Blockchain Based Public Cloud Security for E-Voting System on IoT Environment

A Vijaya Kumar¹, Gabbita Viswa Sarvani² and Dama Satya³

¹Assistant Professor, ², ³Student
⁴Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India,

E-mail: vijay.adury@gmail.com

Abstract. Digital Technology is inevitable for better life of the people. Voting by people plays the vital role in democracy as the leaders will be elected by it. IoT helps many edge devices to interact with a Cloud for storage and computation of the Voting results. The major concern for the Voter is whether the vote casted by him is tamperproof or not. Though Cloud offerings are started a decade ago still information security in the cloud is always a concern. This paper focuses on creating a tamperproof framework to ensure the data Integrity. It proposes a novel approach where the user can cast the vote from any of the electronic devices with a unique ID allotted to him, a two-step authentication can be applied like OTP to ensure the security. The votes from these edge devices will be sent to the untrusted third party public cloud for computation, the votes will be saved in the encrypted format. The computation will be done on encrypted data. The Block chain based workflow ensures that the votes which are shared in the encrypted format will not be tampered to ensure trust and Integrity to the proposed model. The model also can be extended to many public shared e-models like E-Auction and healthcare systems.

1. Introduction
Secure E Voting is the Key concern to implement with, E Voting is becoming more popular nowadays specially in the Covid 19 pandemic many democratic countries are looking forward for the elections with E voting as an approach [2]. Many IoT [1] Devices will enable the voters to vote in different modes like EVMs, Mobile, and Web modes. The key concern in this process is the mechanism is secured? Cloud storage and computation are required for these applications. The security in the private or public cloud is always vulnerable [4]. The User votes will be sent to the Center cloud server via IoT Edge Devices [1]. This paper focuses on the data received from the IoT devices to be stored in the encrypted format so that data would be secured. Important attribute in the security aspect is Integrity to ensure that the votes casted by the voters are not tampered. To ensure this model we propose a model in which the block chain approach is applied to the Voting data storage and managing and the result evaluation. This paper gives overview of how block chain works [6] for the related problems where public trust and data integrity is key. The related work happened in this model is discussed and we propose a model which store the data in the encrypted format and block chain models applied on it so that the data is immutable which ensures the security and trust to the end user.
2. Literature Review

The data security in the public and distributed cloud environments is discussed in [6] [7]. There are approaches which dealt with IoT enabled security approaches [8]. The device is so near to the user and relatively close, by the designation it is referred as an edge server [1]. It is a metaphor in which a set of elements providing service need not be addressed individually. Hardware and software managing can be thought as a cloud amorphous. These models how to securely transmit the data from the Fog/Edge IoT devices to the public cloud environment. The Data authorization [9] and security enablement in the distributed environments [10] have their role specific to the cloud environment not to focus on the data mutable nature, though the information stored securely but in the mutable nature. This may lead to tamper the votes posted by the people and election results will be changed with the help of few hackers. This paper discuss the e-voting model enabled with the block chain environment and the parties involved in this process and their roles. It also relates the proposed model with the current models [11] and helps to understand how the model added one more level security to the existing approaches. The approach can also be extended to the areas like healthcare [13] and e auctions [14] models too. Few watermarking models are used in the healthcare models [15], however the data integrity is not addressed in those models. The block chain based voting systems also referred as liquid democracy [16].

2.1 Block Chain Model

The key purpose of using the block chain to provide immutable records to provide data integrity on vulnerable cloud environment. The security provided by the block chain is acceptable across the globe. The distributed nature of block chain makes it best coupled with the untrusted public cloud environment. In the propose application the user votes for the contestant in various modes and the data will be submitted to the untrusted cloud environment through the IoT edge devices. The faster nature of the transactions enable the block chain model to work with many public domains.

As the name implies the previous block information will be stored in the next generated block through which data stealing is difficult as any modification to the data will lead to the invalid consensus to generate the next block of information. The voting process should be secured because it may be attacked by any party to change the poll results. The encrypted format data ensures the security of the model and immutable nature makes the election details cannot be compromised.

![Figure 1. Block chain model](image-url)
Fig 1 helps to understand the basic component working of the block chain model. A person is sending the amount to the second person. The transaction will be represented as a block and the multiple participates in the transaction need to approve the transactions enables the data to be tamper proof. The main business model will be called as smart contract, here its e voting model. Only when consensus generated then only the transaction approves and a new block will be created and added to the existing blocks, any changes to the existing votes will make the system in the instable state, leads to the rejection of the transaction and the block will not be created.

3. Proposed Model

3.1 Block chain-Based E-Voting System in public Cloud

The block chain based electronic voting systems [12] must satisfies the all legal requirements and to obey all the rules that are given by the government. Distributed technology is the present trending technology which is advanced technology in the field. The Block chain technology has infinite number of applications like healthcare [13] and e auctions etc.. This model propose the model with E voting scenario o utilize the advantages of block chain in the implementation of e-voting system.

