2E literature offers numerous solutions to this problem and we port two approaches from this literature.

The first approach to hiding the vote itself is to have voters submit their votes as a message instead of moving tokens to a candidate's address. Bitcoin offers a transaction called OP_RETURN that allows a user to burn a small amount of Bitcoin to embed 80 bytes of data into the blockchain. The data could be a public key encryption of the vote with the EA's public key, to be tallied after the election in a privacy preserving manner (e.g., using additively homomorphic encryption [9] or with a verifiable mix-network [25]). Alternatively, the set of voters might engage in a pre-election protocol that produces for each voter a secret random number they can add to their vote to mask it, such that the sum of all mask values is zero [19]. When the last vote is submitted, the blinding factors all cancel out leaving just the sum of the votes, eliminating the need for an EA. In both cases, this is the core idea of a much larger protocol that needs additional steps and proofs to achieve all the desired properties of an E2E system. A system of the second type has been recently proposed and implemented [20].

The key question this raises is if such a system is really a 'blockchain voting' system any more given it is using the blockchain in a very limited fashion: for posting publicly viewable messages into a ledger. Toward an answer, consider a second approach that can leverage voting through financial transactions, as in our blockchain voting example, while preserving the privacy of the vote until after the election is complete. The main obstacle to accomplishing this is the known, one-to-one mapping between candidates and addresses. Instead, each ballot could have a unique set of one-time use addresses for each candidate. This is closely related to systems like Scantegrity [6] where each candidate on each ballot has a unique code associated with them: in our case, the code happens to be a Bitcoin address. A largely equivalent approach would be to distributed ballots that have the order of candidates shuffled, and have a set of addresses corresponding to the first candidate on the list, the second, etc. Such a system would correspond to Pret a Voter or vVote [24]. The backend of any of these systems would be able to verifiably map votes back to the correct candidates while preserving ballot secrecy.

Our key observation can now be made. There does not seem to be any difference between a voter sending a token to an address affiliated with their selection with a standard Bitcoin transaction, and simply writing the selection in an OP_RETURN message. The latter can do everything the former can (and more). Thus the ability to conduct transactions does not seem necessary for securing a voting system. In attempting to build a blockchain voting system, we have to layer so much additional cryptography on top that it effectively becomes a normal end-to-end voting system from the literature with the one modification that public messages are recorded on a blockchain. This begets an important follow-up question: where do E2E voting systems propose storing their public messages and is it any better than a blockchain?

2.4 The mythical bulletin board

Virtually every E2E voting system assumes the existence of a bulletin board which was first formalized by Benaloh and Tuinstra [3] and is considered to be an append-only, broadcast channel. The vast majority of papers adopt this as an assumption without specifying exactly how a bulletin board should be constructed. When E2E elections have been run in the real world (e.g., Scantegrity, vVote, Helios, etc), the bulletin board is typically implemented as a website with data signed by the election authority, perhaps with some replication by other interested parties.

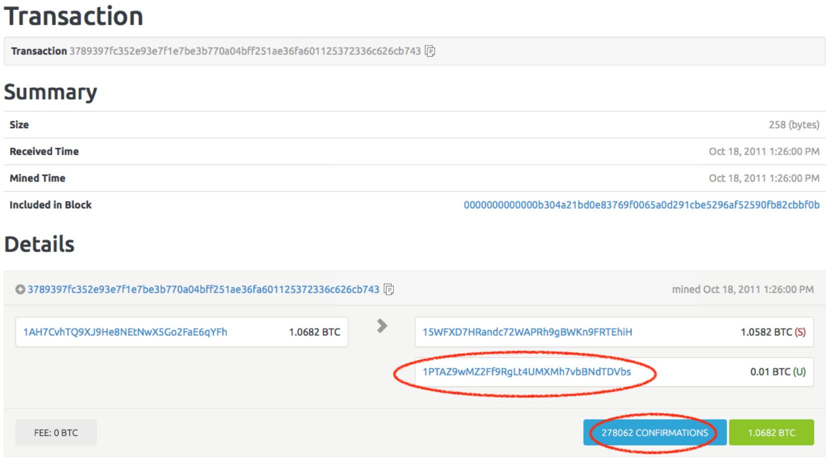
Such an implementation is neither strictly append-only nor a broadcast: the data can be modified or reordered at any time the EA choses, the EA can equivocate on what is stored on the bulletin board by sending different records to different voters, and the EA can play gatekeeper and drop messages it does not want published. Of course, the EA might get caught modifying the bulletin board, and could be held accountable because it signs every version, however this is a detection mechanism more than a prevention mechanism.

