Blockchain Voting Machines: An Approach to Election Security

Zane Hollandsworth

A Senior Thesis submitted in partial fulfillment
of the requirements for graduation
in the Honors Program
Liberty University
Spring 2024

Acceptance of Senior Honors Thesis

This Senior Honors Thesis is accepted in partial fulfillment of the requirements for graduation from the Honors Program of Liberty University.

____
Mike Kipreos, D.B.A.
Thesis Chair

____
Melesa Poole, Ph.D.
Committee Member

____
Cindy Goodrich, Ed.D.
Assistant Honors Director

____
Date

Abstract

Blockchain is a popular technology that has found itself as the proposed solution to a variety of technological problems. In the case of voting machines, blockchain seems to show promise based on its unique attributes of immutability and decentralization. Reviewing the state of current voting machines, there is a displayed lack of support for direct-recording electronic (DRE) voting machines in the general population due to the many controversies. By looking into the functionality of proposed blockchain systems and creating an implementation to test the functionality of a blockchain voting system while taking the data on voting machines into consideration, it can be concluded that the best solution would be to combine scanned paper ballots with a blockchain system.

Blockchain Voting Machines: An Approach to Election Security

Election security is a topic that has become one of the forefront issues over recent election cycles. The question of whether received votes are trustworthy has led to significant division in the political sector as calls for voting reform are made. While many of the present issues may revolve around important questions over which individuals should be allowed to vote and when, there are a multitude of other questions related to the security of the voting process itself. These concerns range from threats of foreign interference, casting multiple votes, and potentially even changing the contents of a ballot after it has already been cast. There have been a variety of different proposals on a reliable system that can strengthen the security of these devices, and one such proposal that has become more widely discussed in recent years is the use of blockchain technology (Zeng, He, Yiu, & Huang, 2023).

As the modern era progresses, it is important for systems critical to governance, namely voting systems, to improve and keep on pace with the advances of technology. Blockchain is an innovative technology that could potentially provide several key benefits in the application of voting systems. The major feature of decentralization allows it to avoid potential security risks in systems that are controlled by a main server, as each system holds a record of entries made. Another major feature that increases security is the attribute of being unable to be changed without a major intervention in the network. This would be impossible to do covertly, as it would require a consensus from the network. The transparency of this network lends itself to being easy to audit as well. When counting votes and verifying that they took place at a valid time, the blockchain serves as a good way to securely track the votes being cast while simultaneously maintaining the privacy of those casting votes through secure hashing algorithms. Finally, the fact that the blockchain is fully electronic and does not require a physical count would significantly increase the speed and reliability of the process of tallying votes received. By making use of these features, it is proposed in this thesis that blockchain technology should be utilized as part of voting machines, and a prototype program to demonstrate this is described.

Current Election Systems

Overview of Voting Systems

Before immediately looking into the matter of how blockchain can be implemented to improve the overall security of an election system, it would first be beneficial to examine the strengths and weaknesses of the current election system. Traditionally, voting was done by stamping a ballot, folding it in a prescribed fashion, and placing it in a box to be counted later (Begum & Kumar, 2012). This method presents many challenges in addressing security concerns. While it does well to ensure the anonymity of the voter, it fails to prevent some of the most important security issues. Namely, these are actions such as casting multiple votes at once in an act of ballot stuffing or an intentionally false count being taken by malicious actors working at a voting center (Begum & Kumar, 2012).

With the improvement of technology and the rise of computer systems, it was possible to create a significantly faster and easier to use system with less room for error. Voting machines to fill this need can come in a variety of formats, however (Begum & Kumar, 2012). These can have varying levels of security and means of input. The broadest of the differences between types of voting machines can typically be found in their means of input. Some of these machines function by reading in a filled in ballot sheet, some use touch screens to navigate input, and others may simply have a button for each candidate (Begum & Kumar, 2012). One important security matter is the possibility that a voting machine may utilize biometric means to verify the identity of voters. This would effectively solve any concerns over the possibility that a duplicate or unregistered vote could occur, but it may be seen in some cases as too invasive of personal privacy, as it would require permitting the government entity running the poll to have access to this sort of biometric information.

Efficacy of Voting Systems

It has been the case that voting systems are often examined based on the method of casting the ballot, but it has been shown to be the case that all types of ballot casting methods have had similar effectivity with different levels of preference by voters based on the type of system used (Everett, et al., 2008). While the older voting systems of mechanical levers and punch cards are now entirely absent in general elections, both paper ballots and direct-recording electronic voting systems, or DRE, are used (Thompson, 2008). An important factor to consider is that electronic voting has had major technological failures in the past. During the early and mid 2000’s, voting machine errors sparked intense debate over how voting should be conducted. The hotly debated Florida election in 2000, caused by punch cards that were not fully punched through, resulted in counting errors and arguments over the intended vote of citizens that ultimately led to an extremely narrow victory for George W. Bush that was plagued by legal arguments (Thompson, 2008). Questionable voting races have even led some states, namely California and Florida, to completely abandon electronic voting altogether. These states still make use of paper ballots today, although they are now tabulated by machines in the same manner as most states.

Although one may think that it would be the case that fully electronic voting would have continued to expand, their use has declined in favor of paper ballots counted by a scanner due to the controversies associated with fully electronic voting machines (MIT, 2023). This trend has only increased after the controversies related to election fraud in the 2020 federal election.

