Analyzing the Effectiveness of Secure Elections as an Application of Blockchain Technology

Salvatore S. Calvo
Stanford University
Management Science & Engineering Dept.
salcalvo@stanford.edu

Abstract

There is a well-documented open debate as to whether the United States' election systems are sufficiently secure and efficient, with some scholars advocating for a return to paper balloting. It has been idealistically proposed that an electronic voting system powered by blockchain technology can completely eradicate voter fraud while ensuring the vote remains secure and private. I review the existing literature on the vulnerabilities of electronic voting systems in United States and elsewhere, and critique two case studies from 2018 of vote-collecting using blockchain in a real live election: the first in the state of West Virginia's midterm elections and the second in the country of Sierra Leone's national elections.

I. Introduction

"[T]here is no recount procedure in place under the State Supreme Court's order that comports with minimal constitutional standards. Because it is evident that any recount seeking to meet the December 12 [deadline] will be unconstitutional for the reasons we have discussed, we reverse the judgment of the Supreme Court of Florida ordering a recount to proceed."

Per curiam *opinion of the Supreme Court of the United States ordering a stay of the statewide manual recount of the presidential election, Bush v. Gore, Dec. 12, 2000*

"I'm not a state – I'm a monster!"

Lisa Simpson, referencing her disheveled, homemade Florida costume at the Springfield Geography Pageant, S5E10

I have begun to write this paper from a workstation at my gate in the San Francisco International Airport, where the Wifi is plentiful and my three Apple devices are charging for the red-eye flight. It is November 9, 2018, and it is obvious that I am in the epicenter of Silicon Valley as I receive immediate alerts from the *Times* on all three devices at once. I'm not even in New York yet. In fact, the headline that got beamed to me won't even hit the

printing presses in Queens for a couple of hours. But 'all the news that's fit to print' comes to me via pixels instead of ink, and even the content in the headline seems anachronistic. "It's Déjà Vu in Florida, Land of Recounts and Contested Elections" [14], I am told: a full three days after the 2018 midterm elections, manual recounts were triggered in three of Florida's statewide races, including the Senate and gubernatorial races, and had put the outcomes of each race into doubt. Reading the avalanche of political analysis that ensued in the days after, while the election remained undecided – the Senate and governor's races were both called for Republicans Rick Scott and Bill Nelson by razor-thin margins – I was repeatedly struck by two different thoughts. The first was reactionary nostalgia: my very first memory of US national politics was Election Night in 2000, resulting in the *original* distended and combative recount in Broward and Miami-Dade Counties. The second thought was a more practical concern: how, especially eighteen years after *Bush v. Gore*, could elections in the United States remain tipped by the imperfections of the ballot format and of malfunctioning voting machines [7]? And why did this always seem to happen in Florida?

The 2018 Florida midterm election represents an interesting case study – a confluence of the multiple failures and disadvantages that plague the logistics of US election infrastructure. In this particular case, we can start identifying flaws in the simplest places. In Palm Beach County, one of the state's most populous counties and a reliable Democratic stronghold, aging vote-counting machines were reported to have overheated before Election Day as early voting ballots began to be counted [21]. The machines, all manufactured by a now-defunct company called Sequoia Voting Systems, were budgeted to be replaced by the county but never actually disposed of. As a result, a large number of votes went missing during the

crucial recount period. As per the *New York Times*'s reporting, "Palm Beach County found 'dozens of precincts missing a significant number' of votes during the machine recount ... causing the county to conclude that entire boxes of ballots may not have been counted ... the manufacturer of the high-speed scanner used in Palm Beach said its technicians had witnessed Palm Beach County elections workers, apparently worried that one of the machines was running too fast, jam a paper clip into the scanner's "enter" button in an effort to slow it down. That, in turn, caused a short circuit that cut off the power" [18]. Palm Beach County lost thousands of votes overall, a number on the same order of magnitude as the margin of the gubernatorial and Senate races. Meanwhile in Broward County, the manual recount was obfuscated by the county's assertion that it would be several days late in reporting its final vote tally, but also because of a regrettable logistical decision. Despite being on the same ballot, the total number of Broward votes for the governor's race and the Senate race differed by more than 24,000 [11]. Voters simply neglected to check a box for Rick Scott or Bill Nelson. This "undervote" phenomenon is widely suspected to have been caused by the awkward placing of the Senate race options in the lower-left corner of the ballot, perhaps leading voters to not notice it on the ballot.

Situations like these, arising from technical or logistical failures of election infrastructure, are surprisingly common in the world's oldest democracy. They represent their own type of disenfranchisement. The probability that an American's vote is *counted* is a function of her voting jurisdiction's investment in modern electronic voting systems, the proper maintenance of those machines and the competence of the local board's usage of them, and in some cases, the decision of a high court to allow a slow recount to continue. Ensuring the integrity of

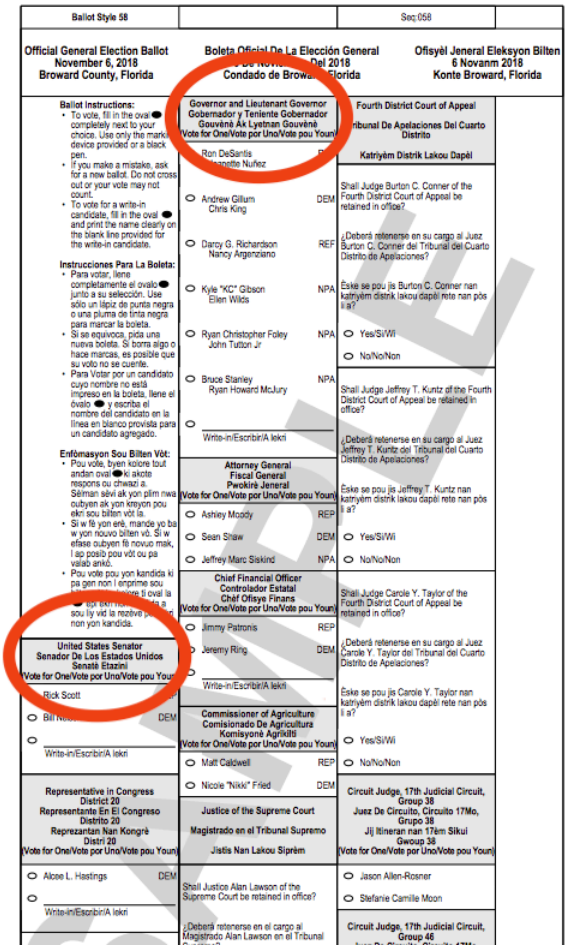

Fig. 1. The sample ballot from Florida's Broward County in the 2018 midterm election, with the placement of the Senate race and the gubernational race both circled. The placement of the Senate race is believed to have caused a systematic undervote because voters hadn't noticed the race at the bottom of the ballot.

elections is thought to encompass a larger scope than these technical issues, of course.

