Proceedings of 12th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2022 Month: October Volume: 12

Blockchain based E-Voting System with Zero-Knowledge Proof using Smart Contracts

Juned Alam [a], Shashidhar Ram Joshi [b]

[a, b] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Nepal*
[a] 076msdsa006.juned@pcampus.edu.np, [b] srjoshi@ioe.edu.np

Abstract
This paper proposes an e-voting system's architecture built on top of block chain. The proposed system uses smart contracts to run all the process related with voting. This proposed architecture also utilizes paillier cryptosystem to achieve zero-knowledge proof, so as to encrypt the voters vote and perform tally on the ciphertext of the votes itself using homomorphic encryption algorithm; homomorphic encryption allows us to perform mathematical computations on the ciphertexts without the need to decrypt the ciphertexts. The focus of this paper is the use of blockchain to conduct eVoting, with zero knowledge proof as the security mechanism to provide anonymity to the voter's vote.

Keywords
blockchain, zero-knowledge proof (ZKP), smart contract, homomorphic addition, paillier cryptosystem

1. Introduction

Voting by definition is the process of selection of choices or people by a group of the populace where the election is happening. Elections have been happening since the dawn of human civilization for various purposes like electing government(s), person, or a group of persons representing a political party. The conventional way of voting by the use of papers to cast vote preserves anonymity and provides security at the same time, but cannot guard against tampering with the ballot box itself; even the ballot box can be replaced during the transferring of the ballot box to the counting center. There are some substitutes for this traditional way of casting vote via paper. The substitutes utilize digital technologies for storing and tallying the casted votes. The substitutes are the Electronic Voting Machine (EVM) and electronic voting, often shortly referred to as e-voting. The e-voting is the way to achieve reliable and secure election.

Unlike the preferential kind of voting system like the ranked voting system, where the voter can rank the candidates on the basis of choice, the e-voting system proposed in this paper only supports the rigid kind of voting structure, where a candidate can only vote to only one person, party or choice, per each voting section.

This paper proposes to utilize the blockchain to store the votes of the users during e-voting. Blockchain technology was introduced in 1991 by Stuart Haber and W Scott Tornetta to cryptographically secure digital chain of records in a tamper-resistant way that too without the need of any central governing body. Blockchain is a progressively appending record of digital transactions. Each of those records of digital transactions is referred to as a block. Each of those blocks is linked to previous and upcoming blocks with the help of pointers; the pointers are nothing, but the hash of the block. The block thus linked with one another form a singular chain-like structure devoid of any branches, hence this system of storing data is referred to as the blockchain. The first block in a blockchain does not point to any block and is thus called the starting or the "Genesis" block. Blockchain technology when properly used and designed makes it very very hard to modify any previously recorded digital transactions, so blockchain is also referred to as an unchangeable register.

2. Related Works

Much research has been done on the use of blockchain, to securely and immutably store data in various domains like food supply chain management[1], product authentication[2], health data

management[3], nuclear waste management[4], retail shopping[5], plasma supply chains[6], and so on.

Paper[7], evaluates the technology, framework, cryptographic algorithm, and consensus mechanisms of various blockchain-based decentralized online voting platforms like follow my vote[8], voatz[9], and polys[10].

Paper[11], proposes a simple e-voting system with a simple user interface, the process of which is dictated by the smart contracts and hosted on etherium framework.

Paper[12], proposes a similar e-voting concept like paper[11]. But these two papers do not focus on preserving the anonymity of voters' voting data.

This paper is based on the paper[13], which proposes a sealed-bid auctioning system, where the auctioning amount of the auctioneers is hidden until the completion of the auction and even the winner of the auction is decided without revealing the auctioneer's data with the help of zero-knowledge proof to achieve this level of anonomity.

3. Methodology

3.1 Blockchain

A typical blockchain consists of blocks, which are linked to previous and upcoming blocks with the help of hash pointers as shown in Figure 1.

A typical block in blockchain consists of block-id, data; in the case of this paper, a list of voters who have voted and the list of tallied votes, after applying a homomorphic encryption algorithm on cipher texts of the voting transactions; a hash of the current block and its previous block, timestamp pointing to the exact time of block creation, and a number called the nonce.

Structure of a typical block looks like in Figure 2. The hash of a block in this paper has been generated by passing a block to a SHA-256 hashing algorithm. SHA-256 produces a 64-length of hash, each character of this hash is a hexadecimal character. The total possible hashes that can be produced by SHA-256 algorithm are 16^{64}.

