JOURNAL OF CYBERSECURITY

Journal of Cybersecurity, 2021, 1–15
doi: 10.1093/cybsec/tyaa025
Research Paper

Research Paper

Going from bad to worse: from Internet voting to blockchain voting

Sunoo Park , [1,2,*] Michael Specter, [3] Neha Narula [1] and Ronald L. Rivest [4]

[1]Digital Currency Initiative, MIT Media Lab; [2]Harvard Law School; [3]MIT CSAIL (Computer Science and Artificial Intelligence Laboratory) and [4]MIT IPRI (Internet Policy Research Initiative)

*Correspondence address. 77 Mass Ave, E14-15, Cambridge. E-mail: sunoo@csail.mit.edu

Received 22 July 2020; revised 6 November 2020; accepted 4 December 2020

Abstract

Voters are understandably concerned about election security. News reports of possible election interference by foreign powers, of unauthorized voting, of voter disenfranchisement, and of technological failures call into question the integrity of elections worldwide. This article examines the suggestions that "voting over the Internet" or "voting on the blockchain" would increase election security, and finds such claims to be wanting and misleading. While current election systems are far from perfect, Internet- and blockchain-based voting would greatly increase the risk of undetectable, nation-scale election failures. Online voting may seem appealing: voting from a computer or smartphone may seem convenient and accessible. However, studies have been inconclusive, showing that online voting may have little to no effect on turnout in practice, and it may even increase disenfranchisement. More importantly, given the current state of computer security, any turnout increase derived from Internet- or blockchain-based voting would come at the cost of losing meaningful assurance that votes have been counted as they were cast, and not undetectably altered or discarded. This state of affairs will continue as long as standard tactics such as malware, zero day, and denial-of-service attacks continue to be effective. This article analyzes and systematizes prior research on the security risks of online and electronic voting, and shows that not only do these risks persist in blockchain-based voting systems, but blockchains may introduce 'additional' problems for voting systems. Finally, we suggest questions for critically assessing security risks of new voting system proposals.

Key words: blockchain voting; election security; electronic voting

Introduction

Computers and the Internet have brought great benefits: improving efficiency, reliability, scalability, and convenience of many aspects of daily life. Some naturally ask, "why don't we vote online?" Voting online seems tantalizingly convenient: just a few taps on a phone from anywhere, without breaking your daily routine, taking off from work, or waiting in line. However, voting online has a fatal flaw.

Online voting systems are vulnerable to *serious failures*: attacks that are larger scale, harder to detect, and easier to execute than analogous attacks against paper-ballot-based voting systems. Furthermore, online voting systems will suffer from such vulnerabilities for the foreseeable future given the state of computer security and the high stakes in political elections.

While convenience and efficiency are essential properties of election systems, just as security is, these goals must be balanced and optimized together. An election system is ineffective if any one of these goals is compromised.

Exposing our election systems to such serious failures is too high a price to pay for the convenience of voting from our phones. What good is it to vote conveniently on your phone if you obtain little or no assurance that your vote will be counted correctly, or at all?

Those who favor increasing turnout, reducing fraud, or combating disenfranchisement should oppose online voting because the

Table 1. Four categories of voting systems

	In-person	Remote
Voter-verifiable paper ballots[3]	Precinct voting	Mail-in ballots
Unverifiable or electronic ballots	DRE[4] voting machines	Internet/mobile/blockchain voting

The top row is "software-independent" and far less vulnerable to serious failure than the bottom row. The bottom row is highly vulnerable and thus unsuitable for use in political elections, as explained further under "Vulnerabilities of electronic voting systems" below.

possibility for serious failure undermines these goals. Increased turnout only matters in a system that meaningfully assures that votes are counted as cast. The increased potential for large-scale, hard-to-detect attacks against online voting systems means increased potential for undetected fraud, coercion, and sophisticated vote tampering or vote suppression targeting specific voter groups.

What is more, online voting may not increase turnout. Studies on online voting's impact on voter turnout have ranged from finding no impact on turnout (e.g., Switzerland [1]) to finding that online voting slightly decreases turnout (e.g., Belgium [2]) to finding that online voting slightly increases turnout but is nonetheless "unlikely to solve the low turnout crisis" (e.g., Canada [3]).[1][4] Studies of Estonian elections have also suggested that turnout changes due to online voting may favor higher-income and higher-education demographics [5]. Recent US studies demonstrate significant demographic disparities in smartphone ownership (e.g., in gender, income, and education) [6].

Yet proposals for online voting have increased. These proposals are often misperceived as promoting the goals listed above: increasing turnout, reducing fraud, or combating disenfranchisement and coercion. Some online voting proposals have promised added security based on blockchain technology,[2] and have continued development and deployment despite vocal opposition by computer security and blockchain experts [7, 8] and technology reporters [9, 10].

A prominent example is the blockchain-based mobile voting app "Voatz," deployed in 2018 in West Virginia for overseas military voters in the US midterm elections [11, 12], and in several other US states for smaller-scale (municipal/county) elections [13, 14]. Recent research shows that Voatz suffers from serious security vulnerabilities enabling attackers to monitor votes being cast and to change or block ballots at large scale, unnoticed by voters and election officials [15].

A blockchain-based voting system was also used in Moscow, Russia, for its September 2019 city council elections [16]. Though some system code [17] was published and security researchers were invited to audit it [18, 19], the system was shown to be gravely vulnerable—not once, but twice (the second time after a proposed fix) [20]. Moscow responded constructively to the first reported vulnerability, but appears to have largely ignored the second. Japan and Switzerland have also conducted smaller blockchain voting experiments [21, 22].

The recent interest in online and blockchain voting proposals appears related to a growing political enthusiasm for improving and modernizing election systems—and for increasing their security from malicious interference (a topic of particular recent prominence in American politics). This is a promising trend, given that historically, many election authorities have been heavily constrained by limited funding for election equipment. We hope that this enthusiasm may lead to support and adoption of more secure, more transparent election equipment (addressing the many security flaws that have been documented in existing voting systems, as extensively documented for US voting equipment [23–25]).

However, the political expediency of adopting a "high-tech" solution also poses the risk that proposals may be too quickly pursued, before allocating sufficient time and funding for independent audits and feedback from security experts. New technologies should be approached with particular caution when a mistake could undermine the democratic process. After all, election systems have been designated as national critical infrastructure implicating a "vital national interest" [26].

The surprising power of paper

A natural but mistaken inclination is to entirely replace existing voting methods with the latest digital technologies. Some ask: "Why wait in polling place lines to cast votes on clunky old voting machines, when votes could be cast from voters' computers and phones over the Internet—using the same security protocols protecting online shopping, banking, cryptocurrency transactions?"

But, perhaps counterintuitively, getting rid of not only outdated voting equipment but also paper ballots risks "throwing the baby out with the bathwater" and making elections much less secure.

Security considerations for online shopping and online banking are different than those for election systems, in two key ways.

First, online shopping and banking systems have higher tolerance for failure—and they do fail. Credit card fraud happens, identity theft happens [27], and sensitive personal data are massively breached (e.g., the 2017 Equifax breach [28]). Online shopping and banking are designed to tolerate failure: merchants, banks, and insurers absorb the risk because doing so is in their economic interest.

Governments may also provide legal recourse for victims (as for the Equifax settlement [29]). But for elections, there can be no insurance or recourse against a failure of democracy: there is no means to "make voters whole again" after a compromised election.

Users of Bitcoin and other cryptocurrencies have lost hundreds of millions of dollars [30] due to theft, fraud, or mistake. Cryptocurrencies have fewer risk-absorption mechanisms than traditional banking; losses often fall directly on the victims, with no third party to provide relief.

The second key way in which the threat profile of online banking, shopping, and cryptocurrencies differs from that of elections is the skill level and aims of the adversary. Elections are high-value targets for sophisticated (nation-state) attackers, whose objective is not fraudulent financial transactions but changing or undermining confidence in election outcomes. A technically unsophisticated voter may be attacked by the world's most sophisticated adversaries.

1 See Stewart and Taylor [4] for a concise overview of relevant studies up to 2018, including additional references.

2 For example, Voatz, FollowMyVote, and Votem.

3 See "Minimal election security requirements" section for more discussion of voter verifiability and paper ballots.

4 "DRE" stands for "direct-recording electronic." This includes any machine that records votes only electronically (e.g., many touchscreen voting interfaces).

From a computer security perspective, securing an online voting system is a starkly different—and much harder—problem than securing online shopping or banking system.

Surprisingly, low-tech *paper ballots* may help protect against malfunctions or attacks of higher-tech voting system components (as discussed more under "Vulnerabilities of electronic voting systems" below).

Minimal election security requirements

Evidence-based elections

"The principle of 'evidence-based elections' is that... election officials should not only find the true winner(s) of an election, but... also provide the electorate convincing evidence that they did" [31, 32]. This compelling requirement implies both that the *election system* must be *auditable* (meaning it creates an evidence trail that can be checked to confirm that each relevant part of the system is functioning correctly as intended) and that *any given election* run using that system must be *audited* (meaning that that evidence trail is actually checked in that given instance).[5] *Auditability* alone isn't enough, and must be accompanied by *auditing* to be effective: auditability without auditing is like collecting receipts so you can check your credit card bill, then never checking the receipts against the bill. In short (paraphrasing [32]), auditability + auditing ⇒ evidence-based election.

Next, we highlight five minimal—necessary but insufficient—requirements for secure elections in an evidence-based framework: (i) ballot secrecy; (ii) software independence; (iii) voter-verifiable ballots; (iv) contestability; and (v) auditing.

The secret ballot. Ballot secrecy is essential to combat voter corruption and coercion. As the US Supreme Court has put it, "a widespread and time-tested consensus demonstrates that [ballot secrecy] is necessary in order to serve... compelling interests in preventing voter intimidation and election fraud" [33]. Protecting ballot secrecy provides a strong and simple protection against coercion and vote selling: if you cannot be sure how anyone else voted, this removes your incentive to pay them or threaten them to vote the way you'd like.[6] Indeed, election law scholars have noted that "[b]ribery of voters was far and away the greatest impediment to the integrity of elections before the introduction of the secret ballot, a fact well known not only to historians but to readers of great 19th century fiction"[7] [34].