Implementing secure elections should also ensure the more basic threats to the democratic process that come to mind: ensuring ballot boxes aren't stuffed, that ballots aren't thrown away, or that individuals unable to vote don't actually do so. Here I distinguish between an election being *efficient*, i.e., held without logistical or technical challenges, versus being *secure*, i.e. resistant to voter fraud or other types of external manipulation.

The development of cryptography and its related systems over the last 30 years hold much

promise for improving the efficiency and security of democratic elections. Cryptographic systems allow computers to perform a multitude of tasks related to secrecy and security: for example, encrypting and decrypting a communication between two entities, proving that one entity knows a certain piece of information without revealing that information, or digitally "signing" documents to prove their origin. A popular offshoot of cryptography, blockchain technology, provides a means for multiple entities to perform cryptographic operations between each other and the system at large without trusting other entities in the system, and without a central authority to organize the system. Blockchain technology has of course grown incredibly popular in the last five years, owing mostly to digital cash schemes that don't rely on a government's central bank to print money. Many firms, startups and non-profits are currently attempting to apply the technology to many other industries and contexts: supply chains, digital storage. Transparent voting is a natural aspect of the blockchain architecture, and therefore the possibility of applying blockchain technology to secure and efficient elections is a promising one.

I intend for this paper to be a survey of the effectiveness of blockchain technology as a substitute for, or improvement on, electronic voting systems in the United States and elsewhere. Section II will first review the current state of electronic voting systems for democratic elections in the United States and in the developed world: surveying the diversity of methods, machines and vendors by which democracies have conducted recent elections. I will also review the literature of 'attack vectors' and potential flaws that experts have identified in currently used systems, taking note of the documented failures – including the manual counting and machine slowdowns like the ones plaguing Palm Beach County in 2018

– that contribute to imperfect vote-counting and, ultimately, disenfranchisement. Section III theoretically explores political elections as a potential application for blockchain technology. This includes a primer of the key underlying concepts of blockchain technology, aimed at an audience of non-cryptographers or computer scientists: the concept of the ledger, private and public keys, permissioning systems, and the incentives of participants to contribute to the continued maintenance of the blockchain. Here, I describe the interface for an election held electronically, and how features of the system can produce valid votes that are manipulation-resistant, verifiable after the fact, and anonymous. Section IV examines two recent, real-world use cases of the technology in political elections. Both examples are not without controversy. The first example, the June 2018 national elections in Sierra Leone, saw a blockchain-based application record votes in parallel alongside traditional electronic voting system. The second is a pilot program organized by the state government of West Virginia, in which overseas members of the armed forces were allowed to cast their 2018 midterm votes via a blockchain-based mobile application on their cell phones. The two methods – Sierra Leone providing a very traditional 'interface' for voting, and West Virginia's involving a very complex one – are contrasted for their potential advantages toward the goal of ensuring fair elections that maximize turnout (and thus, franchise). Section V synthesizes this analysis into a set of recommendations for applying the technology in the United States. Special considerations are made for the US, a developed country with few instances of legitimate voter fraud but with a decaying electronic voting infrastructure, as well as the source of much of the world's venture capital resources and blockchain-based development talent – no doubt a source of pressure for mass adoption on the state and national level. Section VI approaches the same

question for jurisdictions outside the US, especially in situations where disenfranchisement and explicit voter fraud may be more prevalent. Most importantly, I intend for this paper to be an honest and skeptical look at a technology which, for all its ingenuity, has few widely adopted real-world applications and has attracted a disproportionate amount of hype.

II. A Review of Modern Voting Systems

> "[T]he manufacturer of the high-speed scanner used in Palm Beach said its technicians had witnessed Palm Beach County elections workers, apparently worried that one of the machines was running too fast, jam a paper clip into the scanner's "enter" button in an effort to slow it down."
>
> *The New York* Times*'s description of the Palm Beach County's elections board's attempts at ensuring a smooth manual recount, published in the aftermath of the 2018 midterm elections*

There is abundant academic literature describing the state of modern voting systems in United States elections. As with any engineered system, there is an enumerated set of necessary features and virtues that an elections infrastructure ought to have; electronic voting systems are engineered with these in mind. In the dense survey book edited by Feng Hao and Peter Y. A. Ryan, *Real-World Electronic Voting: Design, Analysis and Deployment*, there is a global consensus surrounding the necessary security features of a voting system (whether that system is electronic or manual) [8]:

- **Vote privacy.** The right of a voting individual to ensure their choice of candidate(s) remains a secret is a fundamental feature of a democratic election, and is explicitly stated in United Nation's Universal Declaration of Human Rights (among other places).
- **Vote verifiability.** The election is trustable from a micro and a macro perspective. From the micro perspective, this means that an individual can confirm that their vote was included in the final tally of their given jurisdiction. From the macro perspective,

this means that an election observer can confirm that the result of the election is valid given the set of all cast votes.

- **Voter verifiability.** The identities of the voter should be able to be confirmed such that every vote is associated with an eligible voter (and only once), and that the eligible voter is in fact the person that cast the vote in his or her own name.

In addition to these, some other basics of product design are vitally important: the voting system should be user-friendly, easily accessible to those mobility issues or other disabilities, and *extremely* fault-tolerant: the consequence of the failure of the voting system is a botched election!

According to a American election transparency advocacy group, VerifiedVoting, there have been five types of voting systems currently in use in the United States during the 21st century [23]. Each achieves the above three features but with varying degrees of sophistication, and with varying levels of accessibility and fault-tolerance. They are:

- **Optical scan paper ballot systems,** where the voter physically marks a paper ballot and a machine scans the markings afterward. Sequoia, the maker of the infamous Broward County machines, produced a machine of this variety, along with vendors such as Diebold and ES&S. The implementation requires a large physical presence of the machine, along with staff of the elections board of the jurisdiction feeding the paper ballots into the scanning machines.
- **Direct recording electronic systems,** where the voter interacts with a computer screen (via a keyboard or touch-screen interface) and their vote is electronically recorded into a secure database. Auditability can be guaranteed by allowing the voter to confirm

their choices on paper before submitting their vote via the interface; which can be used to verify the database of votes in the event of an audit.

- **Ballot-marking devices,** a hybrid of the above two models where a user interface or computer assists the voter in physically marking a paper ballot. The paper ballot is then tallied via optical scan. This option allows the advantage of marking a ballot in a very consistent way, unlike the pen marks or circles drawn onto the ballot in the first option.
- **Hand-counted paper ballots,** which generally are used in situations such as mail-in balloting or absentee / provisional voting is available.
- **Punch-card and mechanical lever voting systems,** which use the assistance of machines to physically perforate a paper ballot. These machines are susceptible to human error, malfunction, and partial malfunction: the infamous hanging chad associated with votes in the 2000 presidential election were created as a result of partial malfunction of these types of machines. As a result, no mechanical lever-based machines were actively used in US elections in 2010, and the last punch-card-based machines were decommissioned in 2016.