The above model explains the election roles and the process. First admin will create the E voting model and whenever elections announced he must activate the model as regulated by the Government. The system should progress throughout the voting procedure and will be closed once the election voting timing is completed.

The voter first will be authenticated with the Unique IDs provided by the Government for Identification and authorization purpose

Now root node is the important node where he is going to establish the network without the network the process cannot move forward so his job is to observe the network is in good condition or not. These are the election roles and the process.

Figure 2. Voting process as smart contract

Every business model in the block chain environment will be considered as a small contract between the entities [17].

We propose the election process as the smart contract, first the voting regulatory starts creation of election contract where he want all the details of voting districts or constitutions and the number of candidates present later it has been taken by the ballot contract where they can divide them into different districts and also candidates according to their districts later on all the ballot contract give the details to the connected block chain. Consolidated details from the block chain box it has been separated by the POA network.
In that network there are different foot nodes which are needed to manage the network and the root node start separating all the nodes into different nodes example: node1, node2 and soon.

### 3.2 Securing e-voting based on block chain in peer to peer distributed network

The voter should provide the credentials of his unique ID and may be 2 level authorization [18] have to give all the credentials for the authentication box to know about authenticity after that he have to be retrieved by the voter to find his constitution or district and generate him a voter wallet and then he return to corresponding smart contract these all details are again verified by the API including the OTP generated randomly to authenticate the user.

The Voter then will be navigated to the Election poll details where the vote can be casted for the desired contestant. The confirmation of the Voting completion also will be intimated on the IoT device or through an SMS feature. The process will be implemented with the POA network mentioned below.

The voter has to enter to the smart contract and sign in to his vote and verify his vote with the help of the POA network. Where the POA network consists of the root node and root node must manage all the nodes and verify the condition of the network.

At the last step it transacts its id by the POA network to the voter.

![Figure 3. New block added to the chain for vote casted](image)

The POA network which is arranged by the government which is manages the boot node and the boot node is handles network connection. We are adding a new block to the chain where all the hash values are valid with the help of consensus process which helpful to authenticate the person every node is connected to the previous node and contains a chain and it is connected to a POA network.

### 3.3 Securing e-voting based on block chain in peer to peer distributed network

E-voting model is based on the chain of blocks which is connected with the help of P2P network. In P2P the network is connecting to the every other client or sometimes all the clients are connected to the server which is main.
All the data will be sent to the Centralized Public cloud for the computation. The data alteration in the untrusted party is vital, in this regard ledger based block chain model is part of the proposed model.

The block takes the basic details of voter ID preferably the Unique ID allotted to the Voter, Voter digital signature could be an OTP to confirm the voter’s identity, timestamp of the vote and Hash of the previous block. These details will be verified by the consensus and based on the outcome of it a new block will be generated means the vote will be validated. In this approach every vote contains the previous hash values through which any previous vote will not be tampered.

The officer update the details to time into the vote database and there are some query keys with the voting officer for the public key and start encrypting all the details of the people and if correct he can vote otherwise he is not able to vote. If he is minor he is going to enter the voting block chain if he is allowed he is started voting .if there is no access he have to withdrawal the vote.

The message parsing is going to analyze all the messages and they start doing all the 64 iterations for the every chunk and they gives the hash values.

The voter can vote only when the given credentials and OTP authorization matches with the details of the Government database. The voting office role update the keys by the public keys and verify the votes and query the votes and in the voting block chain. Initially only one parent block will be there and all the votes casted will be chained to that.

When we are adding the first block we are going to check the person is able to vote or not.

And now the voter has both the public and private key with him he can enroll into it and start his process and cast his voting to the particular person.
The voter before voting also can ensure the details provided by the Voting officer matches to the details.

We know electronic voting is a secure process technology. The electronic voting machines must satisfies the all legal requirements and satisfy all the rules that given by the government. Distributed IoT based edge devices helpful in constructing in latest. To utilize all the applications of block chain in the implementation of e-voting system.

3.4 Secure E Voting system with Block chain in the public cloud model

The E Voting process steps:
1. Voter Must be registered as voter at competent authority
2. The Voter would have the Online credentials with Unique ID allotted to him by authority
3. The User would be login through any of the edge devices
4. The user credentials will be verified and OTP will be generated to ensure 2-Step authentication
5. The voter details will be verified and the Candidate details will be displayed to vote.
6. The User vote will be encrypted and stored in the Cloud for security reasons
7. For tamperproof nature the votes will be saved in the form or blocks using block chain technology
8. The user will close the voting application and he would get unique ID for the vote casted.
We can see here the block headers which are connected to the hash blocks of previous block of data. And we can see the hash of previous block data in the structure of the block chain which connected to the root of the block chain and operate it.