Software independence. *Software independence* [35, 36] is the property that an undetected change or error in a system's software cannot cause an undetectable change in the election outcome. Software independence is a key property to ensure auditability of the casting, collecting, and tallying components of election systems. And even beyond ensuring that any errors that occur are detectable, software independence also reduces the likelihood of large-scale errors or attacks occurring in the first place: software-based systems are much more susceptible to "scalable" failure than non-software-based systems (as discussed more under "Vulnerabilities of electronic voting systems" below). For example, a remote programmer changing a line of code could in principle change millions of electronic ballots in milliseconds, whereas changing millions of paper ballots requires physical access and one-by-one handling.

Software independence does not require systems to not use software at all: rather, it means that the work of any software-based piece of the system (including auditing components) be checkable, in principle, using non-software-based means.[8] For example, a system for ballot casting, collection, and tallying would need to produce an evidence trail with an associated verification procedure to check that the system (i) recorded votes as intended, (ii) collected them as recorded, and (iii) counted them as collected, in any given execution. The basic definition of software independence leaves open *by whom* errors should be detectable: the appropriate answer to this question depends on the context, but using the tripartite framework just mentioned, individual voters should be able to detect errors in (i) and (ii), and anyone should be able to detect errors in (iii). Who can verify (i) and (ii) is constrained by the ballot secrecy requirement from above.

Without software independence, an undetected error in a piece of code could cause an undetected or unconfirmable error in the election outcome—and, as discussed under "Vulnerabilities of electronic voting systems" below, our state of the art is far from achieving error-free code. Democracy—and the consent of the governed—cannot be contingent on whether some uncheckable software correctly recorded voters' choices.

Voter-verifiable ballots. Even before ballot casting, a voter composing a ballot must be able to verify for herself that her prepared ballot reflects her intended choices. Paper ballots inherently enable simple verification that ballots are recorded as intended: a property that is challenging for electronic-ballot systems to achieve. "With a hand-marked paper ballot, the marks on the ballot necessarily reflect what the voter did, and we can have reasonable assurance that the human-readable mark on the ballot is for the candidate actually intended by the voter" [31]. A voter looking at their completed paper ballot can directly see whether their intended choices are marked (and, in principle, detect any mistakes they made).

Contestability. Software independence alone leaves another question unresolved: when an error is detected, can the one who detected it convince others that an error indeed occurred? Some types of errors may be publicly detectable, rendering the second question moot (since then anyone can run the verification procedure for themselves). However, certain verification procedures may be nonpublic: e.g., certain errors related to a given voter's vote might be detectable only by that specific voter. A *contestable* voting system is one that

5 The term "audit" in the elections context is often used to refer to "post-election audits," a particular type of auditing that checks some subset of the ballots after the main count is performed, in order to check that those ballots are consistent with the claimed outcome. Here, the term "audit" is used in its more general sense, encompassing checking or verifying the correct functioning of system components at any point during the election process.

6 After taking your payment, they would have no incentive to follow your instructions; and in the case of coercion, there would be no way to credibly follow up on the threat.

7 They reference Charles Dickens' *Bleak House*, George Eliot's *Felix Holt, Radical*, and Anthony Trollope's *Doctor Thorne*.

8 In practice, of course, the "practicality" of running the verification procedure with minimal or no dependence on software is an important issue—and one unaddressed by software independence, which just represents a minimum threshold of auditability. To facilitate auditing with little to no realistic reliance on software, it is strongly preferable to have systems that either have verification procedures simple enough for people to execute without using software at all "or" have open-source and publicly documented verification procedures, which voters and/or the public can perform using their own software.

provides publicly verifiable evidence that the election outcome is untrustworthy, whenever an error is detected [37].[9]

Auditing. As already mentioned, in addition to being *auditable* (which, for casting-and-tallying systems, corresponds to software independence and contestability), elections should be *audited*. Auditing checks that the evidence is trustworthy and, for casting-and-tallying systems, consistent with the announced election outcome. Both auditability and auditing are necessary for evidence-based elections. Such auditing should include compliance audits and risk-limiting audits [32]. Furthermore, "[t]he detection of any software misbehavior does not need to be perfect; it only needs to happen with sufficiently high probability" [35].

Election equipment may fail. The system must be designed not only to prevent failures, but also to ensure timely detection of failures when they occur: the public has a right to know about failures in the election process.

We refer the interested reader to Appel and Stark [31] for a more in-depth discussion of security requirements in evidence-based elections. This article's analysis focuses on the limitations of online and blockchain-based voting which means that they will not foreseeably be able to satisfy even these minimal requirements. Indeed, the interaction of these requirements is remarkably complex; it is surprisingly challenging to design systems that achieve even the minimal requirements all at once, and no currently known technology—including blockchain—is close to enabling mobile or Internet voting systems to simultaneously achieve of all these requirements.

Categories of voting systems

This article suggests four main categories of voting systems, determined by two key system attributes (also depicted in Table 1):

1. Are votes cast in person at a polling site, or remotely?
2. Does the system have voter-verifiable paper ballots or are ballots represented in a format that is not verifiable by voters?

Not every voting system that uses a phone, the Internet, or blockchain technology necessarily falls in the bottom-right category. For instance, an in-person paper-ballot-based voting system could use such technology as an auxiliary tool: e.g., allowing voters to use their phones to better understand the instructions or streamline creation of a paper ballot,[10][38] and/or saving a copy of the vote cast by paper ballot in an electronic format (perhaps on a blockchain). This article does not oppose the use of technology in the context of in-person voting systems with hand-marked paper ballots, and would support it in many contexts.

However, almost all proposals billed as "Internet voting," "mobile voting," or "blockchain voting" involve remote voting over the Internet with electronic-only recording of votes; such schemes all fall in the bottom-right category.

Accordingly, this article uses "Internet voting" and "blockchain voting" to refer to schemes in the bottom-right category only.[11] We consider "blockchain voting" a subcategory of "Internet voting," since all blockchain voting proposals transmit information over the Internet.

The top row and the left column of Table 1 are, respectively, strongly preferable to the bottom row and the right column in terms of security risk. We consider the top row suitable for political elections, with in-person voting preferable to mail-in voting wherever feasible (as indicated by their graduated green color). Importantly, top-row systems are software independent; bottom-row systems are not.

We consider the bottom row unsuitable for political elections for the foreseeable future, due to their lack of software independence and the greater risk of compromise compared to corresponding alternatives in the top row. The "Vulnerabilities of electronic voting systems" and "Blockchains as a ballot box" sections, below, explain this heightened risk.

The left column of Table 1 is preferable to the right column, because remote voting systems enable coercion and vote selling. Voters using remote voting system lack the seclusion provided by a physical polling place, so a coercer or vote buyer can look over the shoulder of a voter to confirm that they are voting as instructed (or paid) to.[12] In contrast, if voters are secluded at physical polling sites, coercers or vote buyers cannot know the vote really cast, rendering coercion and vote-buying ineffective.

A final note on our categorization: there may be proposals that appear not to fall squarely into our four-way categorization, by attempting to have electronic records that are somehow voter-verifiable.

A number of recent pieces of proposed legislation in the USA have recognized the need for paper-ballot-based voting systems (i.e., the top row of Table 1) and put forward the requirement of paper ballots (e.g., [39–41]). For example, the SAFE Act [39] requires durable paper ballots; that voters be able to inspect marked ballots before casting; that voters with disabilities have an equivalent opportunity to vote (including privacy and independence) to other voters; that voting technology be manufactured domestically; and other basic security requirements such as air-gapping.[13] However, such legislation is not necessarily likely to pass in the near future; in order to become law, it must also pass an eventual vote in the Senate.

Scope and terminology

This article uses "online voting" and "Internet voting" synonymously, in accordance with popular usage, to refer to any system where voters cast votes via the Internet—including blockchain-based and mobile voting systems. We use "electronic voting" to refer to any system where votes are cast purely electronically (i.e., the bottom row of Table 1).[14] Online voting is a kind of electronic voting. Much of our reasoning applies to all electronic voting, while some applies only to online or blockchain voting.

9 Of course, it would be even better if the correct outcome could be recovered in case of a dispute; detectability is just a minimum requirement. A system that is software independent and also enables this sort of recovery (based on the evidence trail produced during the election) is called "strongly software independent" [37].

10 For example, Los Angeles County has allowed voters to preload decisions on their phones and easily transfer the saved choices to ballots at the physical polling place [38].

11 We do not distinguish between "mobile voting" and "Internet voting" more generally; mobile voting transmits information over the Internet, and is a subcategory of Internet voting. We avoid the term "mobile voting" henceforth.

12 Mitigation proposals (such as allowing voters to submit multiple votes but only counting the last one) may help, but only if the adversary can't monitor the voter until polls close (e.g., because the polls close soon, or because they live together).

13 Air-gapping means maintaining a device disconnected from the Internet and from any internet-connected device.

14 This may include systems that use paper somewhere: e.g., if votes are cast and stored electronically, but a nonvoter-verifiable copy of each electronic vote is printed out during the process.

This article focuses on systems for casting and tallying votes (the focus of recent online and blockchain-based voting proposals). Internet- or blockchain-based technologies may help with other aspects of elections (e.g., auditing or voter registration), but that is not covered here.

Finally, this article focuses on the heightened security required, and particular threats faced, by political elections. Some elections, such as professional society elections, may have less stringent security requirements.[15] Whether electronic voting is suitable for such applications depends on the circumstances and is not covered here. "Election" should be read as "political election" henceforth.