Furthermore, usage of DRE systems is often debated due to accessibility for those unfamiliar with the operation of a computer system, such as some elderly voters. These accessibility concerns have led DRE systems to have a significantly higher rate of error when voting: 4.2% on DRE systems as opposed to 1.5% on paper ballots with 2.5% of voters choosing the incorrect candidate (Everett, et al., 2008).

Considering these factors, it is no surprise that voting systems have gradually moved back toward paper ballots. In fact, some rural areas still primarily use hand-counted ballots, despite their proclivity to human error impacting results (MIT, 2023). However, regardless of whether there is a preference for paper or fully electronic DRE systems, as long as the vote is counted by a computer, it can benefit from the usage of blockchain voting systems. In fact, for the sake of security, the existence of a paper ballot to accompany the electronic vote is beneficial. In the case that there is any question as to the validity of the vote, the possibility to retrieve the original ballot improves the ability for anyone with questions to determine that proper steps were followed, ensuring transparency in the process.

Storage of Votes

It is likewise important to consider the means by which vote information is stored. In traditional hand-counted voting, the paper ballots are tallied one by one, and a total is created. Likewise, scanned paper ballots follow the same premise but are scanned by a machine either at the time of casting the vote or later at a central location, although being scanned later on opens up more opportunity for tamping at the ballot level, as they will require handling before they can be scanned. DRE systems can vary in how they store their information, but in the modern day, they typically are comprised of an internal count with a printout similar to a receipt that creates a paper record of the voting event occurring (MIT, 2023). This provides the ability to examine the results and manually recount, if necessary, based on the paper trail created by the DRE machine. This can also be used to allow the voter to check that their vote was properly received.

While most systems approach the security of their systems by remaining completely offline and isolated, some systems have utilized the Internet to transport voter counts. Typically, it has been considered safer to remain offline as it would require physical tampering with the machine to cause the vote count to be altered. This would likely be easy to spot if occurring, and each instance of tampering would have to be done individually. When votes are transferred over the Internet, however, there is the possibility of bad actors managing to alter this during transit. In the modern day, this is less likely to be possible if proper precautions are taken when designing the system used through the usage of encryption and other methods of validating the legitimacy of Internet communications.

Blockchain for Voting Systems

Overview of Blockchain

Blockchain is a technology that has risen to prominence over the past decade. It was originally introduced in 2008 with the cryptocurrency Bitcoin for the purpose of creating a currency not governed by any one individual (Pawlak, Guziur, & Poniszewska-Marańda, 2018). Although it stayed relatively unnoticed for nearly a decade, 2017 and 2018 saw an immense rise in the popularity of cryptocurrencies like Bitcoin, and by extension, blockchain technology as a whole (Gorkhali, Li, & Shrestha, 2020).

The premise behind blockchain technology itself is quite simple: blocks of data consisting of several individual attributes are chained together by an attribute that uniquely identifies the block and an attribute that identifies the block that came before it (Gorkhali, Li, & Shrestha, 2020). This provides a link innate to each block that can identify its spot within the blockchain, and it can be further used to verify that each block is aware of the previous block, allowing traceability and verification all the way back to the beginning of the blockchain. Blocks are also not allowed to be modified once they have been created and the data contained within is set. Additionally, blockchains are typically characterized by decentralization in which members of the blockchain each keep their own copy of blockchain data that must be coordinated with others on the blockchain network (Gorkhali, Li, & Shrestha, 2020).

Security of Blockchain

The security of blockchain is one of the defining features that makes blockchain an appealing choice for those considering its use. Bhutta (2021) goes over several different basic features that contribute to the security of a blockchain. Some blockchains may be configured to only allow certain individuals to join, thereby preventing unwanted intrusions. In other cases, it is beneficial to allow anyone to join, and the extremely decentralized nature of the blockchain due to these members promotes security through democracy, preventing bad actors from having too much influence over the state of the network. Additionally, peer-to-peer communication is typically preferred in order to avoid the risks associated with a centralized server having control over data, which can result in situations where one compromised server sends illegitimate data to all members of the blockchain, changing it in an unintended manner. One more important aspect of many blockchains that Bhutta addresses is the usage of encryption keys. These are used as part of a cipher to allow users placing data onto the blockchain to hide the contents of the data used. This is beneficial as it is unnecessary for the blockchain to have access to the unencrypted data, and as blockchain data can be viewed by anyone, those who would benefit from learning the details of what was present are unable to view the contents without the decryption key held by the owner.

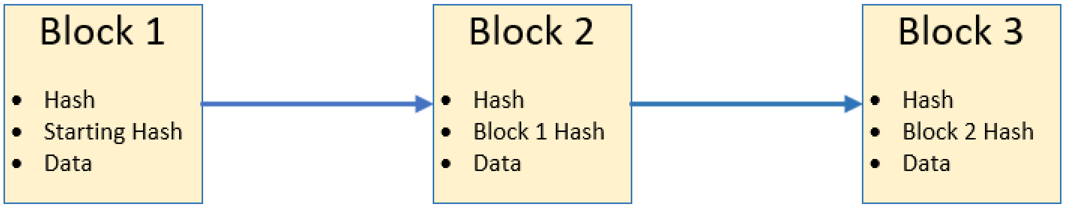
Hashing Algorithms. Hashing algorithms are a useful class of algorithm that are capable of taking in an input of an unspecified length and convert it into an output of fixed length, called a hash (Chaves, et al., 2016). These can serve a variety of purposes, ranging from authentication to data integrity or even random number generation. In general, most hashing applications make use of the fact that any given input will provide a unique output. Factors that differentiate hashing algorithms often include the length of the output and likelihood of two different inputs resulting in the same output, causing a hash collision.