While a wide variety of methods exist for implementing elections in the United States, there is a common regulatory regime that voting systems, and the vendors that produce them, must adhere to. In the United States, the jurisdiction who controls the election process must hold its election with officially certified equipment, but its choice of vendor, machine, and implementation is entirely localized to that jurisdiction. It is often the case that this entity is the elections board at the county, or town, level. The regulatory body charged

with certifying voting equipment is the US Election Assistance Commission. The EAC is a bipartisan commission of the federal government, born out of the aftermath of the 2000 election and the passing of the Help America Vote Act (HAVA) of 2002 [4]. The EAC's main responsibilities are to direct grant dollars toward applications from jurisdictions requesting appropriate upgrades to voting equipment, and to certify and decertify voting system hardware and software. HAVA authorized $3 billion of grant funds for the EAC to allocate to grant applicants [10].(Surprisingly prior to passing of HAVA, no federal government protocol or standard existed to advise or require local jurisdictions to keep systems of a certain level of security.) The EAC's *Testing & Certification Program Manual* [5] describes the technical specifications required of systems applying for certification, which involves an approval of all source code and system architecture included in a hardware or software solution, and describes the process by which physical products are subject to extended testing in an EAC laboratory.

There is a surprising lack of competition from US vendors of electronic voting systems. According to a Penn Wharton Public Policy Initiative study on the industry, "[t]he seller side of the election technology industry has come to be characterized by a consolidated, highly concentrated market dominated by a few major vendors, where industry growth and competition are constrained ... [t]he industry has a two-tier structure with the three top-tier vendors, Election Systems and Software("ES&S"), Hart Intercivic ("Hart") and Dominion Voting Systems ("Dominion") covering approximately 92% of the total eligible voter population" [10] with a dozen smaller firms capturing the remaining 8% of the entire market. The market has rapidly become concentrated by the large incumbents, ES&S and

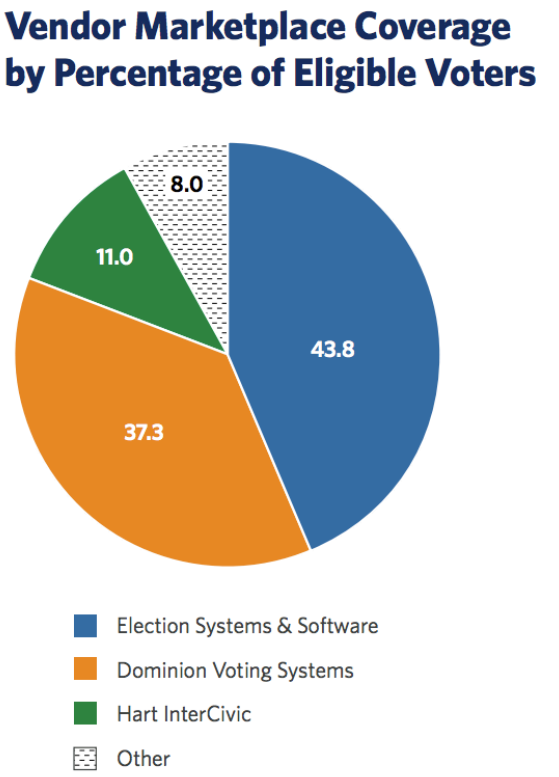

Fig. 2. A pie chart depicting the market share of US electronic voting system vendors in 2016, courtesy of the Penn Wharton Public Policy Initiative's study on the voting system industry.

Dominion, acquiring smaller competitors over the last decade. This concentration coincides with the $3 billion HAVA allocation dwindling over time, and the study concludes that the lack of funding and the subsequent market concentration has diminished the pressure for competitors in the space.

The American market for voting systems stand in stark contrast with the market for the same software in the rest of the developed world, which recently (and especially in the last 15 years) has collaborated with academia on ways to make voting systems more secure. The adoption of cryptographic techniques to ensure election integrity, and to prevent bias in the placement of a referendum on a ballot of the ordering of the candidates' names, has become more common. A system called Prêt à Voter (described in detail in Figure 3), designed in

2004 by Chaum, Ryan and Schneider [8] allows a paper ballot to be submitted by a voter which includes its candidates in a randomized order, but encrypted with a private string of text that the voter can identify as their own. Any individual can audit their own vote's contribution to the set of all votes, as the election results are "publicized" in that votes are tied back to the private strings in a transparent way. A variant of this system was implemented in state-level elections in Australia starting in 2014.

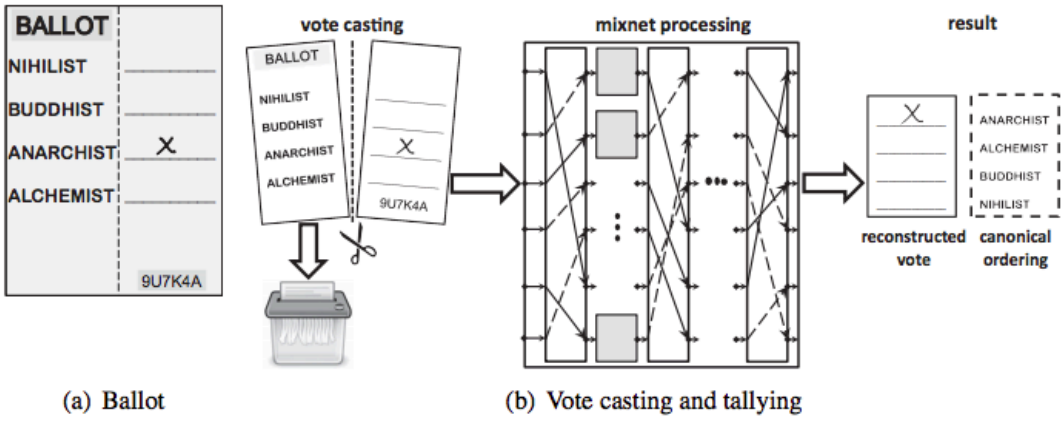

Fig. 3. A diagram predicting an implementation of Prêt à Voter, designed in 2004, courtesy of Feng Hao and Peter Y. A. Ryan's *Real-World Electronic Voting: Design, Analysis and Deployment*. Prêt à Voter introduces a cryptographically secure method of submitting a paper ballot, while allowing for randomized ordering of candidates.

Encryption has proven to be a popular common element in methods recently adopted for paper ballot elections [8], and several other systems are in various stages of design or pilot at this point: Votegrity, Wombat and MarkPledge among the catchy titles. Researchers are beginning to study the propensity for remote voting with electronic ballots, which involves the content of a vote sent over the Internet over an insecure channel. The first devised system, called Adder and designed in 2008, allowed voters and the elections authority to arrange the Adder connection and protocol "offline" before elections begin to take place [8].