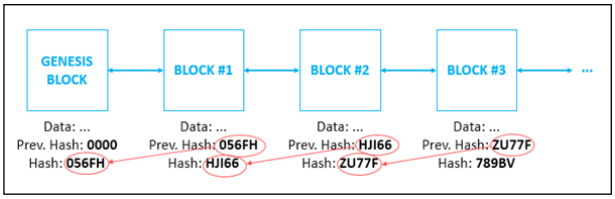

Figure 1: Structure of a blockchain

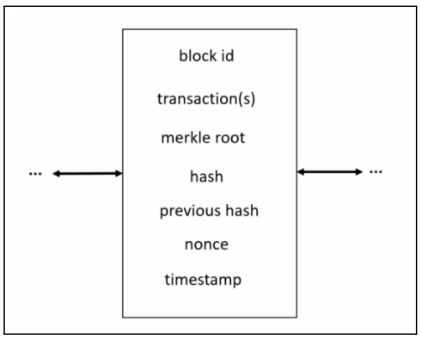

Figure 2: Structure of a block

3.2 Proof of Work

Consensus in a decentralized system can be achieved in various ways. Most of the systems using blockchain, use Proof of Work (PoW) to achieve consensus. PoW is a consensus mechanism, where miners; a group of computers that solve a mathematical task; create a valid block by spending some time and processing power to show that the valid block the miner has proposed has come after some actual spending of computing resources. The mathematical problem previously mentioned is the generation of an SHA-256 hash with a certain amount of leading zeros termed as the difficulty level of mining. With 16^{64}, possible hashes, the probability of finding a hash with 1 leading zero becomes $(16^{63})/(16^{64})$, which is 0.0625. For a difficulty level of x, the probability of finding hash with x leading 0 is given by the equation 1.

$$probability(p) = \frac{16^{64-x}}{16^{64}} \tag{1}$$

Since the parts of a block like block-id, data, previous hash, and timestamp are constant, the thing that can be changed to create a different hash is nonce. Nonce is a variable that is varied; in the case of this paper, I have chosen the datatype of nonce as an unsigned integer in java, whose maximum value is 2^{31}, which is about 4 billion. So, in PoW, that value of nonce is incremented by 1 form 0 to the maximum limit of 2^{31}. Since during the hashing process, even a single bit change results in a vastly different hash value from a previous hash, the miners thus cannot skip even a single nonce value and have to mundanely calculate hash for each possible hash to generate a valid block with appropriate number of leading zeros and thus a

considerable amount of computing and thus time is spent inorder to achieve valid block in PoW.

3.3 Smart-Contract

Smart-contract is a set of rules and automation scripts that set the pace and working of the blockchain application. In this paper's case the smart case governs the difficulty level, manages the various keys and set the time for the different phases of election.

3.4 Memory-Pools

Memory-Pools are caches of temporary transactions which are added stored in in-memory of a machine until the transactions are finally added in a block. Once the transactions are added to the block of a blockchain, the respective transactions are deleted from memory-pools.

3.5 Paillier Cryptosystem Algorithm

Paillier cryptosystem algorithm is a type of ZPF that is probabilistic; the ciphertext produced for the same pair of keys run consecutively in a machine does not match, due to the introduction of a random factor. Paper[14] proposes the pailler cryptosystem algorithm as follows:

3.5.1 Key Generation

- two big prime numbers a and b of equal length should be chosen randomly
- the following computation should be performed:
  $n = a * b$,
  $\lambda = lcm(a-1, b-1)$
- random number r should be chosen such that $r \in Z^*_{n^2}$
- μ should be chosen such that: $\mu = (F(r^\lambda mod n^2)^{-1} mod n$
  where, $F(\mu) = \frac{\mu-1}{n}$
- the public key is: (n, r)
- the private key is: (λ, μ)

3.5.2 Encryption

- for a message m to be encrypted, where $0 <= m <= n$
- a random number x should be chosen such that $0 <= x <= n$
- the cipher text can be obtained as:
  $c = r^m . x^n mod n^2$

3.5.3 Decryption

- for a ciphertext c to be decrypted, where $c \in Z^*_{n^2}$
- message m can be calculated as:
  $m = F(c^\lambda mod n^2) * \mu mod n^2$

3.5.4 Homomorphic encryption property

- the product of two ciphertextes will be decrypted to the sum of their plaintexts as:

$$D(E(m_1, r_1) * E(m_2, r_2) mod n^2) = (m1 + m2) mod n$$

3.6 Proposed System

The architecture of the proposed system is shown in Figure 3.