To expand, it is helpful to come to a better understanding of how a hashing algorithm may function. Many modern hashing algorithms provide a large output, such as a 512 bit output in the case of SHA-512. For a simple example, the mathematical operation of modulo, the operation which returns the remainder in division as the answer, is often used to demonstrate. As all data in a computer is in the form of a number, hashing algorithms work by operating on numbers. In this case, two arbitrary numbers may suffice. If a hashing algorithm was defined as modulo 7, it would be capable of taking an input number of any length and return an output that consists of one number from 0 to 6. If one were to desire to hash the number 43, it can be seen that the number 1 is returned as the remainder of 43 divided by 7. This can be used as a means of data validation, as the number 44 will not yield the same result; it will instead provide the number 2 as a result. There is, however, the potential for a hashing collision in which the same number is returned for multiple inputs. If 50 were used in this scenario, it would also result in a hash of 1, potentially falsely reporting that the same number is present if the task was to determine whether a number was duplicated. However, modern hashing algorithms take great care to ensure that hashing is done in a manner that results in hash collisions being exceedingly rare, making them suitable candidates for validating that data being checked is the same as was used to generate the hash the first time (Chaves, et al., 2016).

Hashing in Blockchain. One of the most important aspects of the blockchain is its usage of hashing data. By executing a hashing algorithm on some arbitrary data, a unique set of data can be created that can be used to validate that data has not been changed. This is particularly beneficial in blockchain as it serves as the method by which blocks can create a unique identifier. By hashing all of the data contained within each block, a hash is created that not only serves as an identifier for other blocks, but it also ensures that blocks are able to maintain the necessary property of being immutable once set (Gorkhali, Li, & Shrestha, 2020). Unlike other systems, blockchain does not need to be especially concerned with safeguarding the ability to change previous data to a high degree; the hashing algorithm and decentralization will serve to validate that data has not been changed. This is shown in an illustration of the basic structure of a blockchain in Figure 1.

Figure 1

Structure of a Blockchain



Note. The hash in each blockchain is created using both the data and hash of the previous block.

Due to the combination of hashing and decentralization, blockchains are very resistant to changes being made to older data. By committing the act of tampering with a blockchain by changing any value present within an older block, the blockchain's hashes will be completely different from that point forward (Jafar, Aziz, Shukur, & Hussain, 2021). This will result in any attempt to validate the integrity of the blockchain failing at that point, as when comparing the next block's previous hash to a new hash of the block, it will generate a differing value. Even in the case that a malicious actor was able to make one of the hosts on the blockchain recalculate all hashes, there would still be the issue that the other instances of the blockchain contain different data, so even if the altered host was able to validate itself successfully, none of the others would accept it, thereby protecting the integrity of the blockchain through decentralization.

Another aspect of major blockchains is the presence of a consensus protocol. A consensus protocol is used to prove that an event on the blockchain network took place and was valid. Consensus can be reached in several different ways with the two most popular on public blockchains being proof of work and proof of stake utilizing practical Byzantine fault tolerance (Gorkhali, Li, & Shrestha, 2020; Liu, Liu, Xu, & Wang, 2023). These systems operate based on the principle that actions on the blockchain need a trustworthy source to perform validation. Proof of work and proof of stake operate in a manner that incentivizes participation in the process of validation by providing a reward in the form of cryptocurrency on the blockchain. This helps to prevent the possibility of attacks that come in the form of overwhelming the network with malicious members that attempt to validate an invalid event on the blockchain (Liu, Liu, Xu, & Wang, 2023). If successful, attackers could manipulate what data is present on the blockchain; in this case, it could theoretically be possible for attackers to change voting data present on the blockchain, though this would require attackers to be present on the voting blockchain in a capacity that allows for being a part of the validation process.

Blockchains may also contain smart contracts. According to Sayeed (2020), these are simply programs capable of residing on a blockchain that are capable of automatically performing their designated task. These generally hinge on the premise that payment for something is performed after certain conditions have been reached, resulting in the execution of the smart contract. An important aspect to note about smart contracts is that because they reside on the blockchain, they cannot be modified at any point after they have been added to it. This necessitates that developers working on smart contract programs should be prudent to carefully address any possible errors or security concerns so that the program will not be rendered either useless or a liability in the future (Sayeed, Marco-Gisbert, & Caira, 2020).

Blockchain and Voting

With the background of voting systems and blockchain technology addressed, it is evident that voting systems have the potential to be integrated with blockchain technology in a way that improves upon current systems. The append-only and decentralized nature of nature of blockchain shows promise in how this could benefit the systems being used at this moment (Huang, et al., 2022). By effectively understanding the strengths and weaknesses of the current voting system, it should be possible to successfully integrate this with blockchain technology to produce a more robust and secure voting system that leaves significantly less room for foul play to take place in the voting process.

