International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 5, October 2021, pp. 4325~4335
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i5.pp4325-4335

Blockchain based voting system for Jordan parliament elections

Mohammad Malkawi[1], Muneer Bani Yassein[2], Asmaa Bataineh[3]
[1]Department of Software Enginering, Jordan University of Science and Technology, Jordan
[2,3]Department of Computer Science, Jordan University of Science and Technology, Jordan

Article Info

Article history:

Received Sep 11, 2020
Revised Apr 1, 2021
Accepted Apr 16, 2021

Keywords:

Blockchain
E-voting
Insert sorted array algorithm
Private blockchain
SHA256

ABSTRACT

Covid-19 pandemic has stressed more than any-time before the necessity for conducting election processes in an electronic manner, where voters can cast their votes remotely with complete security, privacy, and trust. The different voting schema in different countries makes it very difficult to utilize a one fits all system. This paper presents a blockchain based voting system (BBVS) applied to the Parliamentary elections system in the country of Jordan. The proposed system is a private and centralized blockchain implemented in a simulated environment. The proposed BBVS system implements a hierarchical voting process, where a voter casts votes at two levels, one for a group, and the second for distinct members within the group. This paper provides a novel blockchain based e-Voting system, which proves to be transparent and yet secure. This paper utilizes synthetic voter benchmarks to measure the performance, accuracy and integrity of the election process. This research introduced and implemented new algorithms and methods to maintain acceptable performance both at the time of creating the blockchain(s) for voters and candidates as well as at the time of casting votes by voters.

Corresponding Author:

Muneer Bani Yaseen
Department of Computer Science
Jordan University of Science and Technology, Jordan
Email: masadeh@just.edu.jo

1. INTRODUCTION

The importance and necessity of trustworthy electronic voting system (EVS), were demonstrated during the Corona Covid-19 Pandemic. Due to the Covid-19 Pandemic, elections were postponed in several places in the world, and in other places where elections were held, many people were not able to reach election posts due to imposed curfews or fear of infections. In the United States, fifteen states postponed the primary elections due to Corona Covid-19 Pandemic [1].

Blockchain is new imagines to store data in some way that is different from others, which was introduced by Satoshi Nakamoto in 2008 [2], [3]. A Blockchain is a group of blocks connected together by links. Theses links connect a block to the next one, in a manner to produce a chain of arranged blocks [4], [5]. Each block contributes to the building of the next block; this indicates that the connections between blocks are very strong. Each block has transaction data, nonce, signature of a previous block, and signature of the same block. The signature is unique and corresponds to the data and nonce of the block. The process of generating a signature is done using a hashing algorithm [6]. Each block signature contributes to building the signature of the next block. This method of connection is a key to the strength of blockchain technology.

Blockchain technology has several benefits [7]. First, Blockchain is immutable: the content unchangeable and the alternation is impossible and needs high computational power. Second, Blockchain is decentralized [8]: so that it is a peer-to-peer network without centralization points. The content is available for all users in the network. Third, the Blockchain is distributed over many computers, so that there is no single point of failure. Lastly, Blockchain uses cryptographic hash functions for identification and verification processes such as SHA256, SHA512, SHA3-256 to provide a good level of security and to guarantee uniqueness [2], [4], [9]-[11].

Bitcoin is the first application built using blockchain technology [3], [12]-[14]. Another application, which is considered a strong candidate for using blockchain technology, is electronic voting system. Building EVS as a centralized system (Ex: using client server paradigm) can suffer from different weak points such as reliability issues and other problems related to the centralized system like server hacking, denial of service attacks (DoS), and others [3], [7], [14]-[18]. A key factor in a centralized system is that there is only one organization, which has control over all the systems, so the possibility to manipulate the database or system cannot be ignored [9], [14]. If it occurs, different problems can arise in the system such as security and integrity problems. Using blockchain technology in elections rather than traditional ways in voting systems allows addressing essential problems. In conventional elections, quite few people may lose trust in the elections process [7], [19] because of potential cheating, fraud and other issues such as collision, cost, effort and time. Manual voting process suffers from the possibility of the ballot boxes being stolen, broken, or diverted to other locations while in transport [1]. Conventional elections, suffer three major problems: data manipulation, security issues, and transparency [7], [9]. Nevertheless, new technologies carry their own challenges as well, namely challenges related to security, trust, speed, precision and other important standards, in addition to the degree of how the technology is easy to learn, and easy to use.

Choosing blockchain technology versus traditional EVS comes from our desire to build an electoral system with special features and specifications, in a manner to create a fair elections process without any manipulation, changing of votes, cheating, and depriving many people the right to vote due to registration issues, false ID's and the like. Tampering with elections has frequently led to frustration, riots, protests, and vandalism. To overcome these problems, we propose the use of blockchain based election process, to satisfy both government and citizens in the countries. The proposed system, in this paper, is customized and tested against the Parliament election system practiced in Jordan.

The proposed blockchain based election process is designed in a manner to overcome major challenges and concerns such as the denial of service attacks (DoS) and issues like reliability, security, integrity, transparency, voters' exclusion, and anonymity. Using a blockchain structure will also remove the need for a trusted third party, because the blockchain itself is a reliable third party. Furthermore, blockchain based election process eliminates all conventional costs, difficulties, and overcomes the associated challenges [4], [9], [10], [12].

In this paper, we will build our system based on the private blockchain. In the private blockchain, only the entity that owns the blockchain can grant permissions to individuals to use the blockchain and to cast votes. In this model, the government is the sole owner of the election process and therefore is the only responsible party, allowed to give permissions to their citizens to vote using blockchain technology. The government also is the only entity that can add voters to the blockchain. In the private blockchain, because of the limited number of nodes, block construction, and the process of changing the nonce until an eligible signature (mining process), is attained is more cost effective than a public blockchain [20]. On the other hand, the private blockchain, due to the limited computation power of the central owner, will have to compromise on the complexity of the signature.

