Utilizing Blockchain Technology to Revamp Voting Systems

Christopher Boggs, Benjamin Cohen, Hossein Golestani, Tiberiu Vilcu
22 April 2018

Abstract

In this paper, we describe a new voting model which utilizes blockchain technology to increase efficiency, lower cost, and improve security guarantees and public trust. Recent elections have shown that current voting models are insecure and citizens are rapidly losing faith in the voting process. Our revamped model builds on top of present in-person security practices and introduces voting machines that transact ballots on a distributed blockchain. We have implemented a suite of software to test and validate the model, and provide reasoning for its practicality and benefits over current practices and other attempts. The proposed model addresses many of the current weaknesses in voting systems and can help countries protect their democracies from malicious forces through modernization of the voting process.

1 Introduction

This is a research project for EECS 591 Distributed Systems where we propose new a voting system to improve security of elections by utilizing blockchain technology.

Every democratic country in the world values their electoral processes more than just about anything because they are the backbone to democracy. Technology has evolved exponentially since the dawn of democracy, but voting methods have not adapted to utilize this technology. Recent elections all over the world have had serious security concerns regarding the validity of electronic voting. Online voting has also been attempted in recent years but has been unsuccessful for various security reasons.

Citizens are losing faith in their democracies due to insecure voting methods. In order to restore faith in voting systems, we have leveraged the new decentralized, fault-tolerant ledger, blockchain, to create a modern, secure voting system. The recent emergence of blockchain presents a new medium for voting, and we have created a revamped model that uses the same in-person security protocols as current elections in conjunction with blockchain technology to transact and count votes. Our voting model allows voters to remain anonymous, while ensuring the casting of votes as intended through utilizing blockchain's Byzantine-Fault tolerance protocol in addition to a secure private network setup.

2 Related Work

Throughout recent years, there have been many attempts to bring technology into the voting process. Similarly, there have also been many attacks on electronic voting systems, uncovering many vulnerabilities, and a national spotlight was even cast on voter fraud during the most recent election. This section will outline a few of these instances.

2.1 Online Voting

Forms of internet voting include voting through email and voting through an online Web forum. In the early 2000s, the United States spent hundreds of millions of dollars developing the Secure Electronic Registration and Voting Experiment (SERVE) [4]. The government developed the SERVE system to help people in the military vote from overseas and to help American citizens living outside the United States vote. SERVE is an internet and PC-based system. Ultimately, SERVE was terminated after multiple security breaches during testing. Another proposed online voting system was created in Arizona, where voters would receive a personal identification number in the mail and then they would have to enter their personal identification number online and answer two personal questions to cast their vote. This voting method was used in a primary election and is not used any longer. Another example of an internet voting system was developed by the country of Estonia. Estonia uses legally-binding online voting in their general elections. Estonian officials claim the system works well, however the system has received sharp criticism from computer security experts [3].

2.2 Voting Machines

There are many forms of electronic voting machines currently used in elections. These machines have a variety of different issues regarding voting accuracy and voter security. Voters typically do not have access to the code running the machines, so they have to trust that the code works and is not designed

maliciously. Typically, machines will also create a paper trail of the votes in addition to counting them electronically, so the paper trail can be used to verify the count; however, some machines do not do this and most elections do not require machines to use this verification method. Additionally, some machines also require voters to enter personal information to verify their identity, which leaves the system susceptible to violating the anonymity of the voters. Electronic voting machines are also susceptible to a variety of hacking methods. Many of these machines have open physical ports that could be used to equip the machines with malicious software or to tamper with the votes directly. Security researchers also showed that some of these machines could be hacked wirelessly [7].

3 Background

The current process to vote in elections in the United States includes two steps: Registration and Vote Casting. The vast majority of people vote at polling sites although some people submit absentee ballots through mail.

Registration to vote is handled on a per-state basis, so American citizens register to vote in the state they live in. In order to register, voters must obtain registration forms either in-person or online (not all states offer them online) and mail them in or physically hand them in at specific locations. Once the application is received, the voting clerk sends a voter card to the voter that notifies them which polling site they must vote at. When the voters go to vote at their assigned polling site they must present a valid ID (different states have different standards for what a valid ID is) or sign an affidavit in order to vote.

