Secure E-Donation System using Blockchain Technology

Anant Kakrania, Kakelli Anil Kumar

Abstract*: It is highly common for organizations to receive anonymous donations. Financial resources with centralized leadership are far easily prone to corruption. There is a need to decentralize the distribution of resources. This can be achieved using blockchain. It is a transparent, decentralized and distributed technology which operates without a central control organ. The main purpose of this system is to eliminate the corruption in organizations by means of which, black money is legalized or resources are manhandled. The system will be decentralized, not owned or operated by a single person or organization but rather shared among users. We have developed a blockchain-based Donation System capable of decentralizing the resources used by a given organization. The user may create any new organization and others will be able to donate money in the organization in the form of ether. This resource can only be used by the organization after getting consent from the users in network. This makes the people the actual owners of the resources and brings transparency to the entire process of charity fund system. Thus, making wrong and illegal use of money donated to organizations will be minimized to a great extent and thereby, corruption will be reduced with the use of this blockchain system, solving problems in many nations.

***Keywords*: Blockchain, Decentralization, Donation, Ethereum, Solidity.**

I. INTRODUCTION

Donations for public welfare, NGOs, orphanages, organ transplantation are the means by which children, old age people, handicapped and the less privileged ones are able to receive facilities which they won't be able to get otherwise. However, this system must work very honestly and with no corruption to have actual benefits towards the betterment of society [1]. Owners and managers of these organizations may turn out to be corrupt and use the working of anonymous donation as the main means for corruption on large scale. Let us consider a situation to explain the corruption in this system. A person wants to sell his property. Suppose the profits from selling the property is subjected to 20% tax. So, let us suppose, the person bought the property for INR 30 lakhs a few decades ago. Now he realizes that the market price for the same property is close to INR 1 crore. Now when he sells this property, his profit may gain INR 70 lakhs. However, he has to pay 20% of property tax as per the government norms in India which may vary for other countries as per their federal rules. Seller realized that almost INR 20 lakhs from the above amount must be paid as taxes. Now he may proceed further and inform his buyer to pay INR 50 lakhs for the sale which is taxable, and expect the rest of INR 50 lakhs raw money which is non-taxable. Hence, the house is sold for the cost of INR 50 lakhs on officially and well documented as per the government norms, and his tax obligations have reduced substantially. Now, he may receive remaining INR 50 lakhs without paying government taxes and without any transaction proof. This unaccounted amount is considered as black money which is biggest financial threat to many countries. However, the black money holder decides to make the amount legal using the tie-ups with some Non-government organizations (NGO). Now the question is, how this black money is getting legal with the help of an NGOs? A Non-government organization gets anonymous donations every day from various sources. So, it can accept the donation and may show it as an anonymous donation. Later, they could make a fictitious expense (purchase of products etc.) for the same amount, and return the money back to donator with deduction of their own charges considered as bribe [2, 3].

Blockchain, by definition, is a list of blocks (or records) that are linked together with the help of cryptography. It is a kind of decentralized database that is named as a distributed ledger. It keeps computerized records that are of what is claimed by whom, exposed to ceaseless updation. Unlike customary database frameworks like in bank or governments, there is no central head in a distributed record, which rather has a network of database copies that are synchronized over the web and are noticeable to any individual who is a part of the network. Blockchain networks are chiefly of two kinds: Private networks that have confined enrolment like that of an intranet and Public networks, that are open to everybody, like the Internet. The property of immutability and consensus among all is the reason blockchain is the best technology for our use case. Blockchain can be used in the decentralization of any resources in which transparency is required. It can be used to curb corruption in places where a central control is monitoring the resources without any knowledge to the users. Ethereum [4] as shown in Fig. 1 is a public blockchain network which provides functionalities termed as smart contracts. It is open-sourced and provides a virtual environment and an operating system for compilation as shown in Fig. 2. It also provides the virtual machine known as EVM (Ethereum Virtual Machine) for running the smart contracts. It provides a lot of test network and environment like Ropsten, Kovan and Rinkeby for development and testing projects on small to medium scale. Smart contracts is written in Solidity which is a very simple and effective object-oriented programming language which is used for developing smart contracts for various blockchain platforms.