Organization

Next, the "Vulnerabilities of electronic voting systems" section defines serious failures and explains how online voting systems are vulnerable to such failures. After that, the "Blockchains as a ballot box" section discusses blockchains and how they might be used in election systems, noting that blockchains do not mitigate any of the weaknesses inherent to online voting systems (discussed in the preceding section) and may sometimes introduce yet additional weaknesses. Then the "Critical questions" section provides a framework for election officials and citizens to critically evaluate voting technology proposals taking into account the state of the art in computer security. Finally, the "Related work" section discusses other related work, and the "Conclusion" section, naturally, concludes.

Vulnerabilities of electronic voting systems

This section argues that there is a class of security flaws that so gravely undermine election integrity—and thereby, democratic legitimacy—as to outweigh countervailing interests, and that electronic voting is more vulnerable to such failures than paper-based alternatives.

We call these *serious failures*: situations where election results have been changed (whether by simple error or adversarial attack) and the change may be undetectable, or even if detected, be irreparable without running a whole new election.

Merely the fact and public perception that the system is vulnerable to such failures may reduce an elected official's legitimacy and therefore destabilize a democracy. Vulnerability to serious failures thus undermines government legitimacy, whether or not the vulnerability was exploited by an attacker.

Even simple, well-understood tools like paper ballots are not totally immune to serious failures. For instance, if an election official may handle ballots in secret, they may undetectably destroy ballots cast against a particular candidate. If the malicious authority is crafty enough, and the margin of victory small enough, it can discard ballots such that the public may never know. This is why most election authorities employ transparency measures, such as allowing independent observers (including representatives from either party) to monitor and contest any part of the election process [42].[16][43, 44] Such monitoring enhances accountability in the presence of an auditable paper trail, but could be meaningless if key parts of the election process are shrouded in the internal operations of computers.

Unfortunately, independent observers and monitors have limited ability to prevent such failures: no group has infinite funds, time, and expertise. While acknowledging such limitations, we identify two categories of "showstopper" vulnerabilities that effectively eliminate election authorities' ability to prevent or remediate serious failures.

1. Scalable attacks: If the adversary's cost to tamper with the election is much less than the defender's cost to prevent such attacks, attempts to prevent, remediate, or even discover the failures may be impossible in practice. Scalable "wholesale" attacks affecting election outcomes are much more dangerous than "retail" attacks affecting only a few votes.
2. Undetectable attacks: If an attacker can alter the election outcome without any realistic risk of the modification being caught (by voters, election officials, or auditors), the attack becomes impossible to prevent or mitigate.

Systems attacks

Device exploitation is the act of adversarially modifying a computer's hardware, software, or equipment to allow attackers to gain access to information and/or otherwise change the system's operation. It is clear that exploitation can provide an adversary with an incredible amount of control over digital systems, allowing for scalable and undetectable attacks.

Once exploited, attackers have complete control over targeted voting systems and how they interact with the voter. Such malware may prevent casting votes (potentially stealthily, leading voters to believe they did cast votes), deceive voters about any aspect of the voting process, publicly expose voters' choices, or degrade the experience to deter voters from voting at all.

Exploitation is often imperceptible to users, and can often be done so undetectably that a forensic examination of the device will not reveal malware's presence. For example, ShadowWalker, a particularly advanced example, exists only in memory, and cannot be examined by the most privileged levels of the operating system [45]. Such malware is difficult to detect and, after the fact, may remove itself from the system without leaving a trace.

Worse, *any* communication between a system and the outside world may lead to exploitation: even when a device is not Internet-connected (i.e., is "airgapped"). Malware has been installed on airgapped devices, e.g., via USB and other removable media [46].

Systems attacks are incredibly scalable and cost-effective

Perhaps surprisingly, election-scale attacks may be inexpensive. In 2012, an unpatched "zero day" Android vulnerability cost roughly $60 000 [47]. To make a conservative estimate, we can assume that weaponizing, testing, and leveraging the exploit and associated malware might increase the cost by two orders of magnitude to $6 000 000. While this might seem like a hefty sum, the total campaign expenditure for one candidate in the 2016 US Presidential election was roughly $768 million [48]. Compared to the research and development budget of a nation state's intelligence apparatus, such exploit costs would be negligible.

Once prepared, a vulnerability may be used many times, and a single use could affect many votes. Attacking centralized services like voting machine manufacturers, or voter registries (as in the 2016 US Presidential election [49]) may provide a cost-effective way of affecting many votes via few compromised machines, potentially enabling quietly alteration of an election outcome.

15 Also, blockchain protocols and smart contracts may employ "voting" as part of their consensus protocol: such protocols are not designed for, and do not meet the security requirements of, political elections, and they are not covered here.

16 Specific examples include Ref. [43, 44].

Devices are vulnerable, and digital-only defenses are lacking

Device security relies on many different organizations. Voting system flaws might be introduced by the voting software vendor, the hardware vendor, the manufacturer, or any third party that maintains or supplies code for these organizations. A voter using a phone to vote depends not only on the phone vendor, but on the hardware companies providing drivers for the device, the baseband processor, the authors of third-party code in the voting software, the manufacturer of the physical device, and the network or any other systems that the device relies upon to cast the vote. This also raises geopolitical concerns: where are devices manufactured, and who controls the voter's network?

Cryptography does not prevent most systems bugs from being exploitable. Conversely, systems flaws may enable breaking cryptographic guarantees. Writing software to implement cryptographic primitives and protocols is difficult and subtle [50], and there are numerous examples of systems flaws leading to the compromise of cryptographic protocols or primitives [51, 52].

Attacks on systems used in practice

Researchers have repeatedly shown that polling-place electronic-only voting devices are vulnerable, even without direct connection to the Internet. For example, a 2006 paper demonstrated that the voting system used by much of Maryland and Georgia was insecure and easily exploited [53], and more recent analyses have shown that such systems have not improved [24].

Internet-connected electronic voting has also been attempted and shown to be equally vulnerable. Analyses have been performed on Internet voting systems in Estonia [54]; Washington, DC [55]; and Switzerland [56], all of which were found to be vulnerable to serious failures.

Alarmingly, there is significant evidence that election systems have been targeted by foreign adversaries. For example, the Russian government has infiltrated voter registration databases related to Florida and Illinois [49], and there are indications of similar issues in Georgia [57].

Mail-in ballots

When a voter cannot otherwise access the polls, election authorities may provide a remote voting solution, e.g., mail-in ballots for overseas military and other absentee voters.

However, the risks discussed in this section strongly favor in such cases (i) limiting remote voting to the settings where there is no feasible alternative and (ii) using mail-in ballots rather than online voting. While mail-in ballots enable vote selling and coercion, they are still far less susceptible to large-scale covert attacks than online voting.

Destroying a mail-in ballot generally requires physical access, and large-scale efforts must target ballots across post offices that are geographically and operationally diverse—a very different task from exploiting a single vulnerability that could stealthily affect millions of devices with practically the same effort as one device. As a result, attacks against mail-in ballots are less likely to be scalable or to go undetected than attacks against purely electronic systems.

See also Federal Voting Assistance Program [58] for more information on the US legal regime governing absentee ballots, including paper ballot requirements.

End-to-end verifiable voting

Some promising recent proposals called *end-to-end verifiable* (E2E-V) voting systems [59–62] use cryptographic techniques and post encrypted ballots on a public bulletin board[17] such that voters can verify whether their vote was included in the final tally. End-to-end verifiability can be a desirable feature to add to either paper-ballot-based or electronic-only voting systems, but does not resolve the major problems described in this section. (Paper seems likely a key component in providing receipts in a practical E2E-V system.) Thus, any system that is electronic only, even if E2E-V, seems unsuitable for political elections in the foreseeable future. The US Vote Foundation has noted the promise of E2E-V methods for improving online voting security, but has issued a detailed report recommending avoiding their use for online voting unless and until the technology is far more mature and fully tested in poll site voting [63].

Others have proposed extensions of these ideas. For example, the proposal of Juels *et al.* [64] emphasizes the use of cryptography to provide a number of forms of "coercion resistance." The Civitas proposal of Clarkson *et al.* [65] implements additional mechanisms for coercion resistance, which Iovino *et al.* [66] further incorporate and elaborate into their Selene system. From our perspective, these proposals are innovative but unrealistic: they are quite complex, and most seriously, their security relies upon voters' devices being uncompromised and functioning as intended, an unrealistic assumption. They also involve a complex registration phase where voters may need to visit one or more registrars to securely register their cryptographic keys. Because they rely on the assumption of uncompromised personal devices, they can afford to make no essential use of paper.

Importance of transparency

Software is complicated; it is very hard to get it right, and software bugs are commonplace. Moreover, if the software implements security mechanisms, it should not only be correct but provide credible assurance of secure operation to those who depend on it. Not only is the design challenging to get right, but the implementation can be particularly challenging to get right if the adversary may corrupt insiders (such as software developers) in the supply chain.

Today, it is best practice, including among cryptocurrency implementations, to adopt open-source development methods.[18][67] Disclosed-source implementations allow one to gain substantial (though not complete) confidence that the implementation contains no serious bugs or security holes.

Disclosing security-critical system designs for inspection by experts and even "the enemy" has been considered good security practice since the 19th century (Kerckhoffs's Principle [68]). While intuition suggests that a secret system design is harder for an adversary to figure out, the lack of scrutiny makes it easier for security vulnerabilities to remain unnoticed and unaddressed. Moreover, keeping a system design secret is infeasible for systems in widespread use—underscoring the importance of security guarantees that hold even if the design is disclosed. Thus, security-critical software that is

17 "Blockchains as a ballot box" section discusses how blockchains could be used to implement a public bulletin board. However, we argue that blockchain technology does not add anything "beyond" a way of implementing a public bulletin board, and as such, does not help solve existing issues that E2E-V voting systems share with online voting systems.

18 Here "open-source" means "disclosed-source," where the source code is open for all to read but changes may be controlled. Wallach [67] gives a detailed discussion of open/disclosed source in voting systems.

closed-source carries much higher risk and uncertainty than disclosed-source alternatives. Accordingly, voting systems should favor disclosing system designs and code whenever possible.