– Litreture review

Many empires and countries have adopted different versions of election processes throughout the history of humankind. The advancement of computer technology and the internet has opened the door wide open for introducing different forms of EVS [16], [17]. And some of them implemented using a blockchain [20].

David Shaum introduced the first EVS [12], [13], [21] in the early eighties [22]. Different models of EVS were applied in different countries. Estonia, for example, was the first country in the world that applied full EVS in its national elections [12]-[14], [23]. Switzerland later applied an EVS in the statewide elections [24]. In 2011, Norway applied an EVS provided by Scytel vendor in the country's council election [25]. Also in 2015, Scytel introduced a different design for the EVS for New South Wales to use it in its state election [26]. India replaced its paper-based election with an EVS and deployed electronic voting machines instead of paper ballots in 1982 [27]. One example of EVS that failed and was canceled because of flows in their design is a Washington pilot EVS in 2010; it suffered from security issues [28]. In 2009, Austria used an EVS in the student's union elections and the implementation was not successful because of issues related to security [29].

Since the blockchain technology was introduced in 2008, it had been adopted in several applications and in different fields such as Bitcoin [2], [3], [13], [14], [30]. Over time, blockchain was considered a useful technology to implement the EVS. Many researchers proposed e-voting systems based on blockchain technology [2], [4], [14], [31]-[37]. However, the blockchain structure needs to be modified and adjusted to suit different types of election systems.

Different researches proposed and used different technologies to implement EVS such as client/server architecture [16], [17], and blockchain technology [2], [4], [31], [32], [35]-[37]. However, some voting systems failed because they suffered flows in their basic design. The flows in Washington [29] and Astaria [20] EVS led to security violations. EVS suystems differ in several aspects such as data storage, user identification; know your customer (KYC) techniques including users's credentials, biometrics and others. Systems, which use a centralized server and central database to store the elections data [15], suffer single point of failure issues. Some systems use username and password [19] as user credentials for login to the system, which is a traditional way and less, secured compared with blockchain technology [16], [17], [25]. The model presented in this study utilizes private blockchain technology, which is less costly compared with a public blockchain. Private Blockchain saves on the costs resulting from the mining process incurred by publick blockchains. We adapt a singly linked list structure to achieve the requirements of private blockchain technology, such as anonymity, accuracy, transparency, integrity, and others. In our system, the basic unit of a data storage is the blocks, and the data is stored in secured chained blocks. The blocks are immutable and impossibly changed. Those who have key permissions can only access data. The login process in our system uses a signature, which is a secure method of accessing blockchain data. The proposed system is decentralized and distributed, which avoids the problems of single point to failure. Also, we use SHA256 cryptographic hash function for identification and verification processes to guarantee the uniqueness of signatures, and the duplications of records.

The architecture of bbvs for parliament elections; Figure 1 shows the architecture of our proposed system. We consider Karak province in Jordan country as an example to explain our system architecture. Karak province has one electoral district. So, each district in the province has two blockchains: the first one for voters and the second one for candidates.

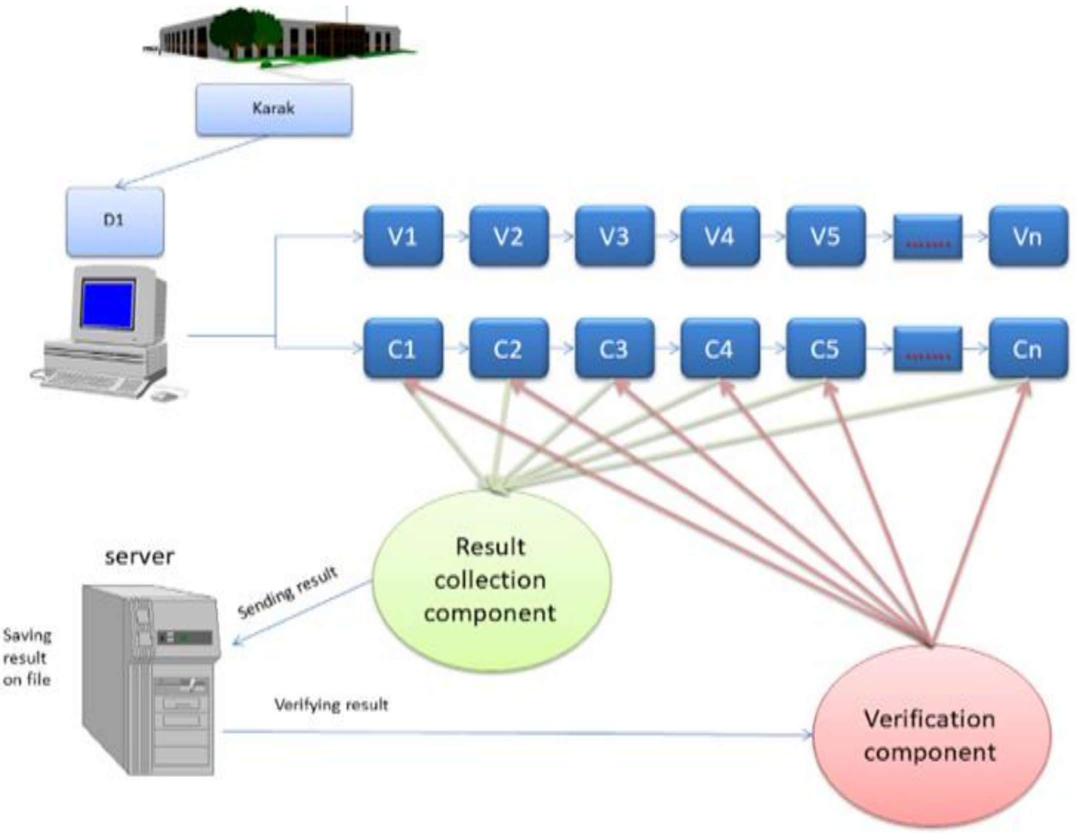

Figure 1. The architecture of BBVS for Jordan parliament

As we mentioned previously, each district has two blockchains: one for the voters and another for the candidates. Figure 2 shows the structure of voter blockchain. Each voter, e.g., voter #6 (V6) is uniquely identified by a hashed code, while its record contains the signature of previous record, the voter's data, a link to the next record, and a random code "Nonce".