Revised Manuscript Received on October 15, 2019.

Anant Kakrania, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, TN, India. Email: anant.lk.1997@gmail.com

Dr. Kakelli Anil Kumar, Associate Professor, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, TN, India. Email: anilsekumar@gmail.com

Smart contract [5, 6] is a block of code which is stored on the blockchain. It provides with a lot of functionalities and events which can be used to interact with users. It being a part of the blockchain is immutable and can only be accessed by nodes of the blockchain network. To protect the system from malicious users the execution of every transaction includes transaction fee, referred to as Gas in Ethereum. Gas is the measure of the unit of work which is accomplished for any operation and the prices is measured in terms of ether.

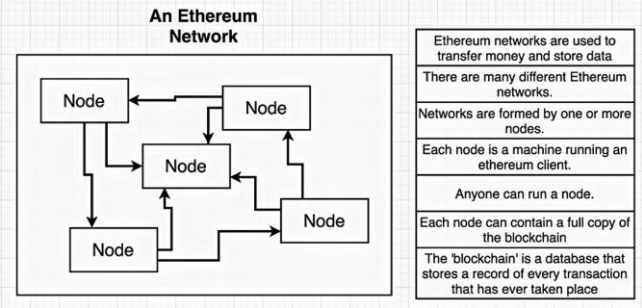

Fig. 1. A basic Ethereum network

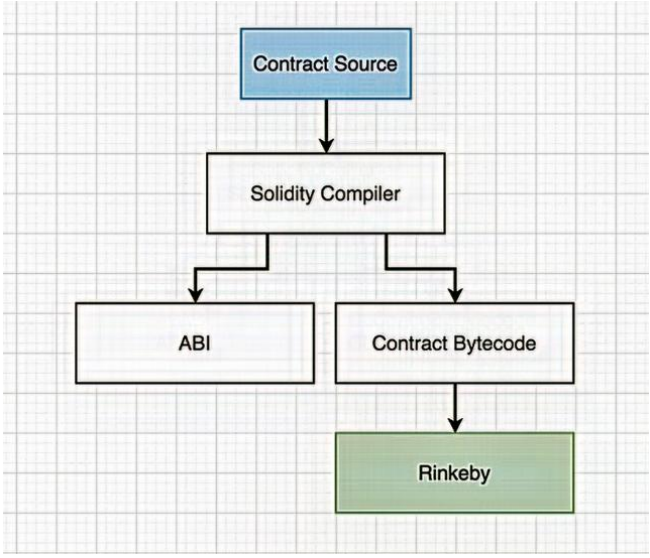

Fig. 2. Compilation of Solidity smart contract

Our proposed system uses similar concepts and technologies to approach the above problem. In general, we are not able to keep track of the money after donation. The proposed mechanism allows us to keep track of the money and how it is getting used further. We are able to see all possible activities and mitigate the corruption by taking the decision-making task in our hands.

II. RELATED WORKS

A. Smart Contract for Boardroom Voting with Maximum Voter Privacy

The usage of a decentralized and self-counting E-Voting system [7] enabling maximum voter privacy utilizing the Blockchain technology. The open vote network is appropriate for meeting room decisions and smart contracts for Ethereum is used for implementing this concept. It doesn't depend on any centralized authority for computing the count or for maintaining the security of the voter's privacy. The open vote network, as proposed in this paper which is a self-tallying system, and every voter is responsible for the security of their vote with the end goal that it can only be breached by full consensus including every single other voter. The implementation of the protocol is authorized utilizing the consensus mechanism that additionally verifies with the Ethereum blockchain. The execution is tried on Ethereum's social test network environment to exhibit its plausibility.