By contrast, embedding messages into a popular blockchain offers several improvements. First, there is no single entity to serve as gatekeeper and no successful cases of censorship have been observed in Bitcoin's history. Once a record is incorporated several blocks from the tail-end of the blockchain, it cannot be feasibly modified, and everyone holding the longest chain will see exactly the same records in this part of the chain (this is called a common prefix in the blockchain literature [14]). For these reasons, it is our opinion that a blockchain is the best bulletin board implementation we are aware of.

2.5 Carbon dating messages

Blockchains have one more trick that a traditional bulletin board cannot do. Because updating the blockchain requires computational work, a message that is X blocks from the tail-end of the blockchain was embedded X puzzles ago in time. Computing the solution to X puzzles takes a measurable amount of time. We might not be able to be very precise, since the computational power of the network varies, but we cannot shortcut months of the entire network's computational work in a few days. Thus a message embedded in a blockchain accumulates work over time, much like a fossil accumulates carbon, and this work cannot be faked even if the entire network is malicious.

This unusual property was observed by Clark and Essex [8], two contributors to the Scantegrity system. In Scantegrity, certain commitments are made prior to the election and if a corrupt election authority changed and backdated them after the election, without being noticed, the soundness of the tally is no longer guaranteed. Naturally voters may only become interested in verification after some unexpected election result, and such a voter has no way to know from a bulletin board whether the pre-election commitments were really made before the election or not. Thus, the researchers embedded the pre-election commitments for the 2011 municipal election in Takoma Park, MD (which was using



Transaction

Transaction 3789397fc352e93e7f1e7be3b770a04bff251ae36fa601125372336c626cb743

Summary

Size	258 (bytes)
Received Time	Oct 18, 2011 1:26:00 PM
Mined Time	Oct 18, 2011 1:26:00 PM
Included in Block	0000000000000b304a21bd0e83769f0065a0d291cbe5296af52590fb82cbbf0b

Details

3789397fc352e93e7f1e7be3b770a04bff251ae36fa601125372336c626cb743 mined Oct 18, 2011 1:26:00 PM

1AH7CvhTQ9XJ9He8NEtNwX5Go2FaE6qYFh 1.0682 BTC

15WFXD7HRandc72WAPRh9gBWKn9FRTEhiH 1.0582 BTC (S)

1PTAZ9wMZ2Ff9RgLt4UMXMh7vbBNdTDVbs 0.01 BTC (U)

FEE: 0 BTC

278062 CONFIRMATIONS

1.0682 BTC

Fig. 1. A screenshot from Blockexplorer showing the Scantegrity pre-election commitments embedded in Bitcoin's blockchain prior to the 2011 municipal election in Takoma Park, MD. At the time of writing, this message has been extended by 278062 blocks (each block requiring an average of 10 minutes of computation work from the Bitcoin network). The commitment value itself was embedded as a Bitcoin address since OP_RETURN was not supported at the time. The 0.01 BTC at this address is thus unspendable.

Scantegrity) into Bitcoin's blockchain so that voters after the election could be reasonably sure that the pre-election commitments must have really been made before the election, since by the time of the election, they had acquired a month of computation work (see Figure 1).

2.6 What would deployment look like?

If a blockchain were deployed as a bulletin board in an election, how would that look? For in-person voting systems, voters would use standard voting technology. E2E systems have been built on both optical scan and direct-recording electronic (DRE) architectures. In this case, the EA equipment (whether the devices the voter operates or some central tabulator) would post the events to the blockchain. The voter would retain some sort of privacy-preserving receipt (e.g., a confirmation code, ciphertext, marked ballot position, etc.) that they would use to cross-check with the data posted on the blockchain. Thus, their first and only interaction with the blockchain would come after the election for the purposes of an audit. This would require a computer program or website they trust. While the blockchain itself will not equivocate on what is stored, the voter's computational device can certainly display the wrong information (the voter cannot compute hashes in her head to detect something is amiss). The benefit of using Bitcoin's blockchain is the number of third party software and web-based tools for reading data from the blockchain.

In an online voting setting, voters cast their votes from their device which requires write access to the blockchain in addition to read access. The voter's device might not be able to obtain the correct software (or client-side scripts) due to a network attack and well-crafted malware can always interfere with a voter's ability to vote even if the network connections are secured with HTTPS. A malicious device can change a voter's selections, while simulating the process of casting the correct choices, and then simulate the receipt check should the voter perform one after the election from the same device.