That said, transparency is not a panacea. One cannot generally verify that the code running on a given machine is actually the compiled version of the open-source software that was reviewed; devising such verification methods is difficult and an area of ongoing research.[19][69] While transparency (disclosed software and good cryptographic protocol documentation) seems necessary for security, it is by no means sufficient.

Blockchains as a ballot box

Some recent proposals claim using blockchain technology adds security to electronic voting [70–72]. We show that blockchains do not address the issues discussed in the "Vulnerabilities of electronic voting systems" section and might introduce new problems.

We begin by reviewing blockchain technology ("Blockchain technology overview" and "How to achieve a blockchain interface" below). Those familiar with blockchain technology may skim or skip these subsections. Then the third subsection, "Blockchain technology applied to voting", re-emphasizes and gives examples illustrating that blockchain voting is still online voting, and thus suffers the same vulnerability to serious failures described in the "Vulnerabilities of electronic voting systems" section above. The fourth subsection, "New problems blockchains introduce", discusses how blockchain-based electronic voting could create additional problems for election systems. Finally, the fifth subsection, "Voting within blockchains", describes voting used "within" blockchain technology, which we distinguish from voting in political elections.

Blockchain technology overview

The term *blockchain* is used, confusingly, to refer to a wide range of technologies, including distributed databases, hashing, digital signatures, and sometimes even multiparty computation and zero-knowledge proofs. All of these technologies individually predate the use of blockchain technology by Bitcoin [73].

A blockchain implements what cryptographers call a public bulletin board: a linear ordering of data with the following properties. It is *append-only*: data can only be added to the end of the board, and never removed; and it is *public* and *available*: everyone can read the data on the board, and every reader sees a common prefix of the same ordering.

For example, Bitcoin's blockchain is a list of transactions. Users can add transactions to the end of the blockchain, and read the transaction list to learn who owns which bitcoins.

Blockchains have validation rules: by consensus, only data with a certain format may be appended. For example, cryptocurrency transactions transferring money must pass certain validity checks or they will not be appended: the sender must have sufficient funds, and the transaction must demonstrate the sender's authorization to move the funds.

Security is guaranteed only under certain assumptions. In Bitcoin, security only holds if a majority of the mining hash power is "honest" (i.e., adheres to the Bitcoin protocol). In other blockchains, the required assumption might be that at least two-thirds of the participants are honest. If such assumptions are violated, the blockchain might lose its availability, linear ordering, and common prefix guarantees.

Unlike Bitcoin, E2E voting protocols (see "End-to-end verifiable voting" above) generally require an *authenticated* bulletin board or one where some voting authority signs the contents to indicate that this is the agreed-upon board for the election.

How to achieve a blockchain interface

To achieve the public bulletin board functionality, blockchains typically operate as follows. A network of computers runs a common (public) piece of software to agree on an ordered log of data. Users submit new data with digital signatures, and the software enforces validation rules: e.g., users cannot create new coins outside the specified monetary policy. The software also runs a consensus protocol to agree on the continuing log of data and links the data together using hashes to prevent undetected tampering with past data.

Consensus

Distributed consensus is the problem of many computers agreeing on a single value in the presence of failures. Before Bitcoin, designers of consensus protocols assumed that the set of participants was known, and relied on sending messages to everyone. The core innovation behind Bitcoin is a *permissionless* distributed consensus protocol whose security is incentive-based, known as Nakamoto consensus [74]. Bitcoin uses a technique called *proof-of-work* [75, 76] to select the next block in the blockchain; in Bitcoin, the "work" is producing a preimage of a partially-fixed hash. Participants who do this work are known as *miners*. The first miner to find a preimage broadcasts their block to the Bitcoin network and, once the block is accepted, is paid in Bitcoin specified in the block they produced; this is called the block reward. The block reward consists of both newly minted Bitcoin and the transaction fees of the transactions included in the block.

Miners must expend a lot of computational cycles to find this preimage; this makes proof-of-work energy intensive and its cost dominated by operational costs. Because of this, most miners have gravitated to geographical locations with cheap energy, and many large miners are based in China. The security guarantees of Nakamoto consensus hold only if the majority of the mining power behave honestly (i.e., follow the protocol).

Some cryptocurrencies implement a newer type of consensus protocol called *proof-of-stake*, which is much less energy intensive. These protocols are more like traditional consensus protocols except the set of participants is determined by who holds stake, or coins, in the system. The security guarantees of these protocols hold only if a certain fraction of stakeholders (i.e., coin owners) behave honestly.

The advent of permissionless protocols has caused many to take a second look at distributed databases, where different database nodes are run by different organizations. These types of databases are sometimes called *permissioned blockchains* because, similarly to permissionless blockchains, they are a verifiable log of records; but they differ in that the participant set is limited and determined ahead of time (nodes need permission to join the system). These protocols improve fault tolerance, and can even tolerate some fraction of malicious nodes (typically up to a third).

Distributed database technology can improve databases' resilience to computer failures; however, we shall see that this does not

19 For example, Fink *et al.* [69] study the potential use of "trusted platform modules" (TPMs) to mitigate concerns that the software running is not the software that is supposed to be running. Of course, one still has the concern that the TPM system itself is free from bugs, and in any case this doesn't address the correctness of the voting system software.

address the core problems with electronic voting, discussed in the "Vulnerabilities of electronic voting systems" section.

Authentication

Users create a digital signature to authorize a transaction to be added to the blockchain, perhaps spending coins. There is no "user identity" in the system beyond the signing key itself, and a user may have many unrelated signing keys. Nodes in the network validate signatures and check that each batch of transactions maintains financial invariants, e.g., the spender must have sufficient funds to spend, and/or coins are created following an agreed-upon schedule. In a blockchain without an associated coin, nodes might validate other application-specific rules.

Smart contracts

Blockchains may support operations more complex than just transferring coins: e.g., coins may be transferred conditionally, using scripts or smart contracts. For example, in Bitcoin, coins can be locked up for a period of time or require multiple signatures to spend. Blockchains like Ethereum support even richer smart contracts: the Ethereum network functions like a single, global computer running different smart contract programs; these include applications like prediction markets, games, and marketplaces.

Transaction secrecy

By default, blockchains do not keep transaction details secret: all Bitcoin transactions are public. A key feature of blockchain technology is that transactions are verifiable, and public verifiability seems at odds with secrecy. In permissioned blockchains, the participants running the blockchain can restrict read access to the blockchain. This can be helpful to limit data leakage, but it comes with a price: those without access cannot download and verify the blockchain. In a permissionless blockchain (like Bitcoin), the participant set is unrestricted, so the entire transaction history is public. Some cryptocurrencies use *zero-knowledge proofs* to hide transaction details (the participants in the transaction and the amount) while still maintaining public verifiability. A zero-knowledge proof shows credibly that some statement is true without revealing why that statement is true. For example, using a zero-knowledge proof, I could convince you that I know the solution to a specific Sudoku puzzle without revealing the actual answer. Zero-knowledge proofs were invented many decades before blockchain technology [77] and may be useful for electronic voting systems (especially E2E-V systems) though they are not enough alone.

Applications

Blockchains have application beyond cryptocurrencies. For example, IBM uses the Hyperledger Fabric blockchain to record the provenance of food traveling through a supply chain [78]. Participants include producers, suppliers, manufacturers, and retailers and the goal is to "provide authorized users with immediate access to actionable food supply chain data, from farm to store and ultimately the consumer." Everledger is a company aiming to track diamonds using blockchain technology, whose goal is to "create a secure and permanent digital record of an asset's origin, characteristics, and ownership [79] ." Note that these applications require entities to make in-blockchain claims about assets and operations in the real world.

Blockchain technology applied to voting

Bitcoin, the best-known (but not first [80]) example of blockchain technology, operates in an adversarial environment: anyone can download the software and join the network, including attackers. The idea behind Bitcoin is that participants sign transactions to indicate authorization to transfer, and are constantly downloading and validating the blocks to check that rules are being followed and their coins are valid. Blockchains use consensus protocols to avoid a single point of failure; these protocols can tolerate a small number of participants acting maliciously.

These ideas seem as though they might be helpful for electronic voting: e.g., using cryptographic signatures to make forging votes difficult, and using hashing and distributed consensus to maintain a ledger of votes that attackers cannot tamper with unless they co-opt much of the network. However, it is extremely challenging to make these techniques work reliably in practice, and a blockchain's properties only hold if the assumptions underlying a blockchain are not violated; e.g., a sufficient portion of the participants are honest and users can get their transactions accepted onto the blockchain.

Even if this is the case, blockchains alone are insufficient for a secure electronic voting system. At most a blockchain could serve as the public bulletin board in a greater electronic voting protocol; one would still need to devise ways to achieve the voting requirements in "Minimal election security requirements" section, such as a secret ballot, voter-verifiability, and contestability. There are simpler, more reliable ways to achieve a public bulletin board, such as traditional database technology.

Importantly, blockchains alone do not address the problems described in "Vulnerabilities of electronic voting systems" section. In particular, users must still use potentially vulnerable devices and network infrastructure in a blockchain-based electronic voting system, meaning such a system is still vulnerable to serious failures. The cryptographic and consensus guarantees of blockchains do not prevent potential serious failures.

Next, we sketch a possible blockchain-based voting system and discuss how it fails to address several security issues. This design does not consider every detail of implementing a voting system on a blockchain and is not exhaustive, but it demonstrates issues that would apply to many designs.

Coins as votes

Here is a straw man proposal for "blockchain as a ballot box," which underscores several of the points made in this section. The voting authority, which maintains a voter registry, has each registered user create a public/private key pair, and each user sends their public key to the registry. Then, the voter registry spends one coin to each public key. To vote, each user spends their coin to the candidate of their choice. After a period, everyone can look at the blockchain, total up each candidate's coins, and select the one with the most coins as the winner.