B. Decentralizing Privacy: Using Blockchain to Protect Personal Data

Recently, reports of incidents involving security and surveillance breaches have increased exponentially. These incidents compromised the privacy of users of the existing model which has third-parties perform handling and collection of last amounts of personal data. Bitcoin has been shown to demonstrate a trusted system that can be audited by using a network that is decentralized and involves peers and a public ledger. The authors have proposed [8] an automation of access-control manager, independent of a trusted third-party, using blockchain. This system differs from Bitcoin as the transactions are majorly used to carry out instructions like storing, sharing and querying data whereas, in Bitcoin the transactions are financial. The paper also dives into a discussion of future works that could be done which could improve blockchain as a solution to trusted computing problems.

C. Blockchain for IOT security and privacy: The case study of a smart home

A case study about utilization of blockchain innovation in Internet of Things [9]. Privacy and Security of IOT stays a major hurdle towards its development and scope of utilization. This is because of the extensive scale and distributed character of IOT networks. Blockchain-based methodologies helps in bringing decentralized security and privacy. However, they include significant delay and computational overhead that makes it not very much appropriate for most resource-limited IOT gadgets. In this paper, they emphasize on the different core components and their functionalities in smart home tier. In every smart home there is a centralized human resource device which takes care of all the internal and external communications required for the system to work well. This device is termed as miner. The miner also maintains a secure and private blockchain for this system which is used for auditing and controlling the communication among the nodes. They also analyze the security of this blockchain technology based smart home framework with respect to the fundamental security goals of confidentiality, integrity, and availability. They also present the simulation result highlighting the overhead of this system which is very insignificant when compared to the security and privacy gains of this system.

D. Designing Secure Ethereum Smart Contracts: A Finite State Machine Based Approach

The authors [10] have implemented a structure that will be used to write smart contracts that are dependent on Finite State Machines (FSM). This structure has been termed as FSolidM. The number of computing problem solutions using blockchain as a distributed network, is increasing rapidly. One of the blockchain platforms is Ethereum which supports implementation of smart contracts that assist applications in various different fields such as finance and IOT (Internet of Things).

However, with the deployment of smart contracts comes security issues and vulnerabilities. Malicious users involved in the network can steal data and assets of other users that are of financial or personal importance. The system FSolidM proposed by the authors provides a more secure and less vulnerable solution implementation of smart contracts with the use of FSM.

III. PROPOSED ARCHITECTURE

The architecture of this system as shown in Fig. 3 is a layered model. Each layer works independently and interact based on the working model of the project. The layered architecture helps us to give modularity and abstraction to our architecture.

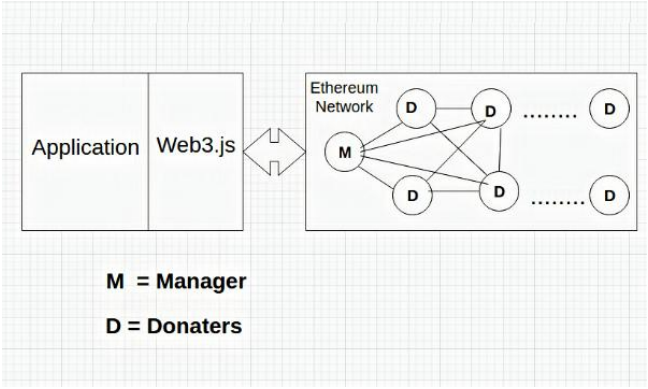

Fig. 3. Overview of our system

A. User Interface

The client side of our project is developed using React JS [11] which is component-based frontend framework. All the functionalities provided by the smart contract can be accessed using our user interface with access control of various functionality based on the node address.

B. Web3 JS

Web3.js [12] is a JavaScript API which is a collection of modules which contains specific functionalities of Ethereum Ecosystem. It is used to connect with local or remote Ethereum node using HTTP, WebSocket, IPC connection [13]. In our proposed system we our using Meta-Mask Tool for providing Ethereum ecosystem for our browser. It is a tool used to access the Ethereum Network without actually setting up Ethereum Node in our system. It injects Web3 JS API in all the websites that we visit using our browser and allow us to access the Ethereum Network with Node address provided to us by Meta-Mask.