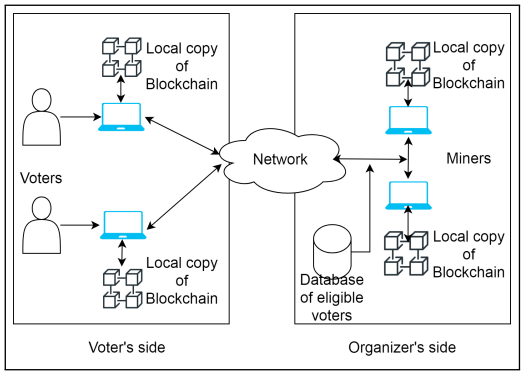

Figure 3: Proposed Architecture

3.7 Stages of election

The system consists of two parts: the voters and the organizers. The organizers of the election would first register the voters, then provide the voters with a unique id and a twelve-character unique string, which the voter will use to log in to the system during the election. The organizers will then initialize a smart contract which is basically a configuration file, where they will set up the election start-time, period of election, difficulty level, and length of the random numbers to be used in paillier cryptosystem. The organizers will then expose port(s) for the voters to connect to during the election. The voters at the time of the election will connect to the organizers end with the credentials provided earlier and a One Time Password (OTP) which would identify a unique voter. They would cast their vote, the encryption of the vote will happen in the system; it would be further explained in the upcoming section; then the voters can either choose to leave the network or be connected

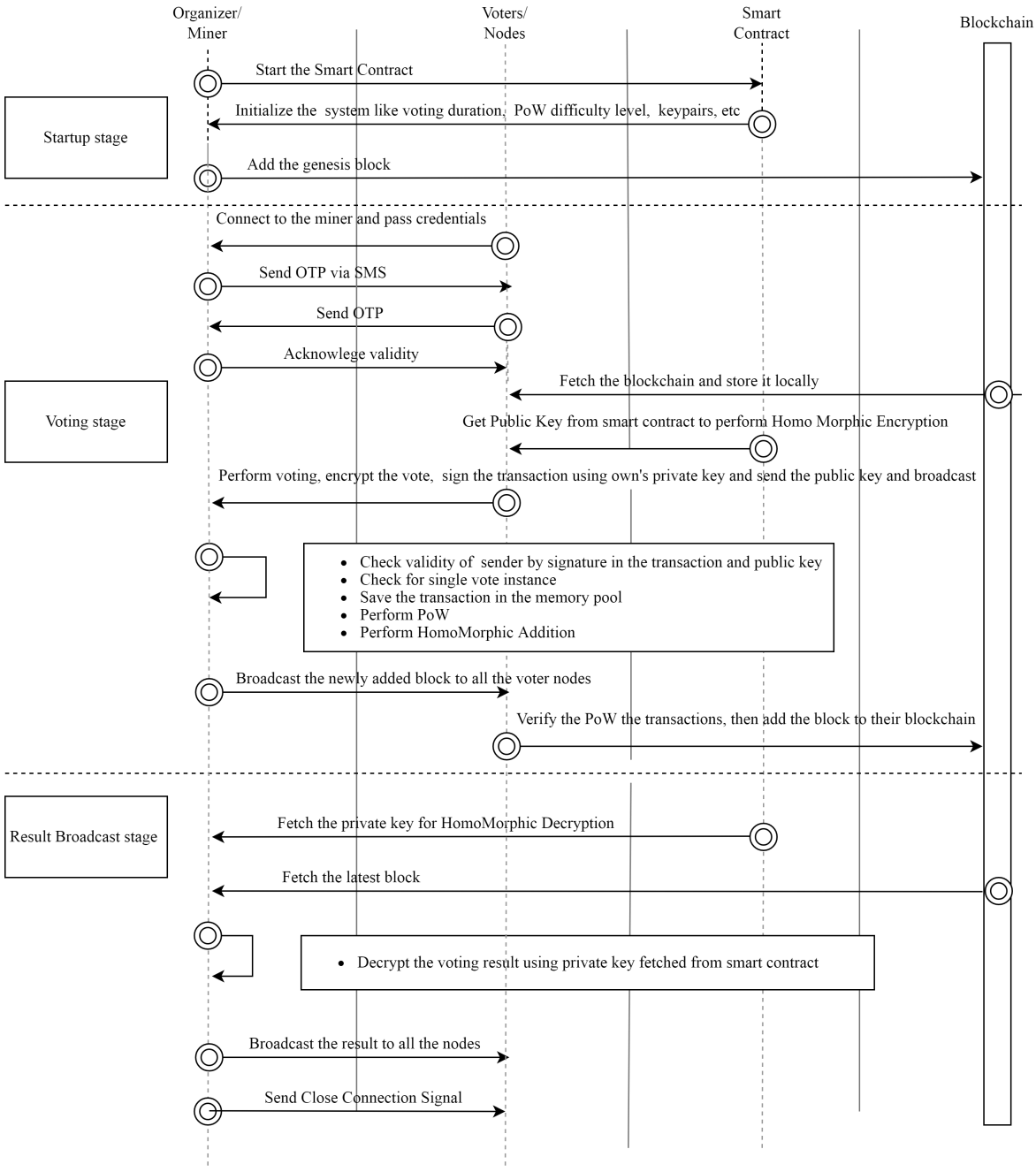

Figure 4: States sequence diagram of different stages of election

until the end of the election process and store the blockchain locally.