To add any voter to the voter's blockchain, he must have a unique and appropriate signature. To generate a signature, we need data, nonce, and a previous block signature. The structure of the candidate blockchain similar to the voter blockchain. It approximately has the same structure with some different attributes. And it works similarly to the voter blockchain. To add any candidate to the candidate's blockchain, he must have a unique and appropriate signature. Also, a condition that must be considered to add the candidate to the candidate's blockchain is that any candidate must be a voter and must be recorded in the voter's blockchain before adding to the candidate blockchain. The next section will contain the Research method. Section 3 will describe the results. Conclusions will be given in section 4.

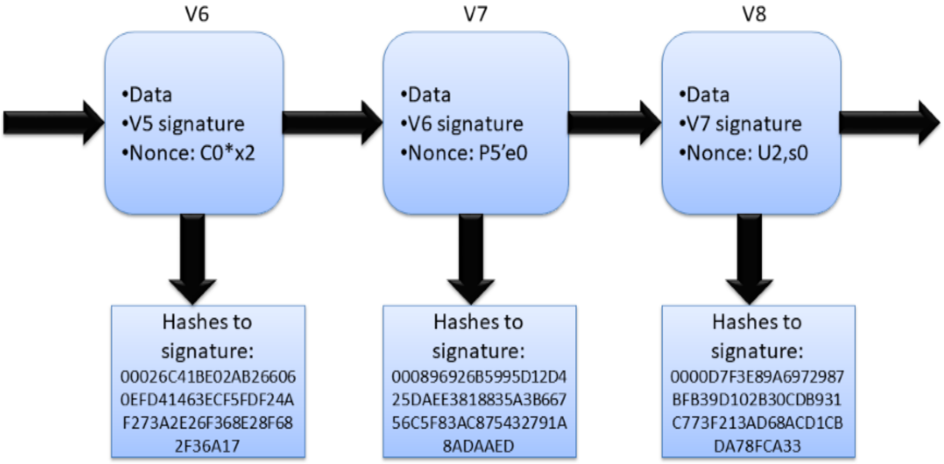
V6
•Data
•V5 signature
•Nonce: C0*x2
V7
•Data
•V6 signature
•Nonce: P5'e0
V8
•Data
•V7 signature
•Nonce: U2,s0
Hashes to signature: 00026C41BE02AB26606 0EFD41463ECF5FDF24A F273A2E26F368E28F68 2F36A17
Hashes to signature: 000896926B5995D12D4 25DAEE3818835A3B667 56C5F83AC875432791A 8ADAAED
Hashes to signature: 0000D7F3E89A6972987 BFB39D102B30CDB931 C773F213AD68ACD1CB DA78FCA33

Figure 2. Voter blockchain structure

2. RESEARCH METHOD

We implement our system using JAVA programming language [36]. We use the eclipse tool [37] and JDK to implement our code. Our electronic blockchain based voting system (BBVS) follows the following steps to achieve a voting process:

2.1. Adding voters

We have voter's blockchain to add all citizens in the country who have the right to vote. So, all privileged voters are added as blocks in the voter blockchain. Each voter has a block to store the required data and his signature. The voter block includes the name of the voter, the social security number (SSN), birth date, voting district, a signature of the same block, and signature of the previous block. Each voter has a unique signature. In our system, the signature is generated using SHA256-bit (cryptographic hash functions for identification and verification processes) from both data and nonce. When a voter is added to the blockchain, we store the SSN in an array data structure (array of voters SSN's) for security purposes. This is necessary to check if the voter already existed or not. A double check step is used to meet this condition by checking whether the signature is stored on the blockchain or in the array of voters SSN's.

2.2. User identification and verification

Any voter should have a unique signature to vote. The login process depends on the signatures as identification of the voter not on the user name and password. Using a signature in the login process gives more security to the voting process. Each voter has a signature according to his/her voting district. The signature will be stored in the voter's smart phone, and can only be accessed through biometrics such as the voter's photo or fingerprints. It is impossible to detect this signature by others. It requires a lot of time and computational power. After the user identifies his credentials, verification of his signature is done to make sure if the user is a valid voter or not. The verification process will be through checking if the voter signature had been stored in the voter's blockchain or not. If the signature exists in any block in the blockchain, so the user is a valid voter and meets all requirements to be a valid voter in the country. If the signature does not exist in any block in the blockchain, so the user is an invalid voter. If the verification process is successful, then the voting process will start. Note that when a voter is created on the blockchain, he/she will be provided with a number of tokens based on the election rules. For example, in Jordan parliamentary elections, the voter will be provided a token for a menu vote and one token for each candidate on the menu.

2.3. Voting

After voter logs in to the electronic voting system, the voting step starts. The voting depends on several parliamentary elections menus. As the voter enters the system, the system will show all parliamentary elections menus of the voting district to which the voter belongs. The voter should select one menu to vote for it. Then the system will show all candidates in this menu on the screen. The voter can vote for one or more candidates on the same menu. The voter can vote to the same candidate only once.