C. Infura API

Infura-API [14] is the platform which provides us with Ethereum Infrastructure. It provides secure and reliable way for nodes to access the Ethereum network. In our project we deploy our smart contract on Infura use the API ID provided by the Infura API as address of our smart contract.

D. Smart Contract

It is the consensus algorithm that we develop for our blockchain network using Solidity programming language. In our project we have two contracts. One for the main consensus algorithm based on which every transaction is approved. Second to store the address of all the deployed contracts on our network. In this architecture of our contract the factory contract is the abstract layer for all the other contracts. This also helps in bringing a layer of abstraction in our architecture. The user interacts with factory to get the details of the organization it wants to interact with and then it can interact with the organization directly. The simple user interaction is represented as shown in Fig. 4.

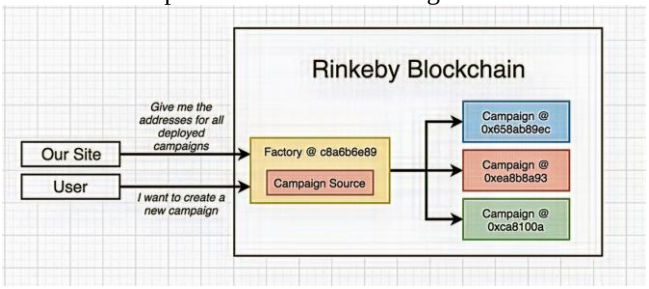

Fig. 4. User interaction with smart contract

The are two contracts in our model are:

- **Organization-Factory**- Stores the address of all the organization deployed in the network.
- **Organization-** It is used for the core logic of the consensus algorithm of our proposed solution.

The smart contracts functionalities can be described as given below:

- **Add Organization:** This function is involved in the creation of a new organization. The details of the organization like its name, description, minimum contribution value is passed as parameters. In the network, the node which creates the organization is the manager of the organization. The details are passed to the Organization-Factory contract which creates a new organization instance in the memory and stores the initial details the manager sends for organization creation. The new contract instance is provided with an address in the network storage which is stored in the Organization-Factory contract.
- **Get Summary:** This function is used to get the current state of an organization. It returns the value of various variables used to get information about any organization like the donation amount collected, total number of pending requests, number of contributors and minimum contribution details.
- **Contribution to the Organization:** This function is to accept donations from users. Donators donate some amount of money to organization such that amount donated > min contribution. Once donated they form the part of the blockchain network. One important thing to take into account is that the money in our project donated is in the form of ether tokens. The address of the donators gets added to the list of donators and the total amount of money for the organization gets updated.
- **Request Generation:** In this function the manager creates a request for spending money for the organization for some purpose. It adds all the details of the request regarding the description of the request, amount required and to whom to send the money for this request. This helps in adding the request to the pending requests queue.

- One thing to be noted is that, this request generation function can only be initiated by the manager of the organization which is checked by the address of the node from which the request is arriving.
- **Request Distribution:** This function is the main component of the contract because unlike the conventional blockchain this function helps in constraining the number of approvers for any transaction increasing the efficiency of the entire system. This function takes as a continuation of request generation. In general, the request for spending money must be sent to all the donators in the network waiting for them to approve it. However, the number of donators increases day by day and can reach a very large number. Then for such cases, the approval of even valid requests may have a very low probability. In this function request is sent to 'm' donators in sequence where 'm' is 'x %' of the total number of donators in the organization. In the next subsequent requests, it shifts to the next 'm' donators and this continues in circular queue manner for the next subsequent requests. The value of 'x' changes as the number of donators increases to keep the request approval rights in hands of proportionally less number of people. This is to keep the probability of valid requests being approved high. This entire request generation and distribution components is what decides the security and efficiency of our system.
- **Request Approval:** This function is for the request approval of the request generated by the manager. In this the donators who get approval rights for request will get the list of requests they have rights to approve to. They can analyze the validity and genuineness of requests and hence approve/reject based on that.
- **Request Finalization:** This function can only be called upon by the manager. It is like a confirm button for a request to get processed. This function only gets enabled when the request gets approved by the donators. The manager may decide not to process the request for some reason and will have a chance to reject it. Once the request is finalized it gets processed and the money is transferred to the address specified by the manager.
- **Deployment:** This is very important for any organization to become publicly available. We create a contract instance Organization-Factory which takes into account creation and deployment of organization instance in our network. Once deployed as shown in Fig. 5, it can be interacted directly by the users. However, the address of the deployed organizations are stored in the Organization-Factory contract.