This arrangement includes a form of cryptographic encryption, which ensures that the vote could not be manipulated in transit between a voter's machine and the election authority's machine. The key attack vector in such an arrangement is known in the literature as the *untrusted terminal problem*: the possibility that malware on a voter's mobile device can be responsible for incorrectly representing the voter's vote without their realization. This threat, while auditable, remains an issue that presents itself to all electronic voting systems which don't force the user to physically attend the voting precinct, or mark a physical ballot.

In short, there are an abundance of ideas surrounding the proper implementation of secure elections, where most are powered by intuitive but powerful forms of cryptography. There remain, at least in the US, little evidence of the adoption of these technologies and little incentive for the existing vendors in the space to begin adopting the technologies as well. In the parlance of Silicon Valley, the space appears ripe for disruption.

III. **Blockchain Applications to Voting Systems: Theory and Practice**

In 2019, it is not uncommon (and quite fatiguing, even for an engineer) to hear the various Silicon Valley technology buzzwords waft into the conversations one might overhear on the Stanford campus, in downtown Palo Alto, or in the tech-ier parts of San Francisco. Machine learning or artificial intelligence can be applied to every business model and startup, but also to every public policy initiative. Blockchain technology – or in the Valley vernacular, just "the blockchain" – is no different. Since the invention of the technology – widely considered to have taken place in 2008, but arguably long before – an impressive amount of venture financing and employment in the tech sector has been captured by the building of blockchain-

based startups. More than $1.3 billion has been invested in companies in the space in 2019 [19], while the sum market capitalization of all digital currencies – an implicit investment in the space, signifying the owners' expectations that those assets will appreciate and experience wide adoption – exceeds $120 billion as of this writing [1]. The level of hype is probably reminiscent of the dot-com era, and perceptions that the space is a speculative bubble are common [2].

A. ***What is Blockchain, Anyway?***

Just as the automobile is an innovative composition of several 20th-century inventions (an internal combustion engine, an ignition, the usage of gasoline as fuel, etc.), blockchain technology is best described as a confluence of several technological features at once. Here, I do my best to describe these features to a lay audience, using the features common to most extant and popular implementations of the technology.

Blockchain as a state machine. At its most basic level, a blockchain is a database design model. One can imagine that a simple alternative is a traditional database, which holds data in a *ledger*. The ledger contains the complete state of a system at any given time, maybe in rows and columns. An example of a ledger might be a list of a bank's debit accounts, along with the user's current balances. The *state*, then, is subject to change based on additions to or subtractions from a user's bank account, or the addition of a new user and her balance to the ledger. The database's *current state* is usually a snapshot of

[1]Market capitalization of the top 100 digital currencies as measured on https://coinmarketcap.com/ on January 5. The smallest of the constituent currencies has a market capitalization (calculated as the number of currency units in existence times the going price for one unit in USD) of $33 million.

the database's state at the current moment in time – that is, the value of all users' bank accounts after the very last state change was made to the database. Traditional databases generally also exhibit some type of version control, which allows an administrator or auditor to understand past states of the database in addition to the current state. A scheme for version control in our bank database example might include a 'snapshot', or cache, of the state of the database saved at a regular interval like at the end of each day. These caches are saved for posterity, and kept on file for financial reporting or auditing purposes.

A novel feature of a blockchain is the explicit use of these snapshots, called *blocks*, to do version control as to the current state of system. The current state of the system, or the current block, generally contains (1) all data necessary for an observer to understand the current state, (2) the block *height* (essentially the version number of the state, which is monotonically increasing), and crucially, (3) a reference to the previous block, or previous current state, of the system. This reference is called a *hash*, and is essentially a one-way pointer from one block to the next. This is a powerful concept because it allows an observer to confirm that the current state is the child to a specific previous state (which itself is the child of another previous state, and so on back to the origin of the system). Thus the blocks are chained together from the beginning of time to the present, which gives the concept its name.

The concept of a *hash*, or *cryptographic hash function*, is worth a brief description. A cryptographic hash function, according to Narayanan et. al. [16], is a mathematical function that obeys the following properties:

- Its input, x, is string of text of any length

- Its output, $H(x)$, is a fixed-length (for example, always 256-bit) series of bytes,
- It can be efficiently computed in a deterministic way,
- It is extremely difficult to derive the input x from the output $H(x)$ (but of course easy to go from x to $H(X)$)

The mathematical operations happening under the hood of a cryptographic hash function are complex (they generally utilize a concept called elliptic-curve technology), but it is easy to demonstrate why they are useful. For an example, the industry-standard function called SHA-256 can be used to generate a hash of any string of text. (You can Google a number of websites, like this one [2] [17], to generate a complex password using SHA-256.) The below table contains some strings of text input to SHA-256, along with their outputs:

x	H(x)
The quick brown fox jumps over the lazy dog.	EF537F25C895BFA782526529A9B63D97AA631564D5D789C2B765448C8635FB6C
The quick brown fox jumps over the lazy frog.	13D8DECE524DC103C661D56CAF826B4C349830EC9EB147C7C67F1C73F1007CBE
123-45-6789	01A54629EFB952287E554EB23EF69C52097A75AECC0E3A93CA0855AB6D7A31A0
The person with Social Security number 123-45-6789 voted in the 2018 Florida midterm election for: Andrew Gillum (Governor) Rick Scott (Senate) Charlie Crist (FL-13) ...	E8648CE1FE149DA1ACDB154E73FB2616067BFEFAF8F03C7BE2510866E4EFCA04

Fig. 4. A series of strings of text as inputs, and the result of hashing those strings with the SHA-256 algorithm. Any arbitrary file can be hashed to produce an output such as those in the right column; these hashes can serve many purposes, including ensuring the completeness or integrity of a arbitrarily large file, or assisting in the production of a digital signature.

These examples showcase two crucial features of the hash function. The first is that very small differences in the input text produce extremely different strings of output text. This

[2]SHA-256 implementations and those of other hashing and encrypting algorithms can be found at https://passwordsgenerator.net/sha256-hash-generator/.

is desirable because it is infeasible [3] to guess an x from a given $H(x)$ [4], and hashes are unpredictable. The second is that $H(x)$ can be used as a proof of knowledge, since only a person who knows a certain input x can produce the corresponding $H(x)$. In other words, the $H(x)$ can be the public proof of a person knowing some secret, where the x is the secret itself. A good analog is that $H(x)$ is like a username (which identifies you publicly), while x is like a password (which authenticates you privately). In the last example, the voter's Social Security number can in no way be derived from $H(x)$, but only knowing the information in x (the SSN, and the person's exact votes) can produce $H(x)$.

Blockchain as a decentralized database. The arrangement described above can be implemented as a single database, but a key limitation remains: one central authority decides on the state of the system, and has the ability to decide the series of state changes to include in the blocks. In our banking example, a bank has the ability to reject certain withdrawals or other transactions if it wishes; a rogue bank could decide to arbitrarily decline certain transactions or place a lock on a given account. Blockchain architectures solve for this 'trust' problem by allowing the state and its history ('the blockchain') to be redundantly stored in several locations, each called a node in a network. Essentially, each node is a separate

[3] *'Infeasible'* is defined as possible, but with processing power that grows exponentially faster than the complexity of the output grows. This means it is also possible to eventually generate two different x values that produce the same $H(x$ value, which is known as a collision. The likelihood of finding a collision in SHA-256 is the same generating a hash randomly, and then generating the same hash again randomly. Since there are 2^{256} possible SHA-256 hashes based on the combinations of bytes – this is a number with 78 digits – it is extremely unlikely to occur.