2.4. Adding votes to the candidates blockchain

After the voter selects his candidate's, the system should store these votes in the blockchain. Each vote will go to the corresponding candidate block in the candidate's blockchain and increase the count of his votes. In the meantime, the voter tokens will be decreased one for each selected candidate.

2.5. Checking the accuracy

After casting all votes to the candidate blockchain, the voting process is completed. In our model, we will check the accuracy of the voting process by tracking the number of tokens. The voting process is accurate when the number of tokens granted by a government equals the number of tokens consumed by the voters during the voting process plus the number of tokens that remain unused after the voting process. Other than that, the voting process will not be accurate.

2.6. Collecting the results

In this model, we apply all experiments on three computer machines two of them for the districts elections and the last one for the result collections. So after each machine completes their work, it will send the elections result to the centralized machine by using a client/server model. The centralized computer machine will store the result on the election results file.

2.7. Verifying the results

After the centralized machine collects the results from other machines, it will verify them. The verification process is done by making sure that each candidate owns the votes that are recorded on the election results file. A centralized computer will send each record to the specified machine to make sure that a number of votes which are recorded in the file match the number of votes for the candidate in his block.

2.8. Detrmining the winners

In this step, the proposed system determines the winners according to the open proportional list method.

2.9. Publishing the results

The last step is publishing the results. Each candidate can know his number of votes by his account through his signature. The candidate should log in by using the signature which presents him as a candidate, not as a voter. And after the verification process of all candidates, the government will determine the winner candidates according to the open proportional list method. And it will publish the result of the voting process for the public in the country.

To add a candidate to the candidate's blockchain, he should have a record upon the voters. A candidate has two signatures: The first signature for voting and the candidate has the right to vote, and the second signature for the candidate to know what his status in the election is and to know the election results. According to the parliamentary elections of the Jordan country, once the voter selects any menu a vote will be added to that menu directly. In our electronic voting system, once the system reads the menu number, the system will add a vote to that menu.

We have two major datasets, the first one for voters and the second one for candidates. Each dataset has a different number of records according to each experiment. In the four districts experiments; the voter's dataset has 100, 1000, 5000, and 10000 voters. In the eight districts experiments; we have two voters' datasets: the first one for the first four districts on the first machine with 10000 voters, the second one for the last four districts on the second machine with 4000 voters. The candidate dataset for both districts has 60 candidates. We build our dataset with virtual information that exactly mimics the real information.

3. RESULTS AND DISCUSSION

This paper has presented an adaptation approach of the Blockchain technology to a particular voting system, the Jordan Parliamentary election system. A private blockchain is implemented in a simulated environment, using the JAVA development platform. We have solved the issue of a multi-level voting per a single cast. A voter can cast a vote by generating a voting token to a particular group of candidates. Then, the system generates sub-tokens, and enables the user to distribute the tokens to candidates within the group. This process is not readily available in publicly available blockchain systems such as the Ethereum. We have introduced new algorithms for performance improvement when adding a voter to the blockchain, and for verifying the authenticity of the voter and the accuracy of the results. The proposed system allows for unlimited parallel voting, with a complete protection of critical sections. Our mutual exclusion algorithms are applied at the voting district level. However, we are exploring further parallelism by applying mutual exclusion techniques at the level of the candidates' records. Next, we present more detailed research results of the BBVS system.

3.1. Four districts experiments

We develop multiple versions of our system to determine the behavior of the system in each version, so use a best one. We develop the following versions:

a. Sequential voting

No more than one voter can vote at the same time, each voter must wait all preceding voters.

b. Multi-threading voting with synchronization over all voting process

Synchronization is on the voting process level. All voting process is the critical section, so no two cooperating processes can be in a critical section at the same time. We note that both sequential voting and Multi-threading voting with synchronization over all voting process have the same flow of voting. But they differ in terms of performance. Sequential voting executes a voter code as a single process for each voter. But the multi-threading voting executes each voter as a thread. A thread is a lightweight process and requires less time to execute compared with a process. Also, the sequential voting prevents voters from entering the voting process together by design and forcing the voters for waiting outside the voting structure. Using multi-threading with synchronization a blocking mechanism prevents voters from entering the critical section together. Also, it should be noted that the sequential voting system does not guarantee mutual exclusion.

c. Multi-threading voting with synchronization maintained over each district

Synchronization is maintained at the electoral constituency level, so voters with different electoral constituencies will be able to vote concurrently without any needs to wait for voters in other districts. Only the voters of the same electoral constituency are not allowed to enter a critical section together to guarantee mutual exclusion and avoid a problem of incorrect and inconsistent values of the shared variables. In the subsequent experiments, we only use multi-threading parallel voting with synchronization at the district level blockchain.

The following experiments use multi-threading with synchronization over a district. We also ran the experiments with and without Quicksort algorithm. As shown in Figure 3 the voting process time for 100, 1000, 5000, and 10000 voters is better when using Quicksort algorithm than without using it. The voting process time depends on both locating the voter block in the blockchain, using the tokens of the voters, locating the candidate block, and finally adding the tokens to the candidate's number of votes.