Once a valid ID is shown at the polling site, the voter is able to cast a ballot. There are many different ballot casting methods used in elections today including Optical Scan Paper Ballot Systems, Direct Recording Electronic Systems, Ballot Marking Devices and Systems, and Hand Counted Paper Ballots. Some of these are verifiable systems, however; many states do not run the verification tests due to lack of resources such as time, money, personnel. Others have no way to verify the results and protect against tampering.

In order to have a voting system which is both valid and trustworthy, a set of security requirements must be satisfied. The first is integrity: where votes are cast as intended by the voter, and votes are counted as casted. This ensures that the votes themselves are not changed, nor are people coerced or misled into voting incorrectly. The second is ballot security: where it is impossible to see how a voter voted (the weak condition) or it is impossible to prove how an individual voted (the strong condition). The third is voter authentication: where only registered voters can cast a vote, and a limited number of times. The fourth is enfranchisement: where all registered voters have the opportunity to cast a vote. The last requirement is availability: where accepting and counting votes is done in a timely manner.

These five stated conditions are the requirements for a correct voting system, and these are the conditions implementations of online voting have failed to meet [8, 2]. Specifically, they have failed to ensure integrity, ballot security, and voter authentication. Our solution to this ever-present and increasingly important problem is to combine the ways of the old and new; to implement voting machines operating with blockchains in traditional polling sites. This hybrid approach will be able to meet all safety five requirements and will be an improvement over the insecure and outdated polling technology of today.

Modified blockchain technology between voting machines in polling places during an election (whether it be in a precinct or state) will store the vote records as transactions in the ledger in a distributed fashion. This will ensure the integrity of votes cast and allow voters to be confident their voice was heard. In order to accomplish this system, it is necessary to develop a private blockchain. Private blockchains can use secure permissioned networks where only nodes with special codes can access. The government would set up this permissioned network similar to how companies such as Google or Amazon do for their workplaces. This will help the system ensure the honesty of nodes.

In order to satisfy the integrity, the system will need to ensure that every vote is correctly counted. This is accomplished in the system through using a ledger to keep track of transactions (votes). An eligible voter will gain access to cast a transaction by showing a valid form of identification at a polling station. Each polling station will be a network node which is able to propose blocks of transactions to the ledger. The ledger must then verify all transactions on a block are valid in order for the vote to be added to the ledger. Since this blockchain will be created by a trusted source, a specific protocol for validating transactions is able to be applied to each proposed block. Since the owner of the blockchain can determine which blocks are accepted onto the ledger, all blocks created within polling station nodes can eventually be accepted. This ensures every vote cast at a polling station is eventually added to the ledger where it will later be counted.

The system satisfies the security principle simply through not including identities of voters in transactions. The voters gain access to a transaction through showing valid identification at polling

place. As there is no record of the voter's identity kept, there is no adversary which could see who voted for who, even if the system is compromised.

The authentication principle is satisfied through requiring voters to come to polling places with identification to gain access to a voting machine. The machine only allows one transaction to be cast until a polling place worker enters code to allow another transaction to be placed. With this, voters can vote in complete privacy, but still will not be able to cast more than one transaction.

Enfranchisement is satisfied the same way it is currently. Voters have access to vote through traveling on election day to polling sites and verifying their identity.

The last requirement, availability, is satisfied as the blockchain is a widely distributed system. Each machine is a node in the blockchain, therefore if one machine fails in a polling place, voters can still use other machines to cast their votes. Later, it is shown that throughput and vote counting supports a busy real-life election day.

In order to develop a system which fits this description, we will create a private blockchain which is capable of supporting anonymous transactions and can guarantee eventually all transactions cast on honest nodes will be accurately counted. This requires us to develop a transaction verification protocol that will ensure that only one vote is cast per voter and the vote is counted both accurately and anonymously. This can be accomplished through developing the blockchain software and testing with very specific test cases.

4 Model

We have designed an updated model and protocol to replace what is currently done by states at local polling sites to securely improve the state of voting. This is all done at a state level, because states control their own voting protocols and it would be difficult to scale nationally. On a federal level, states still count and report their own votes; the electoral college in the US prevents simple counting of votes nationally. Within states, each precinct may have their own local elections on top of the state positions, but all transactions are still shared and counted state-wide on a singular blockchain.