With that in mind, it is necessary to determine exactly which aspects of voting systems blockchain can solve. According to Jafar (2022), some of the most important issues facing electronic voting that need to be solved are the integrity of data, the dependability of the voting machines, the transparency of the process, the ability to keep ballots from identifying persons, the security of the process, the ability to deal with fraud, the repercussions of failure, the lack of knowledge present in voters, the ability to store related equipment, and the cost of the systems. Blockchain has the potential to solve many of these issues thanks to its overall secure nature. Most notably, blockchain has the potential to address data integrity, transparency, and security. However, Jafar goes on to note that it is still difficult to fully draw out the potential of blockchain in this application and that there are new obstacles introduced associated with blockchain, that being the scalability, speed, and privacy.

Security Concerns

Unfortunately, there are some concerns that lie beyond the scope that blockchain alone can address. Park (2021) highlights some of the major flaws that could be associated with blockchain voting. One major aspect that is brought up is that blockchain voting may be heavily associated with bringing about Internet voting. While the ability to vote anywhere that an Internet connection is available would be convenient, it is unfortunately the case that there are significant security risks involved with this. Firstly, there is the possibility that users are compromised and have their vote manipulated or ignored entirely. While it is certainly possible to require many things to be validated to cast a vote on a voting website, it is impossible to maintain security across all devices when there is such a wide range of different operating systems, web browsers, and other variations in the potential pool of people voting. This also creates another accessibility concern for areas that have limited access to Internet connections and people unfamiliar with using technology, much in the way that it is a concern for DRE voting systems.

Additionally, by utilizing an Internet-based approach to voting, it may invite attacks on the system. There are a wide range of entities that have an interest in disrupting or altering the outcome of elections in the United States, and losing connectivity to vote would be extremely damaging in an election. Park (2021) goes on to assert that utilizing paper ballots is the safer approach. Although the paper is quite harsh on blockchain, it acknowledges that while it tackles the issue in the way that most proposals have, stating that blockchain is a subcategory of Internet voting, it also brings up the fact that the blockchain could be used to store the result of a paper ballot (Park, Specter, & Rivest, 2021). Considering the pros and cons of each of the voting systems up until now, it certainly does seem to be the best solution out of those that have been identified. Paper ballots are the most reliable in the areas of user error, machine error, and ease of auditing. By combining the current best solution with a strong implementation of blockchain, it should be possible to bring out the best of both, resulting in a stronger and more secure election system resistant to the current issues of concern in voting systems.

Features of Blockchain Voting

Taking all of this into consideration, the next logical step is to determine guidelines that an implementation would reasonably follow to qualify as a secure blockchain voting system. Yang (2021) presents several security requirements: only authorized voters can cast ballots, only one vote can be cast per voter, votes cannot be modified or otherwise recast, only verified votes will appear in the total sum, and nothing is presented to the voter that would allow them to prove to another individual how a vote was cast. This proposed voting system is set for online voting to take place, presuming that a voter would register by uploading identifying information to the voting website on the Internet. The author goes on to state that each vote's content should be encrypted after submission to prevent people's votes from being identified. Additionally, in this blockchain, the voter performs the proof of work to verify the validity of the vote without the need for any others. In this approach, it is also impossible to count the results early due to the names of the candidates being encrypted until the decryption keys are revealed after the election has ended. This applies to even the final voter, as everything is encrypted and unable to be deciphered until the necessary key is revealed. While the original plan for the system was to also include the ability to have no identifying information afterwards, the necessity to keep a private key for decryption ultimately resulted in this not being entirely possible.

Blockchain Voting Implementation

Decisions on Functionality

Having considered a wide range of different possibilities for how a system for blockchain voting could be done, considering the constraints, it was ultimately decided to do a blockchain implementation in Java. This language was chosen in part due to its familiarity as well as its portability across a wide array of operating systems and configurations. On another note, one major factor in the language itself is the ease with which objects can be created and manipulated within the language. When dealing with blockchain, it is beneficial to be able to quicky add more objects to the chain. In addition to choosing Java as the language, it was also necessary to utilize the dependency manager Maven to properly import external libraries that needed to be used.

Having set up the project, it was necessary to decide on a plan for the local implementation of the blockchain voting system. After reviewing the data on the matter, it was decided not to pursue creating a web interface for a system that would presumably be run locally. While a GUI was considered due to the ease with which it would for a user to enter data and otherwise interact with the program, it was determined that for testing purposes, the systems being used would likely only have the opportunity for a text-based console. Therefore, that is what was chosen as the means of interaction for the program.

Several libraries had to be decided on for the function of several key features. Because of the ease of conversion and storage, JSON was to be used for several key functions. The library that was eventually chosen for the task of handling JSON was Gson. This library is implemented by Google and allows for almost everything to be converted to and from JSON. This is especially convenient for scenarios such as wanting to write data to a file. This would normally involve a process of ensuring that the desired class and all classes used within implement the Serializable interface. By instead converting to JSON, it’s possible to save even static objects that would ordinarily not be able to be saved in this manner. Loading is nearly as simple as saving, as it merely involves reading the data in and calling the opposite function while specifying what class it should go back to.