This straw man design has several problems illustrative of the problems faced by blockchain-based voting proposals more broadly. First of all, it does not provide a secret ballot: all votes are public, and users can prove to a third party how they voted, enabling coercion and vote-selling.

Second, this design relies on users being able to get their votes on the blockchain in the given election time period. The vote tallier cannot wait for all users to spend their coins because that means a single user could prevent the election from finishing; there must be some cutoff point. An adversary able to influence network connectivity or to conduct a denial-of-service attack could keep users from voting

until after the cutoff. Public blockchains, in particular, are limited in throughput and require fees to submit transactions.

During times of high transaction rates, fees can get quite high, and transactions can be delayed. An attacker willing to spend enough money could flood the blockchain with transactions to drive up fees and keep users from voting until after the cutoff point has passed.

Third, the design only works if the blockchain properly implements the public bulletin board interface. If the blockchain is compromised—e.g., if a majority of the miners or validators collude—then they could create multiple versions of the blockchain to show different people, sowing discord. Or, they could censor certain users' votes. Several cryptocurrencies have suffered these types of attacks, where their blockchains have been rewritten [81–83]. Blockchains are often referred to as "immutable," but these attacks show that this is not always true in practice, especially for smaller blockchains.

Fourth, security of this straw man hinges on key management. If a user loses their private key, they can no longer vote, and if an attacker obtains a user's private key they can now undetectably vote as that user. Many users have lost access to their private keys and thus have lost their cryptocurrency. This has even happened to cryptocurrency exchanges, which have lost hundreds of millions of dollars worth of cryptocurrency to attackers or through bad key management [84, 85]. Blockchains cannot help if a user's keys are compromised; in fact, blockchain-based systems seem to require using public key cryptography. This blockchain-based electronic voting system would also need to maintain and run a secure public key directory.

Finally, all of the above depend on secure software and hardware, as blockchains alone do not provide software independence for vote casting. If a user's voting device (probably a mobile phone) is compromised, so could be their vote.

Permissioned blockchains

One might think of using a permissioned blockchain, instead, at least to address the first and second issues. However, a permissioned blockchain system would still suffer from the remaining issues, and, depending on how it is implemented, new ones: if users cannot read the permissioned blockchain and verify that their votes were counted, it does not implement a verifiable tally. (If everyone could read the blockchain, then they could prove how they voted by pointing it out and it would not be a secret ballot.) In permissioned systems, there are even fewer, more homogenous servers to compromise compared to large public blockchain instances. This enhances the possibility that they could all be compromised, especially if they run on the same operating system or run the same software. Permissioned blockchains also do not address the issues of key management or the security of software and hardware on user devices.

Zero-knowledge proofs for secret ballots

Some cryptocurrency schemes keep transaction contents secret while still allowing public verification of certain financial invariants, getting around the tension (described above) between secrecy and public verifiability. These schemes use the zero-knowledge proofs mentioned in the "How to achieve a blockchain interface" section above. For example, Zerocash [86] and its subsequent implementation in the cryptocurrency Zcash [87] provide "shielded" transactions, which do not reveal amounts, senders, or receivers. Despite this, these transactions' financial invariants are still publicly verifiable, much like public blockchain transactions.

One could use these techniques to modify the straw man to support shielded transactions. While this would mean that transaction data would no longer be publicly visible, the resulting scheme would still be far from providing ballot secrecy.[20][88]

First, a digital-only solution does nothing to prevent physical monitoring by coercers or vote buyers. Second, zero-knowledge proofs are designed for a setting where the party with secret information wants to keep it secret (that is why they are using zero-knowledge proofs)—they generally do not prevent that party from revealing information voluntarily.

Importantly, elections have much higher stakes than cryptocurrency. An attack on many cryptocurrency users would cause monetary loss, an attack on many voters can cause government change.

New problems blockchains introduce

Besides all the usual security issues associated with online voting, a blockchain-based voting system introduces new security concerns. Blockchains are designed to be decentralized, run by multiple actors. This means blockchain protocols require governance and coordination, which can inherently be difficult to manage (as exemplified in Buterin [89] and Harper [90]). Importantly, blockchain technology introduces more "complexity" into software and its management.

This additional complexity introduces problems with fixing bugs and deploying new software. Because decentralized systems do not have a single point of control, changing the protocol, even for the sake of addressing vulnerabilities, requires coordination. This means it takes more time and effort to deploy security fixes in a decentralized system than in a centralized one, and blockchain systems can be vulnerable for longer periods of time than centralized counterparts. For example, one Bitcoin software tracker indicates that in 2020, 27% of the Bitcoin network is still vulnerable to CVE-2018-17145, a resource consumption denial-of-service vulnerability disclosed in 2018 [91, 92]. In a critical application like voting, the ability to move quickly to unilaterally fix bugs may be essential.

Other work has proposed frameworks for determining when an application is a good fit for blockchain technology [93–95]. Though voting requires auditing, it does not warrant the complexity introduced by a technology like blockchain that requires shared governance and shared operation. Elections are inherently centralized (with a central organization, the government, that is in charge of election procedures, the contests of the election, the eligibility of the candidates, and eligibility to vote).

Despite Bitcoin launching in 2009, it took several years to gain users and for developers to gain experience securing the platform. The technology is still new and under development.

Another independent concern with using some blockchains for voting is the inadvisability of using new distributed consensus protocols or new cryptographic primitives for critical infrastructure until they have been well-tested in industry for many years. Distributed consensus protocols and cryptographic systems are difficult to implement correctly [96, 97]. Zcash and Monero both use relatively novel cryptographic primitives for privacy, and both have suffered

20 Furthermore, adding zero-knowledge proofs would bring new issues related to the complexity and recency of the technology, which is still in early stages. New bugs are being discovered: e.g., in 2018, a critical bug in Zcash was discovered that allowed undetectably counterfeiting Zcash coins [88]. Moreover, the additional complexity may render the voting system (yet more) opaque to the general public, whereas it is important for democracy for the public to believe in the correctness of election technology—and thus, election results.

from critical bugs [88, 98]. Though this applies to all new protocols, blockchains contribute to this problem.

Voting within blockchains

Blockchain protocols and smart contracts sometimes employ voting within the blockchain or contract application. This kind of "voting" is very different from the main topic of this article, namely, voting in a political election. For example, in EOS, token holders can vote for validators to participate in the consensus network protocol and select blocks. It is important to disambiguate the use of the term "voting" here: voting in a political election has different aims and security requirements from voting in a consensus protocol. A maliciously elected EOS validator could slow down validation or validate incorrect blocks, potentially affecting holders of the EOS cryptocurrency. Malicious validators in political elections could do much worse.

Some smart contracts let token holders "vote" on contract outcomes. For example, Augur is a protocol for creating prediction markets that run on Ethereum where users can bet on the outcomes of sporting events, market movements, weather, and more [99]. Augur has a built-in token called REP. REP token holders stake their tokens to vote on real-world outcomes and report them into the smart contract. REP holders are responsible for participating in contract disputes and will be penalized (they will automatically lose some of their REP) if they do not participate. These kinds of processes do not fulfill the requirements for secure voting in elections and are designed for a very different purpose with a different threat model.

Summary

A bulletin-board-like interface combined with encryption for secrecy may be helpful for voting, but these techniques still do not address several fundamental security issues with electronic voting. It remains unclear what type of role decentralization should play; on the one hand, systems with a small number of homogenous nodes might be more likely to suffer from compromise. On the other hand, elections are inherently centralized, and decentralized systems come with many drawbacks, including potential congestion and difficulty in upgrading.

Critical questions

This section provides a list of worthwhile questions that should be asked about any future online or blockchain-based election system proposal in order to better understand its security implications, before considering its adoption for high-stakes elections. Much of this list is inspired by previous examples of failures in Internet voting schemes [15, 20, 54, 56, 100], questions asked by experts involving past blockchain-based systems [7], as well as the survey of open problems E2E-V systems by Bernhard *et al.* [101].

This list is not intended to be comprehensive, as a short article like this cannot provide a complete guide to all of the issues that might be raised about "voting on the blockchain," or electronic-only voting as a whole. First, the questions raised here relate to voting system *security*, rather than other important aspects of voting systems (e.g., usability, cost, accessibility, etc.). Second, security cannot be achieved simply by "passing a checklist"—even given good answers to all of the questions here, a system could still be insecure. However, a good set of questions illuminates gaps in reasoning, poor assumptions, and implementation problems. We believe that satisfactory answers to these questions are a worthwhile demand: a valuable starting point to evaluate voting system proposals, and a basic level of transparency to which the public is entitled.

Stakeholders and adversaries

Elections often have a number of unique stakeholders and adversaries, which may or may not be disjoint sets. Formally delineating the roles, responsibilities, and powers of these individuals is a necessary first step in analysis. Groups involved might include the public (including observers who might not be voters), candidates, voters, election officials, poll workers, auditors (including foreign observers), system designers, and vendors. Once the list of groups is well understood, one may then ask questions like:

- What guarantees does the system provide if any of these actors are malicious?
- Can any one of these entities or any combination of the above unduly control the outcome of the election?
- What are the capabilities of each of these stakeholders at each point of the election process?
- What can be done to ensure that these actors behave well? Are any here capable of holding others in the list accountable?

Security objectives and threat model

After the actors in the election are defined, one can then begin designing a *threat model* for the election. It is worth noting that election systems often differ in requirements; style of administration, how important the election is (and, therefore, what sorts of adversaries are expected), and the powers of the participants all shape the kinds of objectives an election system might have.

- What security properties is the system intended to have?
- What is the threat model? For political elections the threats considered should include at least:
  - Compromise of a device's hardware and/or software, possibly via supply chain attacks
  - Failure to properly record a voter's choices
  - Tabulation errors
  - Selling of votes
  - Corruption of evidence trail
  - Ballot "stuffing" (extra ballots) or ballot destruction
- What kinds of plausible attacks are not considered in the system design? (E.g., does the security of the system depend on "trusted hardware" or "trusted software"? Are these assumptions valid?)
- How many people would an adversary have to corrupt in order to steal an election?