4.1 Polling Protocol

The protocol at the polling site is mostly unchanged from before. The polling place is revamped to have a computer at the "check-in" desk, and at least five voting machine computers which voters will actually interact with; the polling place no longer requires any traditional paper ballots or privately owned voting machines.

Upon entry, voters check-in with the volunteers/employees, and the employees perform the usual manual verification of identification to ensure the person is a registered voter in the correct precinct. The difference is now that instead of being handed a large paper ballot, voters are now handed a unique (paper) QR code. This QR code is a single-use key or "ticket" for the voter to enter their ballot onto the voting machine. Before confirming a vote transaction, the voting machine confirms the ticket is valid and was generated by the corresponding machine at the polling site.

The QR code is generated by a machine operated by the employees at check-in; it runs a simple program that, upon request, generates a timestamp and signs it using public-private key (RSA) cryptography. The binary data of the message and its signature is transformed into a QR code and is printed. Private keys are unique to each polling place (each single code generator), and the corresponding public key is known by each voting machine at that location. The voting machine verifies the signature came from the code generating machine and verifies the timestamp is recent; this ensures that only voters who legally checked in are able to cast votes, and that voters aren't able to pass on their tickets to other (non-registered) voters throughout the day.

The voting machine itself is loaded with an empty ballot data structure, and the voter can interactively read and cast votes for each election on the ballot. Once the voter correctly casts all of their votes, they scan their unique QR code to the machine and the transaction is applied to the blockchain by the voting machine. The "public" key of the transaction is the unique QR code, and the data blob is the data structure of the filled-in blob. The machine processes a vote/transaction at most once per minute, as we expect voting to take no shorter than a minute. The machines all run continuously throughout the entire day of voting. It is also important to note that the QR code is not tied to the voter's identity in any way; even if the private key of the generator became public information after the election, attackers could not try to brute-force sign a voter's identity to see if a transactions's key matched the signature (as this would violate ballot security).

At the end of the voting day, all of the ballots can be aggregated and counted on one (or more) machine; any of the nodes in the system is able to count, and many should count in parallel to cross-validate. The counting program on the machines reads the entire blockchain and its ordered transactions. Votes for each election are aggregated, and at the end, total counts per precinct and state are output. The program counts votes cast from each unique QR code once; only the first ballot is read, solving the program of people trying to use the same

QR code to spam multiple votes together.

4.2 Machine Hardware

The new model is capable of running on commodity hardware, and does not require special hardware aside from (barcode) scanners. The few explicit requirements are a modern-day processor, and at least 50GB worth of stable storage to support the blockchain, which is expected to require tens of GB of storage for up to 10 million transactions. The machinery and exact implementation can be left up to the state; some states are wealthier than others and can afford to invest more in technology, while others aren't. Currently, large voting machines cost over $2,000 and states can afford one per about 300 registered voters [1]. We expect the hardware required for the new model to cost less than a quarter of that, as the software itself is open-source and the intricacies and quality of the machines themselves are up to the state and its developers.

In preparation for an election day, all of the machines in a state, precinct, and polling site must be initialized. One node must create the (private) blockchain, and all other machines must connect by sharing their public key with the node already in the chain. This individual permissioning is a tedious way to guarantee that no other third-party machine is able to read or append to the blockchain. This must be done ahead of time by a trusted party. All machines in a precinct must be loaded with the same ballot data structure, which contains information about state and local elections happening. Within a polling site, the QR code generator must generate a pair of RSA public/private keys and share its public key with the voting machines; this can be easily done by any administrator inplace at the polling site.

5 Implementation

We have created a simple implementation using open-source blockchain software and our own suite of Python and C++ code to model the workflow of the new voting protocol and test it.

5.1 Blockchain

Blockchains are decentralized, distributed ledgers that keep track of transactions across many computers to ensure the validity of transactions and their order. There are both public and private blockchains. Public blockchains allow any computer to join and act as a node, whereas private blockchains have control over which machines can join the network. Blockchain provides a secure way to keep track of transactions of any type by using a Byzantine Fault Tolerant protocol to protect against malicious nodes manipulating the data. Its decentralized protocol ensures that the ledger is not controlled by a single node and that every node has the ability to verify the validity of transactions and their source.