The entire election process has been divided into three stages: Startup stage, Voting Stage, and Result Broadcast Stage.

3.7.1 Startup stage

During this stage, the organizers would set up a smart contract and with it, set configurations like the election start-time, period of election, difficulty level, and length of the random numbers to be used in paillier cryptosystem. In turn, the miners would get the configurations and accordingly create a genesis block for the blockchain and expose port(s) for the voter(s) to connect to. This stage is relatively shorter, compared to the Voting Stage.

3.7.2 Voting stage

During this stage, the voters would connect to the port(s) exposed by the organizers, and authenticate themselves against the miners with the help of their id and twelve-characters long string and an OTP sent to

their number. Once authenticated, the voter's end will receive a copy of the smart contract, with the modified election duration, containing the end time of the election, along with the public key "(n, r)" to encrypt the votes using paillier cryptosystem and the current blockchain. The voter will then give their vote and broadcast their votes to all the nodes after signing it with their private key using any digital signature tool; in this paper, Elliptic Curve Digital Signature Algorithm (ECDSA) has been used; to prove that a vote has indeed been cast by the appointed voter. Now, in the miner's end, they will fetch a certain number of voter's votes from their memory-pool, depending upon the configuration. The miners will then perform homomorphic summation upon the votes without actually decrypting their votes as shown in Figure 5. The mining will be performed and once it is done, the tallied votes and the list of voters would then be broadcasted to all the other nodes after being added to the block and the miners would update their local copy of blockchain accordingly. The voters after receiving the block would confirm its validity using the hash of the block and if valid would add the block to their local copy of blockchain.

3.7.3 Result Broadcast stage

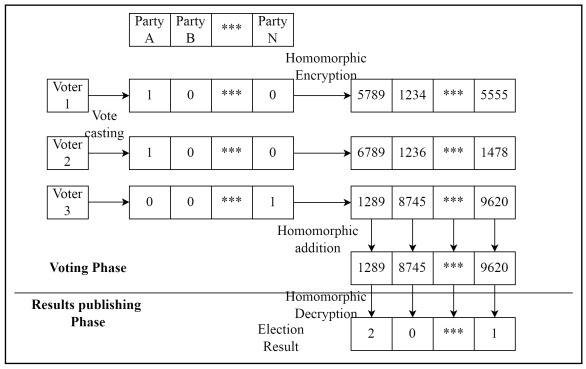

Figure 5: The ballot of voters

During this stage, once the election period has ended, the miners will get the the private key "(λ, μ)".

The miners will fetch the latest block in the blockchain, decrypt the voting result using the private key "(λ, μ)" as shown in Figure 5 and then broadcast the result to all the other nodes.

4. Results

The experimentation for this paper has been performed on a machine with the specifications as shown in Table 1

Table 1: Specification of the machine on which experimentation were done

Parameters	Specification
CPU	2 GHz
Memory	16 GB
Number of Processors	4
Hard disk	1 TB

The execution time of different stages of paillier cryptosystem for different length of prime numbers for the key generation is shown in Table 2

Table 2: Execution time of the paillier cryptosystem (in ms) for different length of prime numbers for key generation

Seed length	encryption	summation summation	decryption
512	3	1	2
1024	20	1	19
1536	55	1	58
2048	131	1	130
2560	255	1	256
3072	388	1	397
3584	619	1	629
4096	901	1	896
4608	3290	1	3239
5120	4394	1	4389

The test including all the parameters is shown in Figure 6, where the experimentation has been done to mine a block of 5 parties with 512 length of prime numbers for paillier cryptosystem key generation. The time taken to mine 50 to 1000 transactions (in ms)