Security mechanism design

Security mechanisms define how one might achieve the protections delineated in the security model. Security mechanism design questions help us understand how the system works to fulfill the objectives given the threat model described.

- What security mechanisms are proposed in the system design?
- Are those mechanisms designed to prevent security violations, or to just detect such violations?
- Does the system provide coercion resistance, receipt freeness, and a secret ballot? How?
- What happens when a security violation is detected? Does the system handle dispute resolution well?

- Do the proposed mechanisms rely on particular behaviors by certain parties (voters, election officials, etc.) to be effective?
- If some, most, or all voting system computers or devices are compromised, what is the worst case effect it could have on the reported election outcome?
  - Would that effect be reliably detectable? How?
- What mechanisms enable voters and observers to verify that the system works as it is intended to, and that the outcomes produced have not been affected by adversarial behavior?

Evidence-based elections

A key goal of an election is to prove to the losing party that they did, in fact, lose. An election system must therefore provide convincing evidence to all parties that the election result is correct, even in the face of intense scrutiny.

- What evidence does the system produce supporting the reported outcome?
- Why should that evidence be considered trustworthy? Does the validity of the evidence rely on any assumptions about the correct operation of the system? Does concluding that the evidence is trustworthy require trusting that one or more computer systems are operating correctly? If so, are those assumptions credible and/or verifiable?
- Is that evidence auditable? What forms of audits are supported? What assurance do they provide, to whom?

Vetting and transparency

Providing proper evidence revolves, in part, on the public's ability to perform their own analysis of the system, and to help find and fix potential flaws. Such analysis not only provides better evidence that the system is safe, but also crucial in providing legitimacy and public trust.

- Who can verify the system's design and operation? Neutral third parties? In the USA, the federal certification process? Has anyone performed a security review of the system, outside of those with financial stake in the company?
- How many different parties can verify? What are their expertise and incentives?
- What credible assurance comes out of these verification processes, to whom, about what?
  - Is such assurance about a sample implementation before the election, or about the operation of the system during the election? (In other words: does the verification check the system separately from any given election, or a particular election outcome produced using the system?)
- What oversight/verification is there that the outsourced components (both people and software) work as intended?
- What if a bug is found in the code? How do you discover it? How do you address it?
- If a bug is found, how do you plan on communicating this to the public? Why should the public believe that this bug is fixed?
- Are there terms and restrictions on those that perform security analyses of the system? Do they allow security researchers to publicly disclose what they found? Can researchers perform this analysis without permission from the company?
- Is the code open source? Can anyone vet the software, or is the vendor secretive?[21][102]
- Does the vendor have comprehensive public documentation of the system's design, goals, and answers to the other questions asked in this document?

Cryptography, credentials, and PKI

Many schemes that require electronic components make heavy use of cryptography, and often also require that voters and administrators maintain public and private keys. As mentioned briefly in previous sections, key management and verification is itself a challenge that should be studied.

- How are keys managed?
- What happens if one or more keys are compromised?
- Can parties "reset their keys" (choose new keys to replace ones that have been lost or compromised)? Could the recovery procedure be abused?
- What credentials are required to vote?
- How do voters obtain those credentials?
- What happens if credentials are lost or stolen?

Operation

The threat model, design, and cryptography matter very little if the actual on-the-ground use of the system bypasses these restrictions. Operational questions allow us to explore what should happen in the case that reality differs from what we expect.

- What instructions are given to voters, election officials, and others to manage exceptional situations or errors? (E.g., what is a voter supposed to do if they see an incorrect printout or a candidate missing from a ballot?) What evidence enables the error to be confirmed?
- How much outsourcing to vendors is involved in the operational aspects of the election? Can the election outcome be trusted if the vendors are not trusted?
- What if the system is discovered to be malfunctioning during the election? How do you discover it? How do you address it?
- It is easy to design a system that works fine if everything goes as expected. How does the proposed system handle unexpected faults and security violations?
- Could a voter credibly prove how they voted to a third party? (This would violate ballot secrecy.)

Operational privacy

Elections necessarily deal with extensive voter information, including location, address, birth date, party affiliation, and more. Such information can be abused in any number of ways, including disinformation attacks, and should be strongly protected.

- What nongovernmental parties get access to what data? When, and for how long is this data retained?
- Are voters aware of these entities? Are they publicly documented?

21 Lack of vendor receptiveness toward security research, researchers, and audits is frequently a good indicator of lack of sophistication and experience in security. See, e.g., Ref. [102] for more discussion.

- Are there legal constraints on any of these entities? For example, are there contractual obligations for deletion and use of the data? Are these contracts known? For example, do they have public, enforceable privacy policies?

Related work

The US National Academies recently produced an excellent report [103] providing an overview of election security. We note that this report includes a section on "Internet Voting" that briefly discusses whether blockchains can be helpful in providing additional security, which concludes (p. 104) that

> "While the notion of using a blockchain as an immutable ballot box may seem promising, blockchain technology does little to solve the fundamental security issues of elections, and indeed, blockchains introduce additional security vulnerabilities."

Other researchers in the computer security and blockchain fields have written about the risks of blockchain voting in publications such as Slate [104] and The Conversation [8]. For a nuanced discussion of election security requirements more broadly, we recommend Appel and Stark's recent paper [31].

A collection of related online resources is available on Duncan Buell's website [105]. Finally, we cannot resist mentioning the lovely XKCD comic [106] on blockchain voting!

Conclusion

A summary of this article's takeaways follows.

1. **Blockchain technology does not solve the fundamental security problems suffered by all electronic voting systems ("Blockchains as a ballot box" section).** Moreover, blockchains may introduce new problems that nonblockchain-based voting systems would not suffer from.
2. **Electronic, online, and blockchain-based voting systems are more vulnerable to serious failures than available paper-ballot-based alternatives ("Vulnerabilities of electronic voting systems" section).** Moreover, given the state of the art in computer security, they will continue to be so for the foreseeable future.
3. **Adding new technologies to systems may create new potential for attacks.** Particular caution is appropriate in security-critical applications, especially where political pressures may favor an expedited approach ("New problems blockchains introduce" section).

The article has also provided a collection of critical questions intended as a reference point for evaluating any new voting system proposal from a security perspective ("Critical questions" section) and provided references for further reading on this topic ("Related work" section).

Blockchain-based voting methods fail to live up to their apparent promise. While they may appear to offer better security for voting, they do not help to solve the major security problems with online voting, and might well make security worse.

Funding

N.N. and S.P. are supported by the funders of the MIT Digital Currency Initiative. R.L.R. has received support from the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370. M.S. is funded by the MIT Internet Policy Research Initiative and Google's Android Security and PrIvacy REsearch (ASPIRE) Fellowship.

Acknowledgements

We thank Madars Virza and Danny Weitzner for helpful discussions, and Tadge Dryja and Joe Bonneau for their comments on earlier drafts.

References

1. Germann M, Serdült U. Internet voting and turnout: evidence from Switzerland. *Elect Stud* 2017;**47**:1–12.
2. Dandoy R. The impact of e-voting on turnout: insights from the belgian case, 29–37. ISBN: 978-3-907589-17-5. DOI: 10.1109/ICEDEG.2014.6819940.
3. Goodman N, Stokes LC. Reducing the cost of voting: an evaluation of internet voting's effect on turnout. *Br J Polit Sci* 2020;**50**:1155–67.
4. Stewart K, Taylor J. *Online voting: the solution to declining political engagement?*, 2018. https://www.rand.org/blog/2018/03/online-voting-the-solution-to-declining-political-engagement.html [https://perma.cc/DTY4-F54U] (6 January 2021, date last accessed).
5. Serdült U, Germann M, Harris M *et al.* Who are the internet voters?. In: Tambouris E *et al.* (ed.), *Electronic Government and Electronic Participation.* Innovation and the Public Sector. The Netherlands: IOS Press, 2015, 27–41. ISBN: 9781614995692. DOI: 10.3233/978-1-61499-570-8-27.
6. Pew Research Center. *Mobile Fact Sheet*, 12 June 2019. https://www.pewresearch.org/internet/fact-sheet/mobile [https://perma.cc/9DFC-G3LG] (6 January 2021, date last accessed).
7. Jefferson D, Buell D, Skoglund K, *et al. What We Don't Know About the Voatz "Blockchain" Internet Voting System*, 2019. https://cse.sc.edu/buell/blockchain-papers/documents/WhatWeDontKnowAbouttheVoatz_Blockchain_.pdf [https://web.archive.org/web/20200622152400/https://verifiedvoting.org/wp-content/uploads/2020/02/What-We-Dont-Know-About-the-Voatz-Blockchain-Internet-Voting-System.pdf] (6 November 2020, date last accessed).
8. Juels A, Eyal I, Naor O. Blockchains won't fix internet voting security - and could make it worse. *The Conversation*, 2018. http://theconversation.com/blockchains-wont-fix-internet-voting-security-and-could-make-it-worse-104830 [https://perma.cc/2VQQ-25H9] (6 January 2021, date last accessed).
9. Barber G. Wouldn't it be great if people could vote on the blockchain?, 2019. https://www.wired.com/story/wouldnt-it-be-great-if-people-could-vote-on-blockchain [https://perma.cc/CR6P-RMZJ] (6 January 2021, date last accessed).
10. Grauer Y. *What Really Happened with West Virginia's Blockchain Voting Experiment?*, 2019 https://slate.com/technology/2019/07/west-virginia-blockchain-voting-voatz.html [https://perma.cc/H9M5-YJSV] (6 January 2021, date last accessed).
11. West Virginia Secretary of State's Office. *24 Counties to Offer Mobile Voting Option for Military Personnel Overseas*. https://sos.wv.gov/news/Pages/09-20-2018-A.aspx [https://perma.cc/CX3E-YBPQ] (6 January 2021, date last accessed).
12. West Virginia Secretary of State's Office. *Warner Pleased with Participation in Test Pilot for Mobile Voting*. https://sos.wv.gov/news/Pages/11-16-2018-A.aspx [https://perma.cc/7VDD-PZFP] (6 January 2021, date last accessed).
13. Glen Mills. Utah county clerk says mobile voting pilot program was a success. *ABC4*, 2019. https://www.abc4.com/news/utah-county-clerk-says-mobile-voting-pilot-program-was-a-success [https://perma.cc/AN2G-ZVFU] (6 January 2021, date last accessed).
14. Selsky A. *2 Oregon counties offer vote-by-mobile to overseas voters*. AP News, 2019. https://apnews.com/8ce0fbc400514f55839fa84fb364d7f4 (6 January 2021, date last accessed).
15. Specter MA, Koppel J, Weitzner D. *The Ballot Is Busted before the Blockchain: A Security Analysis of Voatz, the First Internet Voting Application Used in U.S. Federal elections*, August 2020. https://www.usenix.org/conference/usenixsecurity20/presentation/specter [https://perma.cc/2B42-REEN] (6 January 2021, date last accessed).