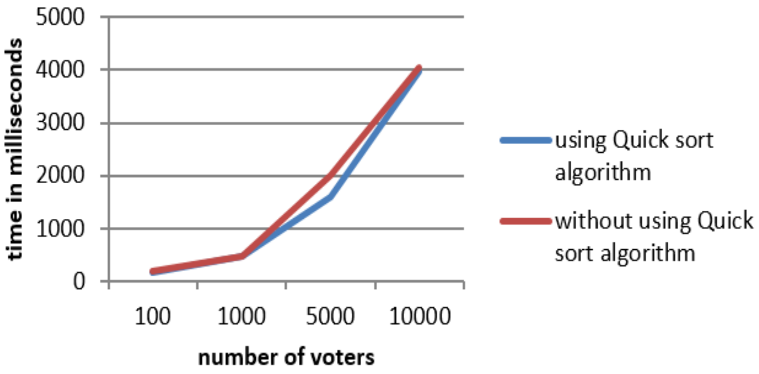

Figure 3. Multi-threading with synchronization over district results using Quick sort algorithm and without

Figure 4 shows that there is a big enhancement on the voting time when we use a multi-threading than sequential voting. Sequential voting is not efficient for building voting systems because the voting process takes a relatively large time and which implies a large waiting time. We implemented multi-threading to shorten the voting time. We also implemented synchronization at the level of the district to protect the critical section and guarantee mutual exclusion when adding votes to the candidates. Although the synchronization adds waiting time, it remains much more efficient than sequential voting and more secure than multi-threading without synchronization. The overall voting time is expected to be reduced. Also, there is the good enhancement when we use synchronization maintained for the district than synchronization over all voting process. So, based on the comparisons between all solutions, we exclude sequential voting and multi-threading with synchronization over all voting process from the experiments of a large number of voters. We conclude that multi-threading with synchronization over the district is the best one for the voting system. Figure 5 shows a comparison of average voting process time between using the Insert sorted array algorithm, Quicksort algorithm, and compared both with without using any sorting algorithm for only 10000 voters. Insert sorted array algorithm is the best one and it has less voting time compared with both using Quicksort algorithm and without using sorting algorithms.

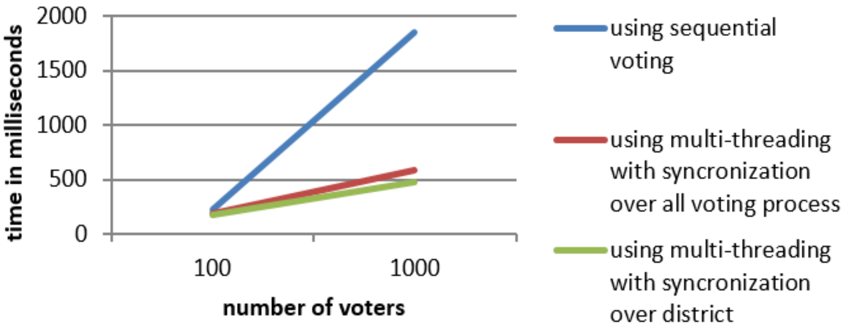

Figure 4. Comparison between sequential voting, Multi-threading with synchronization over all voting process, and multi-threading with synchronization over district voting results

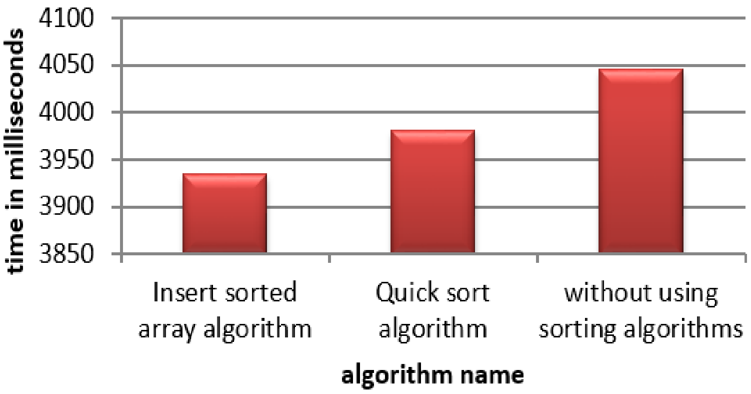

Figure 5. Comparison between the time required to vote using insert sorted array, quick sort algorithms, and without using them for 10000 voters

3.1.1. Adding voter performance

Adding voters can be done only one time. And the generated signatures can be used for one parliamentary election or more. In our system, we tested the possibility of using Quicksort algorithm and Insert sorted array algorithm to store generated signatures in sorted arrays to reduce the time it takes to add a signature to the blockchain. When a new signature is generated, it is checked against previously generated signatures to guarantee uniqueness. We also tested the process of adding voters without using the Quicksort algorithm and insert sorted array algorithm; instead, we build the array of signatures in a manner to maintain it in sorted order. Thus when a new signature is generated, we only use binary search to find whether the signature is unique or not.

Figure 6 shows the Insert sorted array algorithm required less time to add 10000 voters than both using Quick sort algorithm and without using any sorting. The insert sorted array method time complexity for both best and worst case is O(n). But the time complexity for Quick sort algorithm in the best-case O(nlogn) and in the worst-case O(n^2). So, the insert sorted array improves on the required time to add voters.

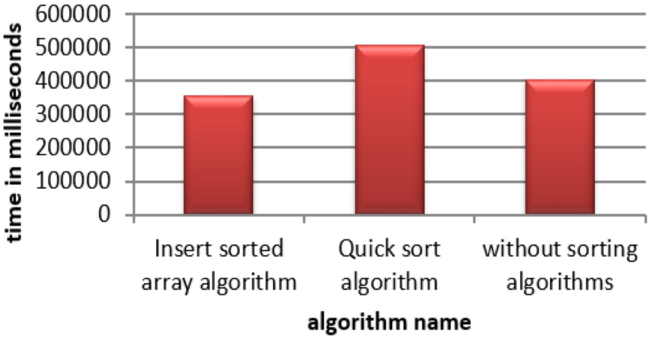

Figure 6. Comparison between the time required to add 10000 voters using insert sorted array, quick sort algorithms, and without using them

3.2. Eight districts experiments

Based on the results obtained from 4 districts experiments, we will use the system model with multi-threading with synchronization maintained for the district and Insert sorted array algorithm. In these experiments, we use three computer machines: the first and second computer machines for districts elections and the third computer machine for the result collections. After each machine completes the voting process, it will send the elections result to the centralized machine by using a client/server model. The centralized computer machine will store the result on the election results file. Verification of the results with blockchain data is carried out after receiving all electoral votes.