The voting literature has a number of schemes, starting with Chaum's SureVote, where voters are mailed random-looking codes corresponding to each candidate to type into their devices so the device has no way of knowing how it might modify the vote. These solutions are largely tangential to the use of a blockchain but we make one remark. In an advanced code voting system like Remotegrity [26], the voter and EA exchange codes until the voter is satisfied and then sends a final lock-in code. A blockchain can provide assurances that this lock-in will be eventually incorporated if it is successfully broadcast.

2.7 Can a human use that?

End-to-end voting systems have evolved over time to systems where as much cryptographic detail is swept under the carpet as possible, especially during voter interactions. In any voting system that requires the voters themselves to authorize a blockchain transaction, the voter will need to safely and reliably store a cryptographic key pair. Key management is known to be difficult for voters whether for secure email or for Bitcoin itself [11]. It is generally not sufficient to use shorter passwords (cryptographically expanding them into keys) as the public keys will be placed on a public blockchain where they can be subject to offline brute-force attacks. In fact, many Bitcoin thefts have occurred because of short password-derived keys. Any system that authenticates voters using cryptographic keys needs to carefully consider usability factors.

A basic trade-off exists between a private ledger which might not achieve censorship resistance, and a public blockchain which generally requires fees. If, for example, a voting system were deployed on a public blockchain like Bitcoin or Ethereum, would each voter be responsible for converting some on their money into the underlying cryptocurrency to pay the transaction fees? The government could distribute small amounts of currency (say by postal mail with the registration card in the form of a QR code) however people may just collect these cards to use as free money. Note that private tokens do not resolve this issue—spending a token is still a transaction and a transaction requires a fee (*e.g.,* miner fee on Bitcoin, gas costs on Ethereum). The government could put up a server that voters authenticate to and the server pays the fees; this is natural but filtering access through a government-controlled server eliminates all the anti-censorship and reliability benefits of a decentralized system. Perhaps the best solution is give voters the option to use their own currency or to vote via a server.

There is also a question of scalability. If each vote corresponded to a transaction, cryptocurrencies would have be designed to handle the throughput of an election. As an upper bound, Bitcoin processes about 7 transactions a second which would take 225 days to process all the votes in the 2016 presidential election in the United States.

3 Voting through smart contracts

Blockchain projects like Ethereum enable verbose smart contracts to be expressed and enforced on a blockchain. This allows an election to add programmability to its eligibility requirements, ballots, or tallying function. This is where blockchain technology can move past being a better bulletin board to something truly novel.

Eligibility. A smart contract could define eligibility with an algorithmic description rather than a list of identifiers. It is hard to imagine all the possible forms this could take but it is one key area for future research to explore. This might be more readably applicable to private-sector elections than public sector ones. For example, assume ownership shares in a company are issued on a blockchain (as was recently approved by the United States Security Exchange Commission for Overstock) and a shareholder vote occurs. Shareholders could cast ballots that are weighted by how many shares they own (their stake) and they could even do so without revealing which shares belong to which shareholder. The outcome of a shareholder election might trigger certain outcomes on the same blockchain: financial transactions, stocks to split, new shares, or a dividend to be paid out. NASDAQ is experimenting with this type of system in Estonia.

Voters might post some money as a surety that is returned only if they behave correctly within the protocol. For example, for private elections on Ethereum, the proposal from McCorry *et al.* provides ballot and tally secrecy (until the last vote is cast) but requires each voter to submit their ballot [20]. Thus, to be eligible, voters must pay a small bounty that is refunded after casting their ballot. This provides an incentive to follow through. A voting system could also be designed to give one vote per unit of computational work performed. Some Bitcoin users generate for themselves vanity addresses where their otherwise random Bitcoin address contains a certain prefix of characters. They do this by generating new addresses as fast as they can until one happens to have the right prefix. An election could be held where votes are only counted from addresses containing a certain prefix (the longer the prefix, the exponentially more work it requires to generate). This could also be layered onto a ‘one vote per voter’ scheme as a spam deterrent. The idea of eligibility based on computation effort is actually used indirectly in Bitcoin today. The Bitcoin community sometimes holds polls on protocol changes and the network nodes that solve the blocks include their vote in their solved blocks, which skews the result toward the side doing the most work solving blocks.