A second library that was utilized was Jgossip, a Java implementation of a gossip protocol. A gossip protocol is a protocol that relays information generated and received to all neighboring hosts (Saldamli, Upadhyay, Jadhav, Shrishrimal, & Patil, 2022). This is beneficial because it is not necessary for a host to know of every other host at the same time, only a few nearby ones. While it may not have been entirely necessary in this small scale blockchain implementation, the usage of a gossip protocol makes it so that it would be very easy to expand into a larger network if desired; the only thing preventing that is the number of hosts available to use.

Layout of Data

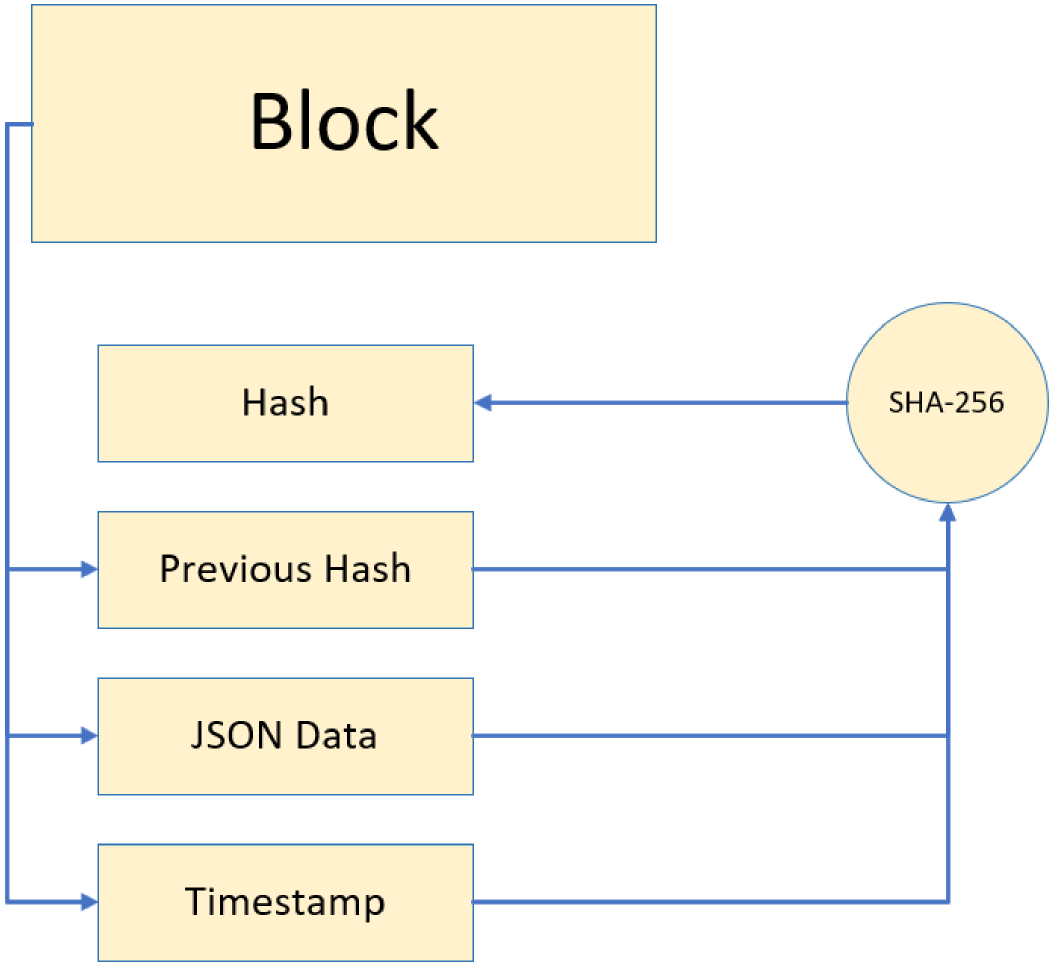
With those determined, the next question to be addressed is exactly how the data is laid out. In this case, a combination of measures taken for security and unchanged data for ease of visualizing the data was taken. To begin, the most basic element of this system, the block itself, consisted of four member variables. It was made up of the block data in a string format, a timestamp, the hash of the block, and the hash of the previous block. By leaving the block itself so open ended, the potential to successfully reuse the blockchain for any other application is available. By effectively making use of JSON as the block data, a simple block like this can be effectively utilized for voting, or almost any other desired data.

The chain is just as simple, as well. It consists of a variable to store the length the chain, and the chain itself is an ArrayList of blocks. With these two classes, a basic blockchain can be achieved. Of course, they both contain a variety of methods to help ensure data integrity and the proper construction of a blockchain, as well. The chain class simply has some methods to complete tasks such as retrieving a particular block, creating a new chain, printing the chain, and adding a new block. The block class contains methods that do similar tasks, namely retrieving the various member elements, adding a new block, and generating the first block, but many of these methods are more complex. One thing to note is that a blockchain needs to start with a set first block to ensure consistent data going forward. This can be done by creating what is known as a “genesis block,” or otherwise just a block containing some arbitrary data used to align a blockchain at the beginning, as the first real block will need a proper previous hash to utilize.