Figure 7 shows a voting process time for 14000 voters. The first time, we execute these experiments on two computer machines pc1 and pc2 then we use Pc3 to all 14000 voters. Pc1 has 4 electoral districts and 10000 voters. Pc2 has 4 electoral districts and 4000 voters. The experiments are repeated N times and the results are average over N (We repeated the experiments 10 times). Note that both PC1 and PC2 have the same hardware and software specifications. PC1 processes 2/3 of the voters within 67% of the total time and PC2 processes 1/3 of the voters and spends 33% of the time. Both voting sets of voters include signatures with specific three leading digits. If we run all voters on one machine, instead of 2 (call this PC3) as shown in Figure 8, we get consistent results for the voting times.

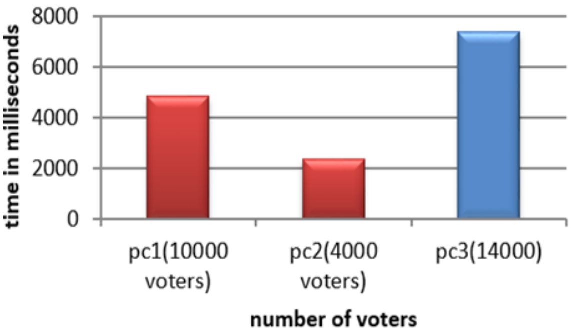

Figure 7. Voting times for the three machines with 10000, 4000, and 14000 voter

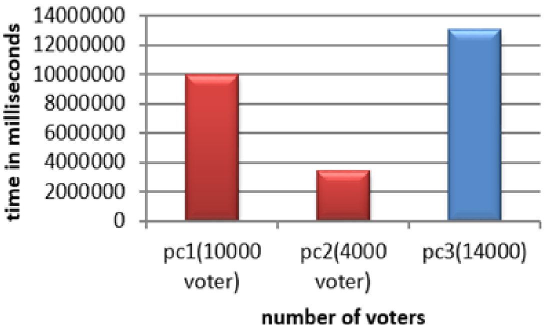

Figure 8. Comparison between the times required for adding 14000 voters using two computer machines and one computer machine

3.2.1. Adding voter's time

Figure 8 shows the average time for adding 10000 voters distributed among the first 4 districts on Pc1 and 4000 voters distributed on the last 4 districts on Pc2, in addition, to add all 10000 voters using Pc3. Pc1 takes almost twice (2.8) as much time to add voters to the blockchains as for Pc2, with the number of voters on Pc1 is almost twice (2.5) as much as for Pc2. Using two machines in the system (for a total of 14000 voters and 8 districts) saves 3% of the total time ((13490600-13042954)/ 13490600) = 3%. For a very large voting population and a large number of districts, the time saving using multiple distributed machines is expected to be significant.

4. CONCLUSION

In this paper, we presented an EVS built over a blockchain technology (BBVS) to benefit from blockchain features. We have adapted the Blockchain technology to a particular voting system, namely the Jordan Parliamentary election system. A private blockchain has been implemented in a simulated environment, using the JAVA development platform. We have solved the issue of a multi-level voting per a single cast. A voter can cast a vote to a group of voters and at the same time selects one or more mbers of the group to receive their own tokens granted by the voter. This process is not readily available in publicly available blockchain systems such as the Ethereum. We have introduced new algorithms for performance improvement when adding a voter to the blockchain, and for verifying the authenticity of the voter and the accuracy of the results. The proposed system allows for unlimited parallel voting, with a complete protection of critical sections. Our mutual exclusion algorithms are applied at the voting district level. However, we are exploring further parallelism by applying mutual exclusion techniques at the level of the candidates' records. Our BBVS is designed to satisfy the requirements of the Jordan parliament elections. Using BBVS, we can overcome the previously mentioned problems of using conventional voting and EVS. BBVS provides complete security, privacy, and transparency. We validate our BBVS using a private simulated blockchain system. We tested several implementations including sequential voting, multi-threading with synchronization over all voting process for all districts and multi-threading with synchronization over each district independently. We also tested several methods for optimizing the processes of creating signatures, adding a signature to the blockchain, and casting the vote. In particular, we ran our experiments with Quicksort algorithm, insert into a sorted array algorithm, and without any sorting algorithm.

The performance evaluation results show that the best results are achieved when multithreading is used to allow voters to vote instantly, anytime, and anywhere. Enforcing synchronization at the district level provides better waiting time performance than synchronization at the level of the entire system. For fast access to the blockchain system, we found that maintaining a sorted array of signatures as they get created provides better performance for adding a voter to the blockchain as well as for casting the votes when compared to either an unsorted array, or an array sorted with quick sort algorithm.

In the wake of Corona Coveid-19 pandemic, Blockchain Based Voting System has become more than just a valid secure way for organizing elections; rather it has become an urgently needed system to enable voting remotely, online, and with zero crowd issues. Besides all the known benefits of blockchain technology, it has added one more feature, which is a crowd management enabler. Many places in the world today are considering either postponing elections or using the regular mail system to conduct elections as is the case in the USA. Our proposed BBVS is by and large an efficient, cost effective, secure, and transparent voting system.