[4] Importantly, hash functions are different from *encryption* functions, which can be reversed. Thus, the application for a hash function is either to create entropy (for example, create a good random string for a password) or to prove that you know an input x to someone else who knows x, by sharing $H(x)$ instead.

computer server that stores its own copy of the blockchain. [5] Each user who seeks to make a transaction simultaneously broadcasts their transaction to *every* node in the network, and each node keeps a running tally of each transaction to be included in the next block to be produced. The series of transactions are considered *unconfirmed* until a new block is produced, and the act of each node including the transactions in the output of the hash function is what confirms each transaction.

Blockchain as a way to incentivize keeping the database accurate. If each node in the network above is 'honest', i.e., will accept each transaction that a user submits to it, then every node in the system will keep the same list of transactions. The situation of nodes being in agreement is called *consensus* – and there is a rich literature concerning the incentive structures of how many non-honest nodes a network can withstand. The point of having the network contain multiple nodes, though, is the ability to ignore a minority of nodes that produce incorrect data for whatever reason. (Nodes may be intentionally dishonest – choosing to maliciously report a state of the world which is in fact untrue – or unintentionally dishonest, due to a technical failure. Both situations must be accounted for in decentralized systems, and in fact aren't treated differently from the perspective of producing consensus.) The incentives question is a curious one: the nodes in the network need to have some selfish, vested interest in contributing to the accuracy of the network and to the production of a consensus state. In a sense, the original blockchain alternative (just

[5]This arrangement contains a remarkable similarity to another popular network: the DNS nameservers that power all traffic on the Internet. Every machine connected to the Internet has a unique IP address – for example, the machine hosting google.com has the IP address 172.217.12.206. You can just type "172.217.12.206" into your Chrome browser and you'll be taken to the Google website, but of course this isn't as user-friendly as using the URL. Any attempt to use a URL in a browser will actually send a request to a node of the Domain Name System. There are 13 root servers in the DNS, all around the world, who keep consensus on the 'ledger' of the Internet: which URLs map to which IP addresses.

one database) is really a network with one node – and, trivially, it cares about the state of the world that it produces. If the same entity controlled all of the nodes in the network, the question then arises of why the authority which controls the whole network has redundant nodes in the first place. To ensure the decentralization of the platform, different entities must control each node separately. In the most widely adopted blockchain systems (Bitcoin and Ethereum among them), the solution for this 'consensus with competition' problem is to monetarily reward the nodes in the network for producing information which is in consensus with the rest of the network. This is in fact the origin of the digital currency phenomenon. Rather than digital currencies like Bitcoin being a means by which nodes in a network agree to maintain the state of a database, the situation is precisely reversed. Bitcoin, the original blockchain system, was designed precisely as a digital cash scheme that didn't require a central authority to print money or maintain the state of the ledger. In the case of Bitcoin, nodes in the network are referred to as *miners*. Their role in the ecosystem is to maintain the record of all unconfirmed transactions, while simultaneously attempting to solve a cryptographic puzzle [16], in a competition with the other miners in the network. If a miner solves the puzzle first (and the system is calibrated to ensure the mean time between solutions is about 10 minutes), the winning miner gets the ability to build the newest block describing the new state of the system. In that block contains the ledger of balances associated with anonymous addresses of the network – and the miner is allowed the reward of adding a fixed number of Bitcoin units to its own balance. This ensures that individually selfish miners all operate in the same way, are incentivized toward the consensus of information in the blocks, and are invested in the long-term stability of the

network (because it contains value that they are literally investing in).

B. ***Applications of Blockchain Technology***

The applications of blockchain technology that have been broadly adopted are few. For the reasons described above, the success of a blockchain-based network is directly related to the incentives users have to participate in the network, and the expectation that the network remains stable and accurate over time [16]. Because of this constraint, it is unsurprising that the most successful applications of blockchain technology has been the invention of digital currencies. Financial applications are by far the most natural applications: they provide a method of value transfer that is completely unarranged by a financial intermediary (apart from an Internet connection). Bitcoin (designed in 2008) [16] and Ethereum (designed in 2015) now both comprise vast ecosystems of users, mining nodes in the network, and companies that are building financial infrastructure for digital currencies. As the space matures, digital currencies are beginning to resemble a permanent alternative asset class, like precious metals. Much of the effort and capital entering the space is likely to be the result of speculation; that said, pieces of the financial infrastructure like the enablement of custody, debt markets, and derivatives markets may cause the largest digital currencies to become institutionalized in a more permanent way. While Bitcoin exists simply as a currency, Ethereum is a more generalized blockchain that *contains* a currency but also is a decentralized ecosystem to store code. This is a powerful concept. Just as the network of Bitcoin nodes exist to maintain the blockchain's state while waiting for the chance to win a reward, Ethereum nodes also store objects called *smart contracts* in their database. A smart

contract is a piece of software that executes when its address is called: every node stores a copy, and the code can never be updated or rewritten. There is important permanence about this: since the node is stored everywhere at once, it's similar to storing money in a safe, just in a safe deposit box in every bank in the world. In fact, escrow of digital currency is an important use case of smart contracts. So too is collateral, or even the minting of secondary digital currencies whose balances are stored in one Ethereum smart contract. These use cases are widespread, but (to this observer) are limited to finance. Despite this, a very large number of technology startups have attempted to apply blockchain technology to other fields with limited success. I describe some of the more interesting ones briefly below:

- **Prediction markets and gambling.** The combination of escrow and Ethereum smart contracts in 2018 produced a tool called Augur, which hosts decentralized prediction markets based on the real-life outcomes of events like sports, politics and the weather. Users can buy into shares of an outcome, which will very with the betting population's beliefs in the likelihood of that outcome. (This dynamic auction pricing is exactly the same way that, for example, horsetrack betting is dynamically priced.)
- **Online collectibles.** The storage of data in a smart contract has been used to create collectible items in online games, as well as a secondary market for those collectibles which have been made artificially scarce.
- **Supply chain recording.** International trade multinationals are currently testing blockchain as an application for providing a canonical source for recording inventory as it moves throughout a vast supply chain [22].
- **Decentralized file storage.** Applications such as Filecoin are attempting to create a

decentralized market for data storage, in the same way that the Ethereum network decentralized the execution of code with smart contracts. The intention is for the decentralized file system to erode confidence in centralized solutions for data storage, like Google Cloud or Amazon Web Services. Filecoin is 'subsidized' with its own digital currency, which is only as valuable as the miners maintaining the network consider the currency to be.