When a new block is created, it starts off being given all its data besides its hash. This data can simply be assigned to the block at creation. It is then necessary to run a hashing algorithm, in this case, the standard Java implementation of the SHA-256 hashing algorithm. This algorithm is a member of the SHA-2 hashing algorithm family, made up of algorithms that can take in a large amount of data and provide a set length output (Abed, Jaffal, Mohd, & Al-Shayeji, 2021). Algorithms such as this one are key to the viability of blockchains, as they are what keep them secure. If even one byte of the hashed data in the block changes, the algorithm will produce a different result, demonstrating that data was altered from its original state. In this case, the SHA-256 algorithm is performed on the timestamp, previous hash string, and the block data put together. This entirely covers the useful data within the blockchain, and it cannot be discreetly changed. A diagram showing the setup of a single block is present in Figure 2.

Figure 2

Visualization of a Block



Note. The Block directly sets its members, the previous hash, data, and timestamp from the initial given data, but the hash must be set through the use of the hashing algorithm, SHA-256.

Having covered the implementation of the blockchain itself, the voter data are also particularly important to establishing an effective blockchain voting system. In this case, there is a distinct class for voters and for votes, though they are similar. The data that was chosen to be used by the voter is their first name, last name, and social security number. Of course, all this data could be exchanged for any other data in a different application. In fact, it may be beneficial in a real system to provide a voter with a secret voter number that could be used to prevent malicious characters from discovering who a candidate voted for by performing the proper hash and getting their data. In any case, the data used in this implementation still results in a system that does not store any voter details despite that voter information being needed to validate a vote.

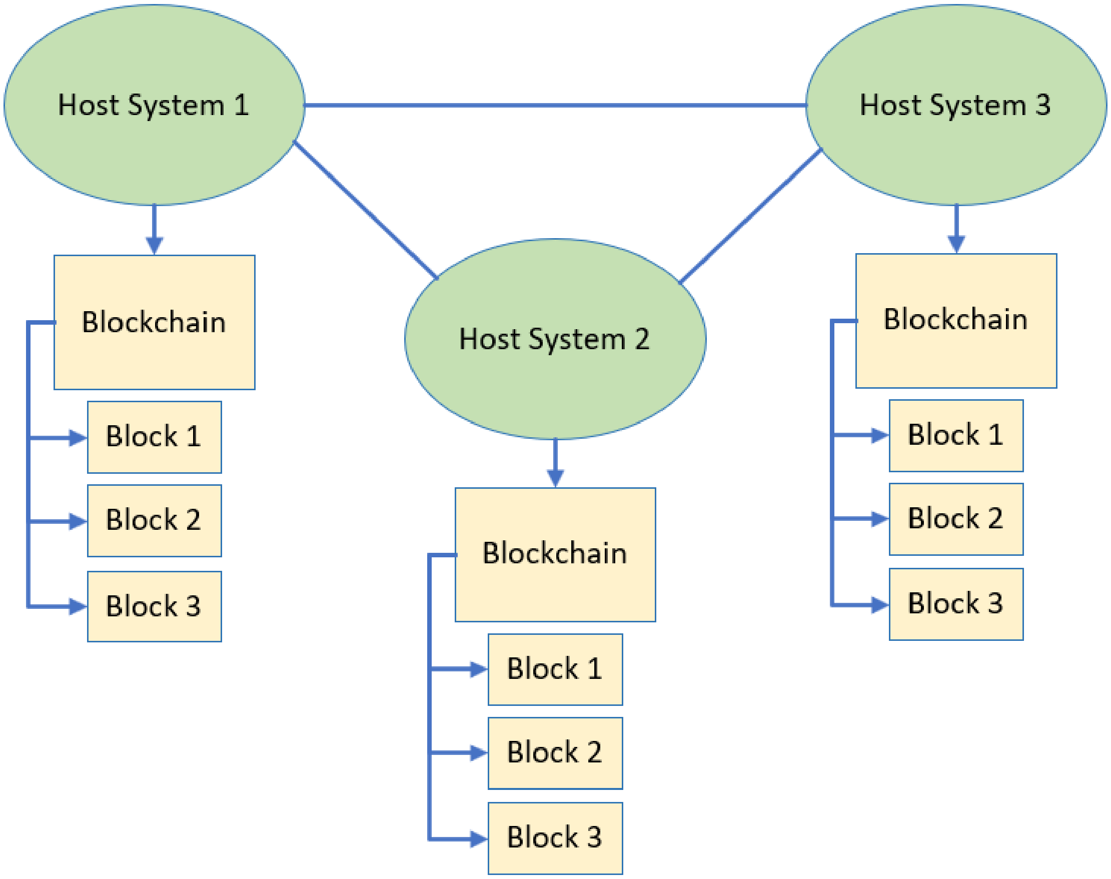
When a new voter is being added, these values are used, the SHA-256 hash is generated using the values given by the voter, and the voter object is given to Gson to be converted into JSON. This JSON is then used as the block data for a new block that is generated and added onto the blockchain. Whenever an individual vote or voter needs to be retrieved, the block can be acquired, and the JSON can be parsed back into a Java object for a voter or vote, thereby allowing the blockchain to cleanly store arbitrary data.

The process for votes is much the same as that of the voter. The only difference is that votes have an extra attribute for the chosen candidate. This could also have been hashed to hide its value in the data, but it was deemed unnecessary in light of being able to properly confirm that all data is being stored as intended. It contains the same hash as the voter, that being a hash of a first name, last name, and social security number. When a new vote is added, the corresponding voter can be searched for in the blockchain. If it is not present, the voter has not been registered and the vote should not be counted. If it is present as a voter and as a vote once already, then it is a duplicate and should not be counted. This effectively halts the ability for the voting system to be abused in the traditional manner of casting invalid votes.