This paper has demonstrated that the blockchain based election process overcomes major challenges and concerns such as the denial of service attacks (DoS) and issues like reliability, security, integrity, transparency, voters' exclusion, and anonymity. Using a blockchain structure is shown to remove the need for a trusted third party, because the blockchain itself is a reliable third party. Furthermore, blockchain based election process eliminates all conventional costs, difficulties, and overcomes the associated challenges.

REFERENCES

[1] C. Kate Sullivan, "Here are the states that postponed their primaries due to coronavirus," CNN, 2020. [Online]. Available: https://edition.cnn.com/2020/03/16/politics/state-primaries-postponed-coronavirus/index.html.

[2] F. Hjálmarsson and G. Hreiðarsson, "Blockchain-based e-voting system," *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, 2018, pp. 983-986, d oi: 10.1109/CLOUD.2018.00151.

[3] M. Rezvani and H. Khani, "e-Voting over Blockchain Platforms: A Survey," *Journal of Network Security and Data Mining*, vol. 2, no. 3, pp. 1-14, 2019, doi: 10.5281/zenodo.3463190

[4] Y. Liu and Q. Wang, "An E-voting Protocol Based on Blockchain," *IACR Cryptology ePrint Archive*, pp. 1043-1045, 2017.

[5] C. Roh and I. Lee, "A Study on Electronic Voting System Using Private Blockchain," *Journal of Information Processing Systems*, vol. 16, no. 2, pp. 421-434, 2020, doi: 10.3745/JIPS.03.0135.

[6] K. Khan, J. Arshad and M. Khan, "Secure Digital Voting System Based on Blockchain Technology," *International Journal of Electronic Government Research*, vol. 14, no. 1, pp. 53-62, 2018, doi: 10.4018/ijegr.2018010103.

[7] R. Taş and Ö. Tanrıöver, "A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting," *Symmetry*, vol. 12, no. 8, 2020, Art. no. 1328, doi: 10.3390/sym12081328.

[8] G. Sun, M. Dai, J. Sun and H. Yu, "Voting-based Decentralized Consensus Design for Improving the Efficiency and Security of Consortium Blockchain," *IEEE Internet of Things Journal*, pp. 1-1, 2020, doi: 10.1109/JIOT.2020.3029781.

[9] R. Hanifatunnisa and B. Rahardjo, "Blockchain based e-voting recording system design," *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, Lombok, Indonesia, 2017, pp. 1-6, doi: 10.1109/TSSA.2017.8272896.

[10] S. Hardwick, F. Gioulis, A. Akram and R. Markantonakis, "E-Voting with Blockchain: An E-Voting Protocol with Decentralisation and Voter Privacy," *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData),* Halifax, NS, Canada, 2018, pp. 1561-1567, doi: 10.1109/Cybermatics_2018.2018.00262.

[11] J. James, R. Karthika and R. Nandakumar, "Design and Characterization of SHA 3- 256 Bit IP Core," *Procedia Technology*, vol. 24, pp. 918-924, 2016, doi: 10.1016/j.protcy.2016.05.184.

[12] A. Ben Ayed, "A Conceptual Secure Blockchain Based Electronic Voting System," *International Journal of Network Security & Its Applications*, vol. 9, no. 3, pp. 01-09, 2017, doi: 10.5121/ijnsa.2017.9301.

[13] P. Shejwal, A. Gaikwad, M. Jadhav, N. Nanaware and N. Shikalgar, "E-voting using block chain Technology," *International Journal of Scientific Development and Research (IJSDR),* vol. 4, no. 5, pp. 583-588, 2020.

[14] D. Shanthi, R. Suvitha and D. Suganthe, "blockchain based e-voting approach in P2P Network," *Journal of critical reviews*, vol. 7, no. 09, pp. 337-342, 2020, doi: 10.31838/jcr.07.09.72.

[15] R. Almeida and L. Camarinha-Matos, "voteChain: Community Based Scalable Internet Voting Framework," *Doctoral Conference on Computing, Electrical and Industrial Systems*, pp. 70-80, 2019, doi: 10.1007/978-3-030-17771-3_6

[16] L. Vo-Cao-Thuy, K. Cao-Minh, C. Dang-Le-Bao and T. Nguyen, "Votereum: An Ethereum-Based E-Voting System," *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, Danang, Vietnam, 2019, pp. 1-6, doi: 10.1109/RIVF.2019.8713661.

[17] S. Ibrahim, M. Kamat, M. Salleh and S. Aziz, "Secure E-voting with blind signature," *4th National Conference of Telecommunication Technology, 2003. NCTT 2003 Proceedings.*, Shah Alam, Malaysia, 2003, pp. 193-197, doi: 10.1109/NCTT.2003.1188334.

[18] S. Srinivas, B. Kumar and R. Srishylam, "Blockchain-based E-Voting System using Proof of Voting (PoV) Consensus Algorithm," *CVR Journal of Science and Technology*, vol. 18, no. 1, pp. 110-114, 2020, doi: 10.32377/cvrjst1819.

[19] S. Shah, Q. Kanchwala, H. Mi, "Block Chain Voting System," Northeastern University, 2016

[20] K. Curran, "E-Voting on the Blockchain," *The Journal of the British Blockchain Association*, vol. 1, no. 2, pp. 1-6, 2018, doi: 10.31585/jbba-1-2-(3)2018 .

[21] R. Shrestha, R. Sah, S. Shrestha, S. Sarawagi and N. Adhikari, "Blockchain Interfaced Sacure E-Voting System," *Journal of the Institute of Engineering*, vol. 15, no. 1, pp. 195-199, 2020, doi: 10.3126/jie.v15i1.27730.