- **Information oracles.** Finally, applications such as Witnet attempt to crowdsource the addition of external information onto a blockchain – a problem that has been whimsically referred to as the 'oracle problem'. As an example, information like the price of a stock, or the weather in a city, could be listed onto the blockchain in a way that requires to central party to create the information. Here, two parties must be incentivized to maintain the network: the miners as before, and the role of user who is incentivized to produce the external information. As with the other applications, the system must be incentivized with the usage of an on-chain digital currency.

*C. **Two Implementations of Voting on the Blockchain***

Given the above applications, it is quite trivial to create an arbitrary balloting system onto an existing blockchain or a new blockchain entirely. Such a system would already include some of the necessary features described in Section II, namely the support of vote and voter privacy, and auditability of the end result. Figure 6 displays the code that deploys a simple balloting system into an Ethereum smart contract.

In addition to the co-opting of existing blockchain systems to implement a voting mech-

```
pragma solidity ^0.4.0;
contract Ballot {

    struct Voter {
        uint weight;
        bool voted;
        uint8 vote;
        address delegate;
    }
    struct Proposal {
        uint voteCount;
    }

    address chairperson;
    mapping(address => Voter) voters;
    Proposal[] proposals;

    /// Create a new ballot with $(_numProposals) different proposals.
    function Ballot(uint8 _numProposals) public {
        chairperson = msg.sender;
        voters[chairperson].weight = 1;
        proposals.length = _numProposals;
    }

    /// Give $(toVoter) the right to vote on this ballot.
    /// May only be called by $(chairperson).
    function giveRightToVote(address toVoter) public {
        if (msg.sender != chairperson || voters[toVoter].voted) return;
        voters[toVoter].weight = 1;
    }

    /// Delegate your vote to the voter $(to).
    function delegate(address to) public {
        Voter storage sender = voters[msg.sender]; // assigns reference
        if (sender.voted) return;
        while (voters[to].delegate != address(0) && voters[to].delegate != msg.sender)
            to = voters[to].delegate;
        if (to == msg.sender) return;
        sender.voted = true;
        sender.delegate = to;
        Voter storage delegateTo = voters[to];
        if (delegateTo.voted)
            proposals[delegateTo.vote].voteCount += sender.weight;
        else
            delegateTo.weight += sender.weight;
    }

    /// Give a single vote to proposal $(toProposal).
    function vote(uint8 toProposal) public {
        Voter storage sender = voters[msg.sender];
        if (sender.voted || toProposal >= proposals.length) return;
        sender.voted = true;
        sender.vote = toProposal;
        proposals[toProposal].voteCount += sender.weight;
    }

    function winningProposal() public constant returns (uint8 _winningProposal) {
        uint256 winningVoteCount = 0;
        for (uint8 prop = 0; prop < proposals.length; prop++)
            if (proposals[prop].voteCount > winningVoteCount) {
                winningVoteCount = proposals[prop].voteCount;
                _winningProposal = prop;
            }
    }
}
```

Fig. 5. Code for a simple smart contract implementing a balloting system, written in the Solidity programming language of the Ethereum network. This implementation is one of the standard files included in the Ethereum network's development environment, Solidity. Note that the boilerplate code supports one chairperson address, who is the authority of the referendum; this address has the ability to hold referenda, give other addresses the right to vote, and even can determine the weights by which address can vote. This is a simple implementation, and thus excludes two necessary features: the ability to link an address deterministically to a person's real-life identity, and the ability to prevent the user's application from submitting their vote maliciously.

anism, separate and new systems have been created that are specifically designed for the task. I analyze and critique both here.

1) ***Voatz****:* Voatz, despite its unfortunate name, is thus far the only technology to have successfully built and deployed blockchain technology to tally votes in an American election. The company, only founded in late 2017, received a seed round of funding in January 2018 with the stated purpose of 'making voting more secure, reliable, and unhackable' [13]. Voatz's architecture builds upon a blockchain infrastructure project called HyperLedger, which IBM originally started. A key enabling feature is that the Voatz blockchain is *permissioned* as opposed to *permissionless*. Bitcoin and Ethereum, for example, are permissionless blockchains in that any person can occupy any role in the system (currency owner, miner, node) without ever divulging their identity or needing to authenticate into the system. A permissioned blockchain, by comparison, forces any user to authenticate into the system with some off-chain identifier. For the purposes of voting in a democratic election, this authentication is done with the cooperation of the election authority's voter identification documents [24]. This authentication is tied to a voter's mobile device, and Voatz claims to anonymize both the record of the device and the record of the voter when the data enters the Voatz blockchain to be stored. Another key feature is the method of auditability that the Voatz blockchain provides – a redundant, paper ballot that is printed in parallel with the blockchain vote. According to the company, "a paper ballot is printed for each mobile ballot submitted on the blockchain, then tabulated like a normal absentee ballot. This ballot contains information that can be used in an audit to ensure that every vote cast from a smartphone was counted exactly once, and counted correctly. For [the first live election use case] a real-time voter-verified paper trail will be generated, which will allow the state to conduct a post-election audit." [24]

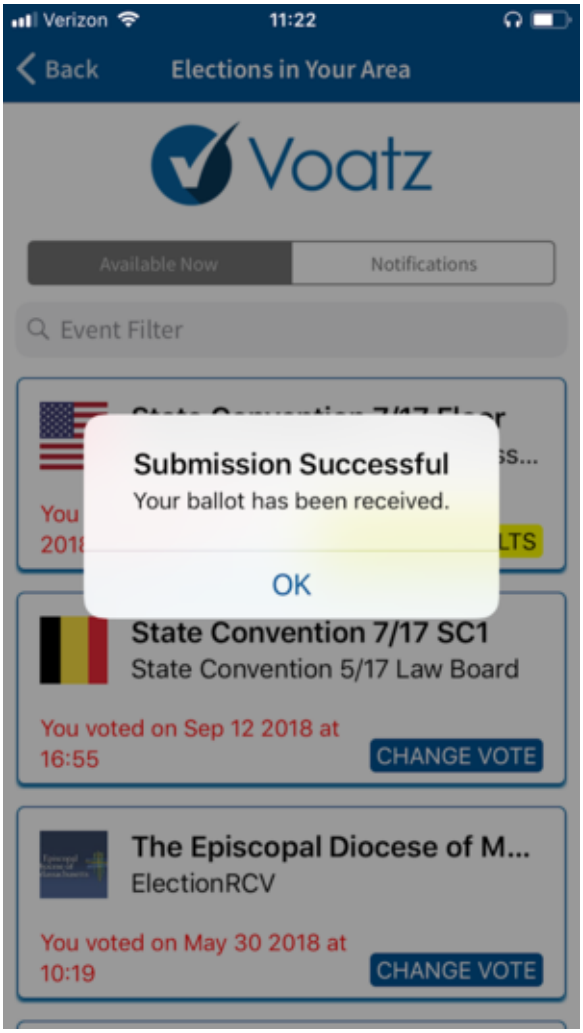

Fig. 6. A screenshot of the Voatz application, accessed on the company blog and as seen from the Apple App Store. The current screenshot would be what an authenticated user would see, who has supplied identifying information to participate in a state election in West Virginia, and also to a church diocese's group election.