The last major aspect of this system is the decentralized nature of it. Whenever one of the potential blockchain events happens, such as the addition of a new voter or vote, if the addition is valid, it will be sent out on the gossip protocol to any available hosts. These hosts will then be able to read in the JSON data, determine the type based on the presence of a candidate field, and add it as the appropriate type onto the blockchain. By doing this, any machines running the blockchain on the gossip protocol will be able to stay up to date. If one of them is tampered with, the other machines will still contain the proper information and can be used to restore a modified blockchain back to the original sequence. A diagram of a blockchain network functioning under a gossip protocol is shown in Figure 3.

Figure 3

Communication between Blockchain Hosts



Note. Each system transfers the same data, resulting in identical blockchains across each system.

Additionally, there are a few validation commands. The status command currently prints out block data, and the validate command performs a hash comparison to ensure that no blocks have been changed. It functions by beginning near the start of the blockchain and comparing a block’s previous hash to the result of running the SHA-256 hash function against the block in question. This is done for all blocks and will definitively tell whether anything has been modified. For good measure, because the program saves the blockchain as JSON to a file after finishing, the file sizes can be compared, and they should match exactly.

Implications for Security

While this program is quite rudimentary, and a fully-fledged program would involve a full team of programmers as well as cybersecurity experts, it seems to be evident that utilizing blockchain for voting would be able to improve election security at the time of and after the vote has been cast. As brought up before, the best method for secure voting would be to combine scanned paper ballots with blockchain, creating an overall more secure system with the best of both (Park, Specter, & Rivest, 2021). Additionally, it would be beneficial for the blockchain holding votes to stay private, not communicating at all with other devices on the Internet. This could be achieved through strict firewall rules that do not allow for connections from unauthorized sources. Finally, it may even be beneficial for a device running a blockchain program to be on a system that can have its memory flashed as read only, effectively preventing tampering with the program itself. In the future, it would be beneficial to verify the effectiveness of these security measures by performing a study on this specific matter. This could be achieved by performing a mock election and making attempts to fraudulently impact results, thereby testing the strengths and weaknesses of the system.

Conclusion

Voting systems have long been overdue for an update. Although it has been seen that the technology used in them has been updated to modern times with varying degrees of success, there has been a distinct trend towards paper ballots over technologies such as direct-recording electronic voting systems (MIT, 2023). The reasons for this are many, but much of it comes down to failures at a critical time that have eroded the public’s trust in these systems.

Blockchain is a dynamic system for storing information that is capable of resisting tampering attempts to a very high degree. Being made up of blocks identified through cryptographic hashes, these systems have come to prominence and have slowly been becoming more well-known by the general public regardless of whether that perception is due to it being considered the next tech buzzword or innovation.

In the implementation overviewed here, it is clear that these systems can be quite powerful while still being understandable. Each aspect of the technology is able to be built on in much the same way as blocks: built on top of one another to create a full structure. With dynamic systems like JSON and gossip protocol, it is possible for even a simple system like this one to demonstrate the abilities of blockchain.

Blockchain for voting systems is a promising premise that may very well see usage in the United States one day. While there are obstacles to overcome, voters need a sense of security in that their elections are not rigged against them or malfunctioning in such a way that it impacts the result of their vote or the outcome of the entire election. Blockchain has the potential to integrate with existing voting systems and increase their security, such as scanned paper ballots registering their count onto blockchain. Given the opportunity, blockchain could present itself to be a valuable asset to voting systems around the country.

References

Abed, S., Jaffal, R., Mohd, B. J., & Al-Shayeji, M. (2021). An analysis and evaluation of lightweight hash functions for blockchain-based IoT devices. *Cluster Computing*, 3065-3084.

Begum, T. U., & Kumar, D. A. (2012). Electronic voting machine — A review. *InternatiConference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)* (pp. 41-48). Salem, India: IEEE. doi:10.1109/ICPRIME.2012.6208285

Bhutta, M. N., Khwaja, A. A., Ahmad, H. F., Khan, M. K., Hanif, M. A., Song, H., . . . Cao, Y. (2021). A Survey on Blockchain Technology: Evolution, Architecture and Security. *IEEE Access, 9*, 61048-61073.

Chaves, R., Sousa, L., Sklavos, N., Fournaris, A. P., Kalogeridou, G., Kitsos, P., & Sheikh, F. (2016). Secure Hashing: SHA-1, SHA-2, and SHA-3. *Circuits and Systems for Security and Privacy*, 105-132.

Donepudi, S., & Reddy, K. T. (2021). Blockchain oriented hyperledger based performance driven framework for mass e-voting. *Intelligent Decision Technologies*, 579-589.

Everett, S. P., Greene, K. K., Byrne, M. D., Wallach, D. S., Derr, K., Sandler, D., & Torous, T. (2008). Electronic Voting Machines versus Traditional Methods: Improved Preference, Similar Performance. (pp. 883–892). Florence, Italy: Association for Computing Machinery. doi:10.1145/1357054.1357195

Gorkhali, A., Li, L., & Shrestha, A. (2020). Blockchain: a literature review. *Journal of Management Analytics*, 321-343. doi:10.1080/23270012.2020.1801529

Huang, J., He, D., Obaidat, M. S., Vijayakumar, P., Luo, M., & Kim-Kwang, R. C. (2022). The Application of the Blockchain Technology in Voting Systems: A Review. *ACM Computing Surveys*, 1-28.