16. Official Website of the Mayor of Moscow. Взломать нельзя, тестировать: программисты проверят надежность электронного голосования. https://www.mos.ru/news/item/58866073 (6 January 2021, date last accessed).
17. Moscow Technologies. moscow-technologies/blockchain-voting. *GitHub*. https://github.com/moscow-technologies/blockchain-voting [https://perma.cc/LL8M-6GN2] (6 January 2021, date last accessed).
18. Krivonosova J. Internet voting in russia: how?. *Medium*, 2019. https://medium.com/@juliakrivonosova/internet-voting-in-russia-how-9382db4da71f [https://perma.cc/EP9B-K6B7] (6 January 2021, date last accessed).
19. Official Website of the Mayor of Moscow. Электронные выборы в Московскую городскую Думу. https://www.mos.ru/city/projects/blockchain-vybory [https://perma.cc/XZB4-FD9F] (6 January 2021, date last accessed).
20. Gaudry P, Golovnev A. Breaking the encryption scheme of the Moscow internet voting system. *Proc. Financial Cryptography '20*. http://fc20.ifca.ai/preproceedings/178.pdf (6 January 2021, date last accessed).
21. Beedham M. Japan is experimenting with a blockchain-powered voting system. The Next Web, 2018. https://thenextweb.com/hardfork/2018/09/03/japan-city-blockchain-voting [https://perma.cc/A9AU-KA74] (6 January 2021, date last accessed).
22. swissinfo.ch. *Switzerland's First Municipal Blockchain Vote Hailed a Success*, 2018. https://www.swissinfo.ch/eng/crypto-valley-_-switzerland-s-first-municipal-blockchain-vote-hailed-a-success/44230928 [https://perma.cc/632Z-BP6M] (6 January 2021, date last accessed).
23. Blaze M, Braun J, Hursti H *et al. DEF CON 26 Voting Village: Report on Cyber Vulnerabilities in U.S. Election Equipment, Databases, and Infrastructure*, 2018. https://www.defcon.org/images/defcon-26/DEF%20CON%2026%20voting%20village%20report.pdf (6 January 2021, date last accessed).
24. Blaze M, Hursti H, MacAlpine M *et al. DEF CON 27 Voting Machine Hacking Village*, 2019. https://media.defcon.org/DEF%20CON%2027/voting-village-report-defcon27.pdf (6 January 2021, date last accessed).
25. Blaze M, Braun J, Hursti H *et al. DEFCON 25 Voting Machine Hacking Village: Report on Cyber Vulnerabilities in U.S. Election Equipment, Databases, and Infrastructure*, 2017. https://www.defcon.org/images/defcon-25/DEF%20CON%2025%20voting%20village%20report.pdf (6 January 2021, date last accessed).
26. U.S. Department of Homeland Security. Election Security. https://www.dhs.gov/topic/election-security [https://perma.cc/2PRL-EMYS] (6 January 2021, date last accessed).
27. Tatham M. Identity theft statistics. *Experian*. https://www.experian.com/blogs/ask-experian/identity-theft-statistics [https://perma.cc/3UEB-JLW5] (6 January 2021, date last accessed).
28. Equifax. Equifax Releases Details on Cybersecurity Incident, Announces Personnel Changes, September 2017. https://investor.equifax.com/news-and-events/news/2017/09-15-2017-224018832 [https://perma.cc/6AD3-P7LV] (6 January 2021, date last accessed).
29. Equifax. Equifax *Data Breach Settlement*. FTC. https://www.ftc.gov/enforcement/cases-proceedings/refunds/equifax-data-breach-settlement [https://perma.cc/38BK-RS33] (6 January 2021, date last accessed).
30. Shaban H. Binance says hackers stole $40 million worth of bitcoin in one transaction. Washington Post, 2019. https://www.washingtonpost.com/technology/2019/05/08/binance-says-hackers-stole-million-worth-bitcoin-one-transaction/ [https://perma.cc/3DGK-MEDM] (6 January 2021, date last accessed).
31. Appel A, Stark PB. Evidence-based elections: create a meaningful paper trail, then audit. *Geo L Technol Rev* 2020;**4**:523–41.
32. Stark PB, Wagner D. Evidence-based elections. *IEEE Secur Priv* 2012;**10**:33–41.
33. *Burson v. freeman*. 504 U.S. 191 (U.S. Supreme Court), 1992.
34. Hays Lowenstein D, Hasen RL, Tokaji DP *et al. Election Law: Cases and Materials. 6th ed. See Page 458 Note (a) for Quoted Text*. Carolina Academic Press, 2017.
35. Rivest RL. On the notion of 'software independence' in voting systems. *Philos Trans R Soc Lond A* 2008;**366**:3759–67.
36. Rivest RL, Virza M. Software independence revisited. In: Hao, F, Ryan, PYA (eds). *Real-World Electronic Voting: Design, Analysis and Deployment*. Taylor & Francis. Chap. 1.
37. Appel AW, DeMillo RA, Stark PB. *Ballot-Marking Devices (Bmds) Cannot Assure the Will of the Voters*, 2019.
38. New Voting Experience. Los Angeles County Registrar-Recorder/County Clerk. https://www.lavote.net/home/voting-elections/voting-options/voting-accessibility/new-voting-experience [https://perma.cc/S328-STFA] (6 January 2021, date last accessed).
39. *H.R. 2722 — SAFE Act (Securing America's Federal Elections Act)*. Congress.gov. Introduced by Rep. Zoe Lofgren on May 5, 2019. Passed the House on June 27, 2019. Received in the Senate on 28 June 2019. https://www.congress.gov/bill/116th-congress/house-bill/2722 [https://perma.cc/NA6K-FMVX].
40. S. *1540 — Election Security Act of 2019*. Congress.gov. Introduced by Sen. Amy Klobuchar on 16 May 2019.
41. *Wyden and Bicameral Coalition Introduce Bill to Require States to Secure Elections*. Ron Wyden's Official Website, 2019. https://www.wyden.senate.gov/news/press-releases/wyden-and-bicameral-coalition-introduce-bill-to-require-states-to-secure-elections- [https://perma.cc/5HCA-Q3AN].
42. S.W.L. What do election observers do?. *The Economist*. https://www.economist.com/the-economist-explains/2017/06/21/what-do-election-observers-do [https://perma.cc/XHV5-SWHG] (6 January 2021, date last accessed).
43. City and Department of Elections County of San Francisco. *Observe the Election Process*. https://sfelections.sfgov.org/observe-election-process [https://perma.cc/3X5L-ETRW] (6 January 2021, date last accessed).
44. European Commission. Eu *Election Missions*. http://ec.europa.eu/info/strategy/relations-non-eu-countries/types-relations-and-partnerships/election-observation/mission-recommendations-repository/home [https://perma.cc/KKL4-EU6N] (6 January 2021, date last accessed).
45. Sparks S, Butler J. Shadow Walker: raising the bar for windows rootkit detection. *Phrack Magazine* 0x0b.0x3d, 2005. http://phrack.org/issues/63/8.html [https://perma.cc/52GP-JMAJ] (6 January 2021, date last accessed).
46. Falliere N, Murchu LO, Chien E. W32. stuxnet dossier. *White Paper, Symantec Corp., Security Response* 2011;**5**:29.
47. Greenberg A. Shopping for zero-days: a price list for hackers' secret software exploits. https://www.forbes.com/sites/andygreenberg/2012/03/23/shopping-for-zero-days-an-price-list-for-hackers-secret-software-exploits/ [https://perma.cc/VF2V-ZSVF]. (23 May 2019, date last accessed).
48. Sultan Niv M. *Election 2016: Trump's Free Media Helped Keep Cost Down*. April 2017. https://www.opensecrets.org/news/2017/04/election-2016-trump-fewer-donors-provided-more-of-the-cash/ [https://perma.cc/6KW5-PPK3](23 May 2019, date last accessed).
49. Mueller RS III. *Report on the Investigation into Russian Interference in the 2016 Presidential Election ("The Mueller Report")*". U.S. Department of Justice, 2019.
50. Anderson RJ. Why cryptosystems fail. *Commun ACM* 1994;**37**:32–40.
51. Adrian D, Bhargavan K, Durumeric Z *et al*. Imperfect forward secrecy: how Diffie-Hellman fails in practice. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2015, 5–17.
52. Brumley D, Boneh D. Remote timing attacks are practical. *Comput Netw* 2005;**48**:701–16.
53. Feldman AJ, Halderman JA, Felten EW. (2007). Security analysis of the diebold accuvote-ts voting machine. In: *2007 USENIX/ACCURATE Electronic Voting Technology Workshop, EVT'07, Boston, MA, USA, 6 August* 2007. Martinez, R Wagner, DA (eds.), USENIX Association, 2007. https://www.usenix.org/conference/evt-07/security-analysis-diebold-accuvote-ts-voting-machine (6 January 2021, date last accessed).
54. Springall D, Finkenauer T, Durumeric Z *et al*. Security analysis of the Estonian internet voting system. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 703–715.