Multichain [5] is an open-sourced API that allows users to create blockchains and interact with them. It provides a feature called Data Streams, which allows users to append and read from the blockchain. The Data Stream abstraction allows users to easily use the blockchain as an append-only key-value store. Data Streams belong to a blockchain, which can have multiple data streams. The Data Stream abstraction ensures that all transactions are eventually posted to the blockchain and does not require a proof of work protocol. It also provides immutability, which ensures that data cannot be manipulated after it is posted to the blockchain.

Multichain uses permissions to control which nodes can perform different actions. Nodes can gain permissions to create, write to, and read from blockchains. There are also permissions specific to each data stream. The node that created the blockchain or data stream initially is the only one with any permissions and has the ability to grant other nodes permissions. The ability to grant other nodes permissions is called "admin" permissions. Any node with access to a blockchain or data stream can be given "admin" permissions from another "admin" node.

For our implementation of the blockchain, we used the default private blockchain parameters provided by Multichain for all the parameters except for three. We set "target-block-time" to two seconds in order to keep the transaction time fast, but not create too many forks in the chain, we set "mining-turnover" to 0.0, and we set "mining-diversity" to 1.0 so that each node would get to post new blocks to the chain at an equal rate. We specifically took advantage of the Data Stream "publish", "liststreamkeys", and "liststreamkeyitems" API endpoints for the implementation.

5.2 Voting Software

Our sample suite of software [6] contains a program to run a voting machine, a program to generate unique ID (QR) codes, a program to count ballots, a program to test the suite, and a library for cryptographically processing the unique QR codes. The voting program code is intended to run on a Linux machine where the user only has an interface to the program itself, and no other access on the machine. Alongside the Multichain blockchain software, the voting program is what is intended to run on each voting machine (node) in a polling place.

The voting program allows a user to interactively "vote" on a ballot by accepting user inputs for indi-

vidual elections. (In a real-life application, this interface would be left up to further UI development.) No identification is required to simply use the voting program. After the ballot is input correctly by the user, the voting machine program waits for and reads the QR code (in a hexdump format) and verifies its correctness. If the code is not properly signed or is outdated, the program rejects the ballot. If the code is verified, the program runs a subprocess to transact the ballot on the blockchain, or queues the job interally, depending on the frequency of votes. Both the voting program and the unique code generating program utilize the unique code library (which was implemented in C++).

We created a separate program to easily perform various tests on the system. The tesing program accepts various parameters for the timing and magnitude of the test, which can be used in conjunction with the size of the blockchain system to test various scenarios. The test performed is discussed in greater detail later. Futher details of the implementation can be discovered on the repository page previously cited.

5.3 Evaluation

Our protocol and implementation are evaluated in terms of practicality at a large scale, security concerns, and testing of the implementation.

5.4 Practicality

Currently, there are a few thousands of polling places within each state (although the number is declining in some states). We expect each polling place to have about five voting machines. For each state, we expect on the order of 10,000 voting machines, which is equivalent to 10,000 nodes on the blockchain. Anywhere between 1 and 10 million ballots are cast on (presidential) election days per state, depending on their size. Throughout a minimum of 8 hours of voting, 10 million votes are cast at a rate of approximately 350 votes per second spread out acorss all nodes. This is well within the Multichain throughput limit of 500 votes per second on the entire system. This throughput limit ensures transactions are posted on the blockchain without starvation or incessant forking. The voting machine program also guarantees ballots are cast at a uniform rate by scheduling transactions, regardless of how voters interact with it.

5.5 Security Concerns

Our blockchain-based voting system gives us a number of guarantees that other electronic voting systems failed to achieve, but it is still susceptible to a number of potential security issues. As stated previously, the use of blockchain assures the ledger will not be tampered with. Once a vote is added to the chain, we have assurance the data will never be changed at a future time. This takes away a major vulnerability of databases which could be hacked to create a fraudulent voting record, the blockchain removes the possibility of the issue.