Jafar, U., Aziz, M. J., Shukur, Z., & Hussain, H. A. (2021). Blockchain for Electronic Voting System-Review and Open Research Challenges. *Sensors*.

JG, S., SJ, M., & JW, J. (2021). A Scalable Implementation of Anonymous Voting over Ethereum Blockchain. *Sensors*.

Jumaa, M. H., & Shakir, A. C. (2022). Iraqi E-Voting System Based on Smart Contract Using Private Blockchain Technology. *Informatica*, 87-94.

Khairkar, S., Yeole, A., Pawar, A., Londhe, S., & Warpe, S. (2023). Secured and Transparent Voting System using Blockchain Technology. *Grenze International Journal of Engineering & Technology*, 34-38.

Khan, K. M., Arshad, J., & Khan, M. M. (2020). Investigating performance constraints for blockchain based secure e-voting system. *Future Generation Computer Systems*, 13-26.

Liu, S., Liu, C., Xu, C., & Wang, J. (2023). An improved PBFT consensus algorithm based on grouping and credit grading. *Sci Rep*. doi:10.1038/s41598-023-28856-x

Ma, X., Zhou, J., Yang, X., & Liu, G. (2020). A Blockchain Voting System Based on the Feedback Mechanism and Wilson Score. *Information*, 552-565.

MIT. (2023, April 21). *https://electionlab.mit.edu/research/voting-technology*. Retrieved from MIT Election Data + Science Lab: https://electionlab.mit.edu/research/voting-technology

Monrat, A. A., Schelén, O., & Andersson, K. (2019). A Survey of Blockchain From the Perspectives of Applications, Challenges, and Opportunities. *IEEE Access, 7*, 117134-117151.

N., S., & N., R. (2022). Smart Electronic Voting Device using Blockchain Technology. *Grenze International Journal of Engineering & Technology*, 457-462.

Park, S., Specter, M. N., & Rivest, R. L. (2021). Going from bad to worse: from Internet voting to blockchain voting. *Journal of Cybersecurity,* 1-15.

Patel, H., Poddar, A., Sekhar, A., & R., S. (2021). Hierarchical Blockchain based E-Voting System. *Grenze International Journal of Engineering & Technology*, 47-56.

Pawlak, M., Guziur, J., & Poniszewska-Marańda, A. (2018). Voting Process with Blockchain Technology: Auditable Blockchain Voting System. *LNDECT*.

Radhika, S., Yuvarani, M., Mathiyalagan, C., & Elakkiya, M. (2021). An Enhanced Security Mechanism For Voting System To Prevent Election Data Tampering Using Iot And Blockchain. *Turkish Online Journal of Qualitative Inquiry*, 3199-3204.

Saldamli, G., Upadhyay, C., Jadhav, D., Shrishrimal, R., & Patil, B. (2022). Improved gossip protocol for blockchain applications. *Cluster Computing*, 1915-1926. doi:10.1007/s10586-021-03504-z

Sayeed, S., Marco-Gisbert, H., & Caira, T. (2020). Smart Contract: Attacks and Protections. *IEEE Access*, 24416-24427. doi:10.1109/ACCESS.2020.2970495

Taş, R., & Tanrıöver, Ö. Ö. (2020). A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting. *Symmetry*.

Thompson, C. (2008, January 6). *Can You Count on Voting Machines?* Retrieved from The New York Times: https://www.nytimes.com/2008/01/06/magazine/06Vote-t.html

U, J., Aziz MJ, A., Z, S., & HA, H. (2022). A Systematic Literature Review and Meta-Analysis on Scalable Blockchain-Based Electronic Voting Systems. *Sensors*.

Wahab, Y. M., Ghazi, A., Al-Dawoodi, A., Alisawi, M., Abdullah, S. S., Hammood, L., & Nawaf, A. Y. (2022). A Framework for Blockchain Based E-Voting System for Iraq. *International Journal of Interactive Mobile Technologies*, 210-222.

Yadav, R. K., Mishra, S. K., Yadav, A. N., & Kamat, S. K. (2021). Blockvote - Blockchain based E-Voting System. *Grenze International Journal of Engineering & Technology*, 562-569.

Yang, X., Yi, X., Nepal, S., Kelarev, A., & Han, F. (2020). Blockchain voting: Publicly verifiable online voting protocol without trusted tallying authorities. *Future Generation Computer Systems*, 859-874.

Zeng, G., He, M., Yiu, S. M., & Huang, Z. (2023). Self–Tallying Electronic Voting Based on Blockchain. *ITNOW*.

Zhang, S., Wang, L., & Xiong, H. (2020). Chaintegrity: blockchain-enabled large-scale e-voting system with robustness and universal verifiability. *International Journal of Information Security*, 323-341.

Zhou, Y., Liu, Y., Jiang, C., & Wang, S. (2020). An improved FOO voting scheme using blockchain. *International Journal of Information Security*, 303-310.