While much of this sounds impressive, Voatz has earned critique for the design of its offering, and for the lax implementation of some of its security features [6]. Based on the compelling features of a blockchain described in Section III, Voatz's solution actually doesn't offer a trustable decentralized platform. The nodes of the blockchain network – there are eight of them, spread geographically across Amazon and Microsoft cloud servers [24] – are all controlled by Voatz. In theory, blockchains are designed to (1) form a network of selfish

nodes whose self-interest compels them to record the same set of vote transactions, and (2) be rewarded to do this via a financial incentive. Since the Voatz blockchain contains no fungible digital currency, and the company controls all nodes – the Voatz blockchain is, in effect, just a redundant database with the same corruption concerns associated with a single database managed by a single rogue party.

2) ***Agora****:* Agora, a well-funded Swiss startup founded in 2015, has a very self-aware sense of positioning. It describes itself as a "voting technology company that has developed an end-to-end verifiable voting solution for governments and institutions. Today's voting systems are slow, costly and exposed to many vulnerabilities that can inhibit free and fair elections. Our team of skilled cryptographers and security scientists has built a blockchain-based solution to provide our partners with a modern, provably secure and cost-effective manner of engaging voters. Elections on Agora's network are tamper-proof throughout the entire voting process and offer full transparency to voters, third-party auditors and the general public" [1].

Agora cites the black-box nature of the current electronic voting systems market, and the archaic nature of the paper ballot, as features it improves upon. Specific features are more promising than Voatz's implementation, including the encryption of included vote content, the anonymizing of any user-identifying data that enters its blockchain. The encryption of vote data, which takes place inside a secure part of the Agora app called the *Voting Booth*, proves to users that the encryption system is working properly on their device: a good solution to the *untrusted terminal problem*.

Agora's blockchain is also permissioned as Voatz's is, but is a multi-layer system which

relies upon consensus observation of the Bitcoin blockchain for consensus. In other words, nodes of the Agora blockchain use a fungible digital currency as the incentive to properly perform consensus. While user permissioning occurs, permissioning of nodes does as well: according to the firm's whitepaper, potential consensus nodes "must be evaluated as a partner of Agora to be authenticated on the network" [1].

IV. **Examining Real-World Examples of Blockchain-Based Democratic Elections**

Both of the implementations above have been piloted during a live election. To the indifferent Google searcher, there has been great PR and fanfare associated with each; to anyone more willing to dig into the specifics of the implementation, that person will be disappointed in the outcome.

In the case of Voatz, the company has throughout 2017 been testing its app in non-governmental elections in the US on a pilot basis [6]. These elections include small organizations' management elections, state caucuses of the two major political parties, and referenda for the boards of universities [24]. Voatz claims that the largest of these elections that ran successfully managed to collect 15,000 votes accurately. Voatz also admits a failure during one election: a state party caucus in the state of Utah, in 2017, where technical difficulties caused the app to crash for every voter in the room. In the words of the co-founders, "we experienced an instance of an on-premise election in Utah where we were unsuccessful in meeting the needs of the client. We were unable to support the large numbers of voters who simultaneously attempted to download the app and become verified within a short

30-minute period before voting started." [24]

Despite this lack of robustness, the company was able to successfully land the contract for the state of West Virginia to be the sole electronic voting system catering to overseas citizens of the armed forces filing absentee ballots in the state. Voatz replaced a paper ballot initative here, with the support of the West Virginia Secretary of State, and partnered with a technology company approved by the EAC to ensure compliance. Citizens of the 24 West Virginia counties which participated in the pilot program were able to use the Voatz app to authenticate their identity, necessitating a screenshot of an identifying document and a 'video selfie' of the voter's face, before voting. A press release by the Secretary of State of West Virginia confirmed the results to a certain amount of fanfare, but described the vote totals were particularly low [24]. In total, only 144 total votes were cast via the Voatz app from West Virginians in the eligible counties – a relatively small total, owing to the small number of West Virginians who were on active duty overseas, but also perhaps due to insuffient turnout on the new medium. This total, much less than many of the domestic pilots that Voatz has performed, lacks the scale required for ensuring confidence in the tool.

Agora's pilot was even more disappointing, as its role in the 2018 national elections in Sierra Leone are actually in dispute. First, a description of the environment of the country in questino. Sierra Leone, the West African nation of 7.7 million, neighbors Guinea and Liberia on the Atlantic coast. Originally a colony of the British crown, it gained its independence as a democratic republic in 1961 after a year-long diplomatic negotiation with the Government of Queen Elizabeth II. The country's independent history has been tumultuous. The chief architect of the independence talks, a Sierra Leonean doctor and politician named Milton

Margai, unexpectedly died during his first term as the inaugural prime minister. The rest of the 1960s were marked by political battles for succession and followed my outright military coups. These led to the twenty-year dominance of a one-party authoritarian regime, the All People's Congress, and ultimately to a bloody civil war fought between 1991 and 2001. Intervention by the United Nations ultimately led to a ceasefire, after which a multi-party democratic order has taken shape. The nation continues to slowly develop – its economy overly reliant on agriculture and the diamond mining industry and decimated by the war – but its relatively young population and access to natural resources give it a similar profile to other developing African nations.

The opportunity to contribute to a stable election in a country with a checkered past is valuable, and apparently was extremely valuable for Agora: enough to greatly exaggerate that contribution. After the elections on March 7, 2018, the company issued a self-congratulatory press release claiming that the event "represent[ed] the first time in history that blockchain technology has been used in a national government election" [9]. The international tech press picked up the press release without verifying the claim, and the next day the government of Sierra Leone's official elections body publicly disputed the claim. What actually happened? Ultimately, "Agora obtained permission from the NEC to act as "an international observer" at 280 of roughly 11,200 polling stations. Sierra Leone election officials recorded the paper votes as they would in any other election. Then, Agora's team recorded those same votes on its blockchain" [9]. This situation remains a proof of concept for Agora, but the technical 'success' of the election was betrayed by the fact that the company exaggerated (at best, misled at worst) the international press on describing its

claim.

V. **Recommendations for Adoption in the United States and Abroad**

While blockchain technology remains an intriguing technology for Silicon Valley, for digital currency speculators, and for academic researchers: it is obvious to me that both the technology being built in the election voting space and the culture of the companies building the tools, has a long way to mature. The successful pilots are somewhat promising, to be sure. The disadvantages brought about by the status quo, of course, do not negate our need for a solution to the decaying election infrastructure in the United States.