Smart Ballots. Typically a ballot is filled out and cast. A ballot based on a smart contract could transition states. This could be a simple as allowing votes to be updated or delegated (the basis of liquid democracy).

Smart Tallies. Some countries, like Canada, have a per-vote subsidy where a set amount of money is paid to political parties for each vote they receive, this could be automatically redeemable after an election. In a participatory budget, money could be automatically allocated as a result of the tally. Bets on an election outcome could be automatically settled. While the outcome of an election can always be reported to a blockchain by a trusted party (called an oracle in the Bitcoin community), having an on-blockchain election can enable direct observation of the outcome by other scripts running on the blockchain. This blend of voting and payments could also be used for nefarious purposes, like automating vote buying (c.f., [18]).

4 Concluding remarks

The blockchain is undeniably a breakthrough for decentralized digital cash, and it may prove useful in disintermediating other protocols and procedures. However when it comes to voting, it seems the most useful contribution a blockchain can provide is a broadcast channel for cryptographic voting systems. In this role, blockchains are not only sufficient but actually provide some advanced properties, like carbon dating, that cannot be directly achieved with a standard bulletin board protocol. Blockchain-based voting systems also have the potential to entangle the eligibility for an election and the outcome of the election with other blockchain payments, assets and smart contracts in new and creative ways.

References

1. A. Back. Hashcash: a denial of service counter-measure, 2002.
2. J. Benaloh and M. de Mare. Efficient broadcast time-stamping, 1991.
3. J. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections. In *STOC*, 1994.
4. J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten. Mixcoin: Anonymity for bitcoin with accountable mixes. In *Financial Cryptography*, 2014.
5. D. Chaum. Blind signatures for untraceable payments. In *CRYPTO*, 1983.
6. D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. L. Rivest, P. Y. Ryan, E. Shen, and A. T. Sherman. Scantegrity ii: End-to-end verifiability for optical scan election systems using invisible ink confirmation codes. In *EVT*, 2008.
7. D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *CACM*, 24(2), 1981.
8. J. Clark and A. Essex. Commitcoin: Carbon dating commitments with bitcoin. In *Financial Cryptography*, 2012.
9. R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multi-authority election scheme. In *EUROCRYPT*, 1997.
10. C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In *CRYPTO*, 1992.
11. S. Eskandari, J. Clark, D. Barrera, and E. Stobert. A first look at the usability of bitcoin key management. In *USEC*, 2015.
12. I. Eyal and E. G. Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography*, 2014.

13. A. Fujioka, T. Okamoto, and K. Ohta. A practical secret voting scheme for large scale elections. In *AUSCRYPT*, 1992.
14. J. Garay, A. Kiayias, and N. Leonardos. The bitcoin backbone protocol: Analysis and applications. In *EUROCRYPT*, 2015.
15. S. Haber and W. S. Stornetta. How to time-stamp a digital document. In *CRYPTO*, 1990.
16. R. Haenni and O. Spycher. Secure internet voting on limited devices with anonymized dsa public keys. *EVT/WOTE*, 11, 2011.
17. A. Juels, D. Catalano, and M. Jakobsson. Coercion-resistant electronic elections. In *WPES*, 2005.
18. A. Juels, A. Kosba, and E. Shi. The ring of gyges: Investigating the future of criminal smart contracts. In *CCS*, 2016.
19. A. Kiayias and M. Yung. Self-tallying elections and perfect ballot secrecy. In *PKC*, 2002.
20. P. McCorry, S. F. Shahandashti, and F. Hao. A smart contract for boardroom voting with maximum voter privacy. *Financial Cryptography*, 2017.
21. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
22. A. Narayanan and J. Clark. Bitcoin's academic pedigree. *CACM*, 60(12), 2017.
23. T. Ruffing, P. Moreno-Sanchez, and A. Kate. Coinshuffle: Practical decentralized coin mixing for bitcoin. In *ESORICS*, 2014.
24. P. Y. Ryan, D. Bismark, J. Heather, S. Schneider, and Z. Xia. Prêt à voter: a voter-verifiable voting system. *IEEE TIFS*, 4(4), 2009.
25. K. Sako and J. Kilian. Receipt-free mix-type voting scheme. In *EUROCRYPT*, 1995.
26. F. Zagórski, R. T. Carback, D. Chaum, J. Clark, A. Essex, and P. L. Vora. Remotegrity: Design and use of an end-to-end verifiable remote voting system. In *ACNS*, 2013.