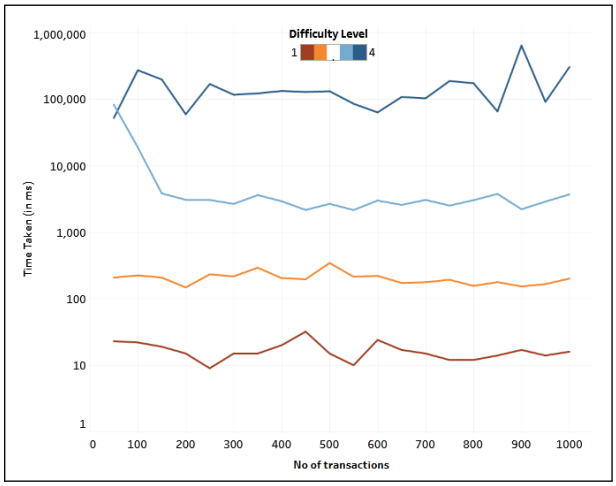

Figure 6: Performance of the system for mining

(average of 20 different transaction was considered for each individual case) for different difficulty levels ranging from 1 to 4, using a single thread was measured. This showed that with each rising difficulty level, irrespective of the number of voting transactions, the time to mine increases exponentially.

Table 3: Time taken (in ms), to complete each stage of the election for different number of nodes

number of voter nodes	initialization phase	voting phase	ending phase
10	1356	121740	31
100	1525	120568	36
1000	1524	120816	6567
10000	1530	120182	13685

Table 3 shows the time taken to complete each stage of the election for different numbers of voter nodes when the difficulty level is set to 1, and the election duration is set to 2 minutes, with the number of transactions set to 50. The startup stage is almost constant for different numbers of nodes, as it does not involve the voter nodes and involves the creation of the genesis block. The voting duration also remains the same since the mining phase and the communication between the nodes gets completed before 2 minutes. The result broadcast stage seems to increase with the increase in the number of nodes; it further validates the blockchain trilemma. two of three benefits at any given time with respect to decentralization, security, and scalability. The blockchain trilemma is the phenomenon of only achieving two out of three benefits at any given time with respect to decentralization, security, and scalability. Here, decentralization and security were opted for, which led to reduced scalability, that is the throughput of the application was reduced, with the increasing number of nodes in the proposed system.

5. Conclusion

This paper proposes a secure system of blockchain-based e-voting system which preserves the anonymity of the voter's vote. The system metrics proposed on this paper can be readjusted based on the voting requirements and the size of the populance and other security requirements. Thus the proposed system in the paper, by preserving the anonymity of the voter's vote can allow the voters to be a part of the blockchain and in turn the entire system, enabling a transparent and secure election in the future.

6. Future Enhancements

The proposed system presents decent results for the e-voting scenario, with various system metrics that can be readjusted for various scenarios, but there is still room for improvement by using efficient homomorphic encryption algorithm and different consensus protocols.

References

[1] Chen Zhang. *The applications of Blockchain in food supply chain management*. PhD thesis, 03 2022.

[2] Md. Rakibul Hassan Robin. *Product Authentication Using Blockchain*. PhD thesis, 07 2021.

[3] Cedric Strub. *CONTRIBUTION OF BLOCKCHAIN TO HEALTH DATA MANAGEMENT*. PhD thesis, 04 2021.

[4] Irene Gelyk. *Applicability of Blockchain for long-term digital preservation of the Canadian nuclear waste management deep geological repository information assets: a literature review.* PhD thesis, 05 2022.

[5] Niru Raj. *How Blockchain transforms the Future of Retail Shopping*. PhD thesis, 10 2020.

[6] Saipavan Vallabhaneni. *Leveraging Blockchain for Plasma Fractionation Supply Chains*. PhD thesis, 04 2020.

[7] Uzma Jafar, Mohd Aziz, and Zarina Shukur. Blockchain for electronic voting system—review and open research challenges. *Sensors*, 21:5874, 08 2021.

[8] Follow my vote. https://followmyvote.com. Accessed: 06 07 2022.

[9] Voatz. https://voatz.com/. Accessed: 06 07 2022.

[10] polys. https://polys.vote/. Accessed: 06 07 2022.

[11] Saad Khan, Aansa Arshad, Gazala Mushtaq, Aqeel Khalique, and Tarek Husein. Implementation of decentralized blockchain e-voting. *EAI Endorsed Transactions on Smart Cities*, 4:164859, 07 2018.

[12] Subashka Ramesh. E-voting based on block chain technology. 02 2022.

[13] Xiaohua Wu, Huan Liu, Fengheng Wu, Fangjian Yu, and Hongji Ling. A low-cost and verifiable sealed bid auction protocol based on smart contracts. pages 1–3, 05 2022.

[14] Paillier cryptosystem. https://www.lamsade.dauphine.fr/~litwin/cours98/Doc-cours-clouds/Paillier%20cryptosystem%20-%20Wikipedia,%20the%20free%20encyclopedia.pdf. Accessed: 06 07 2022.