This exercise highlights to me a certain set of steps that election authorities ought to abide by, regardless of locale. They are:

- In America, allocate more grant money towards jurisdictions in need of updated or new electronic voting systems, in the form of a new HAVA budget. This allocation will serve the short-term goal of replacing desperately needed equipment in certain jurisdictions. It will also spur new competition and innovation into the electronic voting system market in the US, solving the long-term problem of insecure or inefficient elections.
- In America, begin developing electronic voting systems that employ cryptography, encryption, and hashing functions to (1) anonymize ballots, (2) allow for the verifiability of votes, and (3) allow for the verifiability of inclusion in the set of all votes. In other countries as well, such systems can increase the confidence that citizens have in the democratic process.
- In America, begin developing electronic voting systems that identify users with a device

and a document. The blockchain implementations described above both succeed at this step, which is easy to implement (via the cross-referencing of photo IDs in jurisdictional databases) due to the ubiquity of cell phones.

- In all countries, only develop blockchain-based voting systems if their use cases involve a blockchain with a fungible, useful digital currency component and which the same third party does not control all nodes. Without these components, a blockchain is redundant at best and an attack vector at worst.

VI. **Conclusion**

As I observe the aftermath of another American election, it is difficult to overstate the country's need for a robust election infrastructure. Our current ecosystem, like the defunct and overheated machines in Palm Beach County, are decaying while the incumbent providers of technology are failing to innovate. The problem affects all of the world's democracies: even when disfranchisement is not explicit and political, voter turnout and machine malfunction serve as a logistical form of suppression. There are reasons to be optimistic about how technological advance can increase participation in the democratic process: the addition of cryptographic security, the ability to audit. But not all technological advances are valuable for this use case. Like all immature technological advances, the first year of raised capital and of a minimum viable product can result in vastly disproporationate expectations for investor and consumer alike. The blockchain boom has certainly proven to be an example of this. While the Valley will continue its experimentations with blockchain technology, and the technology will likely continue to proliferate for its naturally intended use cases – the

improvements to be made to the democratic process will, likely, exclude it.

References

[1] Inc. Agora. *Agora: Bringing our voting systems into the 21st century*. 2018. URL: https://static1.squarespace.com/static/5b0be2f4e2ccd12e7e8a9be9/t/5b6c38550e2e725e9cad3f18/1533818968655/Agora_Whitepaper.pdf.

[2] Tracy Alloway. "A Year After the Crypto Bubble Burst, Will Bitcoin Ever Recover?" In: *Bloomberg BusinessWeek* (Dec. 3, 2018). URL: https://www.bloomberg.com/news/features/2018-12-13/where-does-bitcoin-go-from-here.

[3] CoinMarketCap. *Top 100 Cryptocurrencies by Market Capitalization*. 2016. URL: https://coinmarketcap.com/.

[4] Election Assistance Commission. *About U.S. EAC*. 2018. URL: https://www.eac.gov/about/.

[5] Election Assistance Commission. *Testing & Certification Program Manual, Version 2*. 2015. URL: https://www.eac.gov/assets/1/28/Cert.Manual.4.1.15.FINAL.pdf.

[6] Jon Evans. "Voatz: a tale of a terrible, horrible, no-good, very bad idea". In: (2018). URL: https://techcrunch.com/2018/08/11/voatz-a-tale-of-a-terrible-horrible-no-good-very-bad-idea/.

[7] Tara Golshan. "Florida's high-stakes recounts for governor and Senate, explained". In: *Vox* (Nov. 15, 2018). URL: https://www.vox.com/2018/11/15/18096697/florida-recount-bill-nelson-rick-scott-desantis-gillum.

[8] Feng Hao and Peter Ryan. *Real World Electronic Voting: Design, Analysis and Deployment*. CRC Press, 2017.

[9] Kristin Houser. "Hold Up: What Actually Happened in Sierra Leone's "Blockchain" Election?" In: *Futurism* (Apr. 2, 2018). URL: https://futurism.com/sierra-leone-election-blockchain-agora.

[10] Penn Wharton Public Policy Initiative. *The Business of Voting*. 2016. URL: https://publicpolicy.wharton.upenn.edu/live/files/270-the-business-of-votin.

[11] Elizabeth Koh and David Smiley. "In Broward County, fewer votes in Senate race raise questions about recount". In: *Miami Herald* (Nov. 8, 2018). URL: In%20Broward%20County , %20fewer % 20votes % 20in % 20Senate % 20race % 20raise % 20questions % 20about%20recount.

[12] German Lopez. "The Florida voter fraud allegations, explained". In: *Vox* (Nov. 12, 2018). URL: https://www.vox.com/policy-and-politics/2018/11/12/18084786/florida-midterm-elections-senate-governor-results-fraud.

[13] Mo Marshall. "Voatz raises 2.2 million to make elections tamper-proof". In: (2018). URL: https://venturebeat.com/2018/01/08/voatz-raises-2-2-million-to-make-elections-tamper-proof/.

[14] Patricia Mazzei and Frances Robles. "It's Déjà Vu in Florida, Land of Recounts and Contested Elections". In: *The New York Times* (Nov. 9, 2018). URL: https://www.nytimes.com/2018/11/09/us/florida-ballots-recount-scott-nelson-gillum-desantis.html?module=inline.

[15] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: *Self-published* (Jan. 9, 2009). URL: https://bitcoin.org/bitcoin.pdf.

[16] Arvind Narayanan et al. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2019-01-01.

[17] PasswordsGenerator.net. *SHA256 Hash Generator*. 2012. URL: https://passwordsgenerator.net/sha256-hash-generator/.

[18] Frances Robles. "Nearly 3,000 Votes Disappeared From Florida's Recount. That's Not Supposed to Happen". In: *The New York Times* (Nov. 16, 2009). URL: https://www.nytimes.com/2018/11/16/us/voting-machines-florida.html.

[19] Jason Rowley. "With at least 1.3 billion invested globally in 2018, VC funding for blockchain blows past 2017 totals". In: *TechCrunch* (May 20, 2018). URL: https://techcrunch.com/2018/05/20/with-at-least-1-3-billion-invested-globally-in-2018-vc-funding-for-blockchain-blows-past-2017-totals/.

[20] Siamak Shahandashti. "Electoral Systems Used Around the World". In: *Working paper* (2016). URL: https://www.researchgate.net/publication/301897933_Electoral_Systems_Used_around_the_World/download.

[21] Allen Smith. "Palm Beach vote machines overheat, botching recount in Florida Senate race". In: *NBC News* (Nov. 14, 2018). URL: https://www.nbcnews.com/politics/elections/palm-beach-vote-machines-overheat-botching-recount-florida-senate-race-n936076.

[22] Gerry Tsoukalas. "Blockchain and the Value of Operational Transparency for Supply Chain Finance". In: (2018). URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3078945.

[23] VerifiedVoting. *Voting Equipment in the United States*. 2016. URL: https://www.verifiedvoting.org/resources/voting-equipment/.

[24] Blog at Voatz. *Official Statement from Voatz regarding Mobile Voting Pilot in West Virginia*. 2018-08-07. URL: https://blog.voatz.com/?p=454.