There is a root of the user and the hash value of the voter and we can see there are different transaction blocks. Where the different transaction blocks are connected to the upper block of the user and they connect one after the other as shown in the above structure.

4. Implementation

The application is implemented in the springboot framework and the the Cloud database used to implement this model is MongoDB. For stronger encryption the SHA – 256 algorithm is applied on the data.

4.1 Saving vote to the Database

```java
@PostMapping("/addVote")
public String saveVote(@RequestBody UserVote vote) {
    List<UserVote> allVotes = repository.findAll(Sort.by(Sort.Direction.DESC, "_id"));
    for(int i=0;i<allVotes.size();i++) {
        if(allVotes.get(i).getVoterId().equals(vote.getVoterId())) {
            return "Already Voted";
        }
    }
    if(allVotes.size() == 0) {
        vote.setCurHash("justanewhashforgenesisblock");
        vote.setPrevHash("justaprevioushashforgenesisblock");
        vote.setTimeStamp(String.valueOf(System.currentTimeMillis()));
    } else {
        UserVote latestVoteObject = allVotes.get(0);
        String currentHashCalculatedForNewBlock = applySha256(latestVoteObject.getPrevHash() + latestVoteObject.getTimeStamp() + latestVoteObject.getVoterId() + latestVoteObject.getCandidateId());
        vote.setCurHash(currentHashCalculatedForNewBlock);
        vote.setPrevHash(latestVoteObject.getCurHash());
        vote.setTimeStamp(String.valueOf(System.currentTimeMillis()));
    }
    repository.save(vote);
    return "Successfully voted";
}
```

4.2 The Result Evaluation

```java
@GetMapping("/getResult")
public VoteResult getVotingResult() {

```
int nofp = 8;
List<UserVote> allVotes = repository.findAll();
VoteResult finalVoteObj = new VoteResult();
finalVoteObj.setTotalNoOfVotes(allVotes.size());
int a[] = new int[nofp];
for(int i=0;i<allVotes.size();i++) {
    a[Integer.valueOf(allVotes.get(i).getCandidateId())-1]++;
}
Map<String, Integer> pVotes = new HashMap<String, Integer>();
for(int i=0;i<a.length;i++) {
    pVotes.put(String.valueOf(i+1), a[i]);
}
finalVoteObj.setPartyVotes(pVotes);
return finalVoteObj;

There are different data blocks which are connected to their hash functions and the data blocks start sending the hash value to the second level.
In the second level we can see they will combine the two hash functions which are generated from the different blocks.
In the third stage the two hash values will be combined to one and send to the final stage in the root function of the hash.
Now the merkle root hash function combines all the hash values and store in them with the user and give to the voter whenever it is necessary.

**SHA-256 hash function**

4.3 The SHA implementation for secure voting

```java
public static String performSha256(String plaintext) {
    try {
        MessageDigest digest = MessageDigest.getInstance("SHA-256");
        byte[] b = digest.digest(input.getBytes("UTF-8"));

        StringBuffer hexadecimal = new StringBuffer();
        for (int i = 0; i < hash.length; i++) {
```

Figure 9. SHA 256
String h = Integer.toHexString(0xff & b[i]);
    if(hex.length() == 1) hexadecimal.append('0');
    hexadecimal.append(hex);
}
return hexadecimal.toString();
}
catch(Exception e) {
    throw new RuntimeException(e);
}

4.4 Theoretical analysis of SHA 256 algorithm

The SHA[19] algorithm to ensure security of the data is as follows, The message will be divided into blocks of 256 bits each. Each block will be padded with another 256 bits and will undergo this process for n rounds and finally 256 bits hash value is generated. It is the strongest hashing technique as it not irreversible.

Now in the proposed system we can find the different blocks where every block consists of 4 bytes and there we can store a message which we are going to encrypt.
Now the encrypted voting data is send to the merkle root which consists of the hash value of the 20 bytes where it going to store the all hash values in it.
The hash of previous block of the data is present in the block whenever needed we are going to use it. And we have a signature where every voter must needed it which consists of the 32 bytes and all of them are stored in it.
There is voter id and the stamps of the voters in the different blocks and which are of 4 bytes each block.
And the every header block is connected to the other header block and keep on proceeding the process till the last voter.
The smart contract[20] based model could be extends to many applications.

5 Results

![Figure 10. The Implementation Dashboard](image-url)
Conclusion

In this paper we have proposed a novel approach in E voting system in the public cloud which collect the information form the edge devices. The model focuses and the secure storing of data in the public cloud with strong SHA 256 algorithm and applying block chain model to ensure the data integrity in the public model. The implementation model and results are also shared in the paper. The similar approach can be extended further to the applications like E- Health care analysis and E-Auctions and many business models where the data is collected from IoT devices and will be stored in the public cloud.