[22] J. Lopes, J. Pereira and J. Varajão, "Blockchain Based E-voting System: A Proposal," *AMCIS 2019 Proceedings*, 2019.

[23] Ü. Madise and T. Martens, "The first practice of country-wide binding Internet voting in the world," *Electronic Voting 2006-2nd International Workshop, Co-organized by Council of Europe, ESF TED, IFIP WG 8.6 and E-Voting*. CC, 2006.

[24] J. Gerlach and U. Gasser, "Three case studies from Switzerland: E-voting," *Berkman Center Research Publication*, vol. 3, pp. 1-17, 2020.

[25] I. Stenerud and C. Bull, "When reality comes knocking Norwegian experiences with verifiable electronic voting," *Electronic Voting*, vol. P-205, pp. 21-33, 2012.

[26] J. Halderman and V. Teague, "The New South Wales ivote system: Security failures and verification flaws in a live online election," *International conference on e-voting and identity*, vol. 9269, pp. 35-53, 2015, doi: 10.1007/978-3-319-22270-7_3.

[27] H. Patil, P. Ladkat, A. Jituri, R. Desai and D. Shinde, "Blockchain Based E-Voting System," *Proceedings of International Conference on Communication and Information Processing (ICCIP) 2019*, 2019, pp. 1-9.

[28] S. Wolchok, E. Wustrow, D. Isabe and J. Halderman, "Attacking the Washington, DC Internet voting system," *International Conference on Financial Cryptography and Data Security*, vol. 7397, 2012, pp. 114-128, doi: 10.1007/978-3-642-32946-3_10.

[29] B. Shahzad and J. Crowcroft, "Trustworthy Electronic Voting Using Adjusted Blockchain Technology," *IEEE Access*, vol. 7, pp. 24477-24488, 2019, doi: 10.1109/access.2019.2895670.

[30] M. Crosby, P. Pattanayak, S. Verma and V. Kalyanaraman, "Blockchain Technology: Beyond Bitcoin," *Applied Innovation Review*, no. 2, pp. 6-19, 2016.

[31] P. Yellamma, P. Anupama, K. Lakshmibhavani and U. Priya, "Implementation Of E-Voting System Using Block Chain Technology," *Journal of critical reviews*, vol. 7, no. 06, 2020, doi: 10.31838/jcr.07.06.149.

[32] K. Khan, J. Arshad and M. Khan, "Investigating performance constraints for blockchain based secure e-voting system," *Future Generation Computer Systems*, vol. 105, pp. 13-26, 2020, doi: 10.1016/j.future.2019.11.005.

[33] K. Sung, Chae-rin Jeong, Eun-a Cho, Jong-ho Lee, Hee-young Kim, Young-woo Kim *et al.*, "An Intramural Electronic Voting System Based on Blockchain," *Journal of The Korea Institute of Information Security and Cryptology*, vol. 28, no. 4, pp. 779-787, 2020, doi: 10.13089/JKIISC.2018.28.4.779.

[34] Y. Li, W. Susilo, G. Yang, Y. Yu, D. Liu and M. Guizani, "A blockchain-based self-tallying voting scheme in decentralized IoT," *IEEE Transactions on Dependable and Secure Computing*, 2020, doi: 10.1109/TDSC.2020.2979856.

[35] B. Wang, J. Sun, Y. He, D. Pang and N. Lu, "Large-scale Election Based On Blockchain," *Procedia Computer Science*, vol. 129, pp. 234-237, 2018, doi: 10.1016/j.procs.2018.03.063.

[36] "Java Programming Language", Docs.oracle.com, 2020. [Online]. Available: https://docs.oracle.com/javase/8/docs/technotes/guides/language/index.html.

[37] "Eclipse IDE for Eclipse Committers | Eclipse Packages," Eclipse.org, 2020. [Online]. Available: https://www.eclipse.org/downloads/packages/release/neon/3/eclipse-ide-eclipse-committers.

BIOGRAPHIES OF AUTHORS

Mohammad Malkawi received his Ph.D degree from the University of Illinois at Urbana-Champaign in computer engineering in 1986. He is currently associate professor at Jordan University of Science and Technology and adjunct professor at Capitol Technology University, USA. Dr. Malkawi has worked at University of Wisconsin Milwaukee, Motorola Inc., ORACLE/SUN Microsystems and Cambium Networks. Dr. Malkawi has published numerous papers in international journals and conferences and has several patents in the area of information and communication technologies. He is the founder of CCT Inc., a startup devoted to fighting forgery and counterfeit in financial, academic, and governmental institutions. Dr. Malkawi has lectured at international conferences in various topics including blockchain technology, cyber security, innovation and entrepreneurship. His interests include blockchain distributed systems, high productivity computing, system modeling and simulation, broadband WiFi, reliable and high availability computing, distributed and parallel algorithms.

Muneer Masadeh Bani Yassein received his PhD degrees in Computer Science from the University of Glasgow, U.K., in 2007, He is currently professor in the Department of Computer science at Jordan University of Science and Technology (JUST), Muneer served as Chairman of the department of Computer science from 2008 to 2010, as Vice Dean of the Faculty of Computer and Information Technology from 2010 to 2012, 2013-2014 and 2018 to present. Muneer is currently conducting research in Mobile Ad hoc Networks, Wireless sensors Networks, Cloud Computing, simulation and modelling, Internet of Things, Bani Yassein has published over 170 technical papers in well reputed international journals and conferences. Professor Bani Yassein is member of IEEE and he is a member of the technical programs of several journals and conferences.

Asmaa Bataineh received her master degree in Computer Science from Jordan University of Science and Technology (JUST) in 2020. Asmaa served as Teacher Assistantant in JUST from 2016 to 2019.