This does not mean the system would be invulnerable as-is. There is still the possibility of a number of potential issues. A problem area could be the communication between the nodes after a ballot is cast. The use of modern encryption for messages broadcast between nodes protects an attacker from tampering with packets and changing votes but there are still potential issues. For example, an attacker could interfere with the connection between nodes utilizing a man-in-the-middle attack to stop messages from being sent between nodes or attempting to send fake transactions to interfere with the system. This is why security of the connections between the nodes is imperative, even with the various uses of encryption in the system.

Another area for potential vulnerabilities of the voting system is the security of both the machines themselves and the polling sites. Currently, there would have to some system, potentially a database, to keep track of eligible voters which would be checked when a voter enters a polling site with their identification. This itself could give a number of potential issues; databases can be hacked with false information or a voter could vote at different polling sites. This is an issue addressed by the polling sites themselves, as the state controls the method for voter authentication and the employees/volunteers used. In addition, there would have to be checks to ensure the validity of the physical machines. It would be possible for individual voting machines to be tampered with and have their code modified, albeit very difficult. This would involve a level of physical security in addition to trust with those who interact with the voting machines. Overall, the proposed voting system delivers certain guarantees to election security, but there are other potential issues where further work and research is necessary.

5.6 Testing

In order to test our voting protool proposal and the developed software, we performed a hypothetical blockchain-based election. Based on the hardware resources we had access to (and our unfortunate limitations), we set up ten Virtual Machines (VMs), each representing a voting machine at a polling site, on one of the department servers, that is, *bowser.eecs.umich.edu*. Each VM is allocated one CPU core and 1.5 GB of memory. We configured a virtual network adapter on the server so that VMs could connect to each other using a virtual network in which a specific static IP is assigned to each VM.

We initialized a private blockchain on the VMs using Multichain, created a ledger-based database (a sample of Multichain data streams), and started the voting procedure using the testing program. The voting procedure consists of repeatedly generating a unique code (which would be realized by the unique printed QR codes in real polling sites), randomly filling out a ballot template, and casting the ballot by publishing a transaction to the data stream. The VMs cast one ballot every 5 seconds (12 ballots per minute), which is higher than the rate in polling sites having even five voting machines. Each VM cast 5,000 ballots in total; therefore, the election lasted about 7 hours with an aggregate of 50,000 ballots, with each VM taking identical time to run.

At the end of the hypothetical election, each VM counted the votes on its blockchain, and the integrity of the blockchain (i.e., having no forks) was verified. Each VM also recorded the votes it cast individually during the test; we verified the transactions on the blockchain reflected the aggregate sums on each VM (with Thomas "Tom" Cat winning our mock election). Counting 50,000 ballots on the blockchain sequentially took a (slow, single-threaded) process on the VM approximately 8 minutes to do, showing an entire election can be counted within a day.

6 Conclusion

We have proposed a series of changes to current models and protocols of voting to potentially revamp how modern large-scale elections are held. When controlled at a state level, blockchain technology is able to provide a durable and secure ledger for ballots cast by registered voters, when following a simple updated in-person protocol for voting. The technologically-driven approach combined with the current voter authentication practices helps ensure security and is not prone to issues faced by other electronic or online voting systems. Our sample implementation of voting software using blockchain shows the practicality of the new voting model and is evaluated as being functionally correct, supporting enough throughput to handle real-life scenarios, and solving common security issues. We encourage further research and development into blockchain-based voting systems to potentially revitalize outdated and expensive voting protocols at a state level, nationally, and potentially globally.

References

[1] "Aging Voting Machines Cost Local, State Governments," http://www.pewtrusts.org/en/research-and-analysis/blogs/stateline/2016/03/02/aging-voting-machines-cost-local-state-governments.

[2] "Hursty Hack," https://en.wikipedia.org/wiki/Hursti_Hack.

[3] "Independent Report on E-voting in Estonia," https://estoniaevoting.org/.

[4] "Internet Voting," https://www.verifiedvoting.org/resources/internet-voting/.

[5] "Multichain," https://www.multichain.com/.

[6] "Project Repository," https://github.com/tiberiuvs/eecs591research.

[7] "US Voting Machines Hacking," https://www.theregister.co.uk/2017/07/29/us_voting_machines_hacking/.

[8] A. J. Feldman, J. A. Halderman, and E. W. Felten, "Security analysis of the diebold accuvote-ts voting machine," in *USENIX/ACCURATE EVT Workshop*, 2007.