References

[1] M. Chiang and T. Zhang, "Fog and IoT: An Overview of Research Opportunities," in IEEE Internet of Things Journal, vol. 3, no. 6, pp. 854-864, Dec. 2016, doi: 10.1109/JIOT.2016.2584538.
[2] https://www.forbes.com/sites/forbestechcouncil/2020/09/10/the-us-election-and-the-pandemic-is-e-voting-the-way-forward/#69e4290be650
[3] Dimitrios Zissis, Dimitrios Lekkas, Securing e-Government and e-Voting with an open cloud computing architecture, Government Information Quarterly, Volume 28, Issue 2, 2011, Pages 239-251, ISSN 0740-624X, https://doi.org/10.1016/j.giq.2010.05.010.
[4] Rao, K.T. & Saidhbi, S.. (2017). Data security mechanism in private cloud – A case study. Journal of Advanced Research in Dynamical and Control Systems. 9. 2060-2067.
[5] Pustišek, M., & Kos, A. (2018). Approaches to front-end IoT application development for the ethereum blockchain. Procedia Computer Science, 129, 410-419.
[6] P. Sharmila, Dr. N. Danapaquiame, R. Subhapriya, A. Janakiram, J. Amudhavel. (2018) Secure Data Process in Distributed Cloud Computing, Biosci. Biotech. Res. Comm. Special Issue Vol No 1 (2018)

[7] Lakum, Tarasvi & Rao, B.Thirumala. (2019). A Key-Ordered Decisional Learning Parity with Noise (DLPN) Scheme for Public Key Encryption Scheme in Cloud Computing. International Journal of Advanced Computer Science and Applications. 10. 10.14569/IJACSA.2019.0101121.

[8] Venkata Krishna Sai, Alapati & Sivaramakrishna, N & Teja, Ravi & Kolla, Bhanu. (2019). A Hybrid Approach for Enhancing Security in IOT using RSA Algorithm. HELIX. 9. 4758. 10.29042/2019-4758-4762.

[9] Chandra, J.Vijaya. (2019). Authentication and Authorization Mechanism for Cloud Security. 10.35940/ijat.F8473.088619.

[10] Naga Divya, Lakshmi & Ahamad, Shakeel. (2019). An analysis of cloud security models in real-time distributed computing. International Journal of Innovative Technology and Exploring Engineering. 8. 1753-1757.

[11] S. V., A. Sarkar, A. Paul and S. Mishra, "Block Chain Based Cloud Computing Model on EVM Transactions for Secure Voting," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 1075-1079, doi: 10.1109/ICCMC.2019.8819649.

[12] Haseeb, Sekhar, K.R., Spandana, Y.V., & Syam, M. (2018). A fabric architecture towards block chain application using hyper ledger. International journal of engineering and technology, 7. 83.

[13] Kumar, Dr & Ali Hussain, Mohammed. (2018). A Review on Employee’s Health Monitoring System using IOT. International Journal of Engineering & Technology. 7. 988. 10.14419/ijet.v7i2.7.11608.

[14] T. Halabi, M. Bellaiche and A. Abusitta, "Cloud Security Up for Auction: a DSIC Online Mechanism for Secure IaaS Resource Allocation," 2018 2nd Cyber Security in Networking Conference (CSNet), Paris, 2018, pp. 1-8, doi: 10.1109/CSNET.2018.8602979.

[15] P. Aparna and P. V. V. Kishore, "Biometric-based efficient medical image watermarking in E-healthcare application," in IET Image Processing, vol. 13, no. 3, pp. 421-428, 28 2 2019, doi: 10.1049/iet-ipr.2018.5288.

[16] https://techcrunch. com/2018/02/24/liquid-democracy-

[17] L. Wei and W. Lanjia, "The Nature of Block Chain Intelligent Contract," 2020 15th International Conference on Computer Science & Education (ICCSE), Delft, Netherlands, 2020, pp. 805-810, doi: 10.1109/ICCSE49874.2020.9201768.

[18] S. Janakiraman, K. S. Sree, V. L. Manasa, S. Rajagopal, K. Thenmozhi and R. Amirtharajan, "OTP on Demand - An Embedded System for User Authentication," 2018 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, 2018, pp. 1-5, doi: 10.1109/ICCCI.2018.8441400.

[19] Gilbert, H., & Handschu, H. (2003, August). Security analysis of SHA-256 and sisters. In International workshop on selected areas in cryptography (pp. 175-193). Springer, Berlin, Heidelberg.

[20] Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. (2016, October). Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 254-269).

Acknowledgements
Authors wish to express the assistance and encouragement from the department of computer science and engineering, Koneru Lakshmaiah Education Foundation.