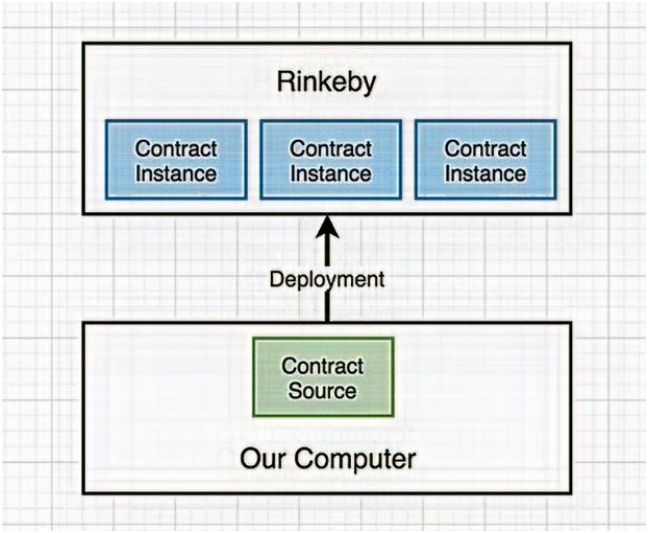

Fig. 5. Deployment of smart contract

IV. WORKING

In the entire system we are using various technologies to bring up a system with simple user interface and efficient working. We have used NextJS module for server-side rendering of the ReactJS Components. This is done for making up the loading of data into the user interface faster and smooth. NodeJS [15] is used for setting up a simple server for rendering of our client-side components. We setup the server, run it on our machine which loads the pages of the user interface. The user has options for joining an organization as a donator or creating one as manager. For example, a user creates an organization as a manager for helping cancer patients. This organization is now public and can be viewed by anyone on our portal. Let us say N donators donate money in the form of ether to this organization. They become part of the blockchain network for that organization instance. Now, the manager feels a need to spend money for renovating the NGO campus and creates a request with all the necessary details as shown in Fig. 6. This request gets added in the pending request queue and gets send to first 'm' donators for approval.

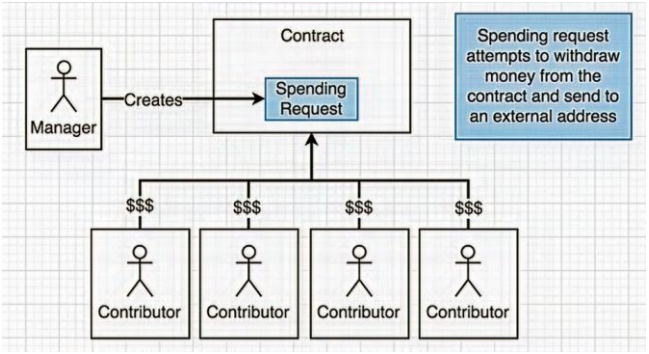

Fig. 6. Request creation by manager for money retrieval

Each donator who receives request can see the list of requests they received with approval rights. Currently, the other donators are not aware of any request sent to other members. On the organization dashboard they can see total number of requests generated for that organization but cannot see who actually received it.