55. Wolchok S, Wustrow E, Isabel D *et al.* Attacking the Washington, DC Internet voting system. In: *International Conference on Financial Cryptography and Data Security*, Springer, 2012, 114–28.
56. Lewis SJ, Pereira O, Teague V. *How Not to Prove Your Election Outcome*. Technical Report, March 2019.
57. Zetter K. *Was Georgia's Election System Hacked in 2016?* https://politi.co/2moAWUS [https://perma.cc/X69Q-DKQP] (23 May 2019, date last accessed).
58. Federal Voting Assistance Program. The uniformed and overseas citizens absentee voting act overview. https://www.fvap.gov/info/laws/uocava [https://perma.cc/A3SK-R2DX] (6 January 2021, date last accessed).
59. Adida B. Advances in cryptographic voting systems. Ph.D. Thesis, MIT, 2006.
60. Adida B. Helios: web-based open-audit voting. In: *Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA*. van Oorschot PC (ed.), USENIX Association, 2008, 335–348. ISBN: 978-1-931971-60-7. http://www.usenix.org/events/sec08/tech/full_papers/adida/adida.pdf (6 January 2021, date last accessed).
61. Benaloh J, Rivest RL, Ryan PYA *et al.* End-to-end verifiability. 15 April 2015.
62. Chaum D. Secret-ballot receipts: true voter-verifiable elections. *IEEE Secur Priv* 2004;**2**:38–47.
63. Overseas Vote Foundation. *The future of voting: End-to-end verifiable internet voting — specification and feasibility study*. (One of the authors, Rivest, was on the Advisory Council for this report.), July 2015.
64. Juels A, Catalano D, Jakobsson M. Coercion-resistant electronic elections. In: *Proceeding 2005 ACM Workshop on Privacy in the Electronic Society*, ACM, **2005**, 61–70.
65. Clarkson M, Chong S, Myers A. Civitas: toward a secure voting system. In: *Proceedings - IEEE Symposium on Security and Privacy* May 2008, 354–68.
66. Iovino V, Rial A, Rønne PB *et al.* Using selene to verify your vote in JCJ. In: Brenner M, Rohloff K, Bonneau J *et al.* (eds.), *Financial Cryptography and Data Security*. Springer International Publishing, 2017, 385–403. ISBN: 978-3-319-70278-0.
67. Wallach D. *On Open Source vs. Disclosed Source Voting Systems*, 2009. https://freedom-to-tinker.com/2009/04/16/open-source-vs-disclosed-source-voting-systems/ [https://perma.cc/6LDL-3KXG] (6 January 2021, date last accessed).
68. Kerckhoffs A. La cryptographie militaire. *J Des Sci Milit* 1883;**IX**:5–83.
69. Fink RA, Sherman AT, Carback R. Tpm meets dre: reducing the trust base for electronic voting using trusted platform modules. *Trans Info For Sec* 2009;**4**:628–637.
70. Follow My Vote. https://followmyvote.com [https://perma.cc/B83K-U3MU] (6 January 2021, date last accessed).
71. Voatz. https://voatz.com [https://perma.cc/J8QA-CCN6] (6 January 2021, date last accessed).
72. Votem. https://www.votem.com [https://perma.cc/E59M-BWBG] (6 January 2021, date last accessed).
73. Narayanan A, Clark J. Bitcoin's academic pedigree. *Commun ACM* 2017;**60**:36–45.
74. Nakamoto S *et al. Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008.
75. Back A, *et al. Hashcash-a Denial of Service Counter-Measure*, 2002.
76. Dwork C, Naor M. Pricing via processing or combatting junk mail. In: *Annual International Cryptology Conference*, Springer, 1992, 139–47.
77. Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof systems. *SIAM J Comput* 1989;**18**:186–208.
78. IBM. IBM food trust. https://www.ibm.com/blockchain/solutions/food-trust [https://perma.cc/V82K-DYYV]. (12 February 2020, date last accessed).
79. Everledger, 12 February 2020. https://www.everledger.io/ [https://perma.cc/3UG7-6EJK] (6 January 2021, date last accessed).
80. Oberhaus D. *The World's Oldest Blockchain Has Been Hiding in the New York Times Since* 1995, 2018. https://www.vice.com/en_us/article/j5nzx4/what-was-the-first-blockchain [https://perma.cc/XG36-TG23] (6 January 2021, date last accessed).
81. Lovejoy J. *Bitcoin Gold (btg) Was 51% Attacked*, January 2020. https://gist.github.com/metalicjames/71321570a105940529e709651d0a9765 [https://perma.cc/2GVA-B8S2] (6 January 2021, date last accessed).
82. Nesbitt M. *Deep Chain Reorganization Detected on Ethereum Classic (etc)*. January 2019. https://blog.coinbase.com/ethereum-classic-etc-is-currently-being-51-attacked-33be13ce32de [https://perma.cc/9RV8-YEL7] (6 January 2021, date last accessed).
83. Nesbitt M. *Vertcoin (vtc) Was Successfully 51% Attacked*. December 2018. https://medium.com/coinmonks/vertcoin-vtc-is-currently-being-51-attacked-53ab633c08a4 [https://perma.cc/S5Z9-XE7L] (6 January 2021, date last accessed).
84. Barrett B. *Hack Brief: Hackers Stole $40 Million from Binance Cryptocurrency Exchange*. Wired, 8 May 2019. https://www.wired.com/story/hack-binance-cryptocurrency-exchange [https://perma.cc/72W6-HMXU] (6 January 2021, date last accessed).
85. McMillan R. Bitcoin exchange Mt. Gox implodes amid allegations of $350 million hack. Wired, 24 February 2014. https://www.wired.com/2014/02/bitcoins-mt-gox-implodes-2 [https://perma.cc/DNP5-ZAKC] (6 January 2021, date last accessed).
86. Ben Sasson E, Chiesa A, Garman C *et al.* Zerocash: decentralized anonymous payments from bitcoin. In: *2014 IEEE Symposium on Security and Privacy*. IEEE, 2014, 459–74.
87. Hopwood D, Bowe S, Hornby T *et al.* Zcash protocol specification. In: *Technical report 2016–1.10. Zerocoin Electric Coin Company*, 2016.
88. Swihart J, Winston B, Bowe S. *Zcash Counterfeiting Vulnerability Successfully Remediated*. https://electriccoin.co/blog/zcash-counterfeiting-vulnerability-successfully-remediated [https://perma.cc/K46G-B2TP] (6 January 2021, date last accessed).
89. Buterin V. *Onward from the Hard Fork*. 26 July 2016. https://blog.ethereum.org/2016/07/26/onward_from_the_hard_fork [https://perma.cc/T2DE-C773] (6 January 2021, date last accessed).
90. Harper C. Bitcoin Independence Day: how this watershed day defines community consensus. *Bitcoin Magazine*, 1 August 2019. https://bitcoinmagazine.com/articles/bitcoin-independence-day-how-this-watershed-day-defines-community-consensus [https://perma.cc/4S5T-LCS4] (6 January 2021, date last accessed).
91. Braydon Fuller and Javed Khan. *Cve-2018-17145*, 9 September 2020. https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2020-September/018164.html [https://perma.cc/QL6P-8VWG] (6 January 2021, date last accessed).
92. Wiki B. *Common Vulnerabilities and Exposures*. https://en.bitcoin.it/wiki/Common_Vulnerabilities_and_Exposures [https://perma.cc/RXL9-RXK7] (6 January 2021, date last accessed).
93. Ruoti S, Kaiser B, Yerukhimovich A, *et al.* Blockchain technology: what is it good for?. *Commun ACM* 2019;**63**:46–53.
94. Scriber BA. A framework for determining blockchain applicability. *IEEE Softw* 2018;**35**:70–7. In:
95. Wüst K, Gervais A. Do you need a blockchain?. In: *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, IEEE, 2018, 45–54.
96. Abraham I, Gueta G, Malkhi D *et al.* Revisiting fast practical byzantine fault tolerance. *arXiv preprint arXiv:1712.01367* 2017.
97. Cachin C, Vukolić M. Blockchain consensus protocols in the wild. *arXiv preprint arXiv:1707.01873*, 2017.
98. dEBRUYNE. A post mortem of the burning bug, 25 September 2018. https://web.getmonero.org/2018/09/25/a-post-mortum-of-the-burning-bug.html [https://perma.cc/B67U-6JFY] (6 January 2021, date last accessed).
99. Peterson J, Krug J. Augur: a decentralized, open source platform for prediction markets. In: *arXiv preprint arXiv: 1501.01042*, 2015.
100. Specter MA, Alex Halderman J. Security analysis of the democracy live online voting system, 2020. https://internetpolicy.mit.edu/wp-content/uploads/2020/06/OmniBallot.pdf (6 January 2021, date last accessed).

101. Bernhard M, Benaloh J, Halderman JA *et al*. Public evidence from secret ballots. In: *International Joint Conference on Electronic Voting*, Springer, 2017, 84–109.
102. Park S, Albert K. *A Researcher's Guide to Some Legal Risks of Security Research*. A joint publication of the Cyberlaw Clinic at Harvard Law School and the Electronic Frontier Foundation, October 2020.
103. National Academies of Sciences, Engineering, and Medicine. *Securing the Vote: Protecting American Democracy*. Washington, DC: The National Academies Press, 6 September 2018.
104. Goodman R, Halderman JA. *Internet Voting Is Happening Now*, January 2020. https://slate.com/technology/2020/01/internet-voting-could-destroy-our-elections.html [https://perma.cc/36LH-R4ML] (6 January 2021, date last accessed).
105. Buell D. Blockchains and voting. https://cse.sc.edu/buell/blockchain-papers (6 November 2020, date last accessed).
106. Munroe R. *Voting Software*, 2018. https://xkcd.com/2030 [https://perma.cc/A2PT-9GH8] (6 January 2021, date last accessed).