The next request for the organization will send to the next set of members and will circularly rotate among the list of donators. The donators who receive the request can approve/reject the request. However, even one donator rejecting the request will lead the request to not get processed. Hence, this will fail our entire motive to increase the chance of valid request getting processed. For this scenario, every request requires 50% approvals from the donators as shown in Fig. 7 making the probability of request getting accepted higher. The above request gets approved and hence can be confirmed by the manager for processing of the request. The request is now finalized/cancelled by the manager. If finalized, the money gets transferred to the vendors account. Once, the request is processed by the manager, it becomes visible to all the donators. This actually brings complete transparency to the system with approval right shifting from groups every time a new request is generated. This entire system serves in

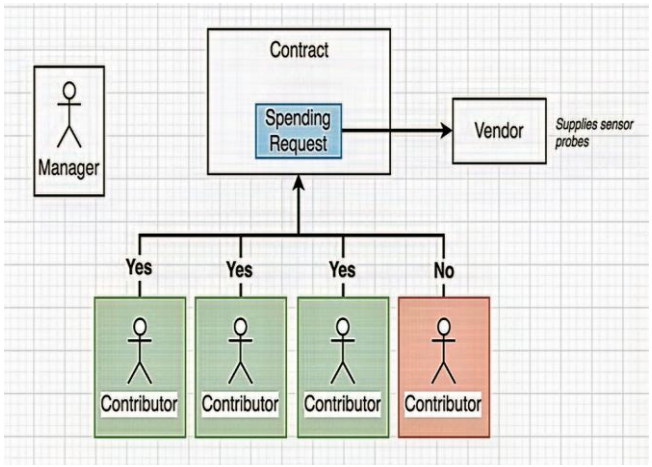

Fig. 7. Request approval by contributors

one way to bring transparency, efficiency in our system with no concentration of power in centralized group of people.

V. RESULTS

In this system we see that the adding of block in blockchain takes around 8-11 seconds in the test network which is slow as per efficiency of the system is concerned. This kind of efficiency as per the transaction speed can be increased if the computation resource is increased or transactions are parallelized. The various views of the resulting system are shown from Fig. 8 to Fig. 11.

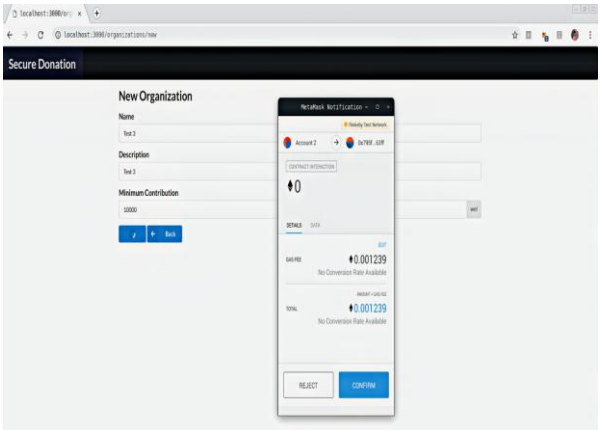

Fig. 8. Adding organizations

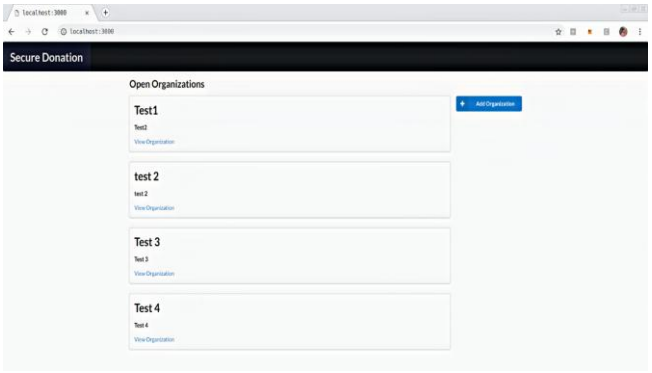

Fig. 9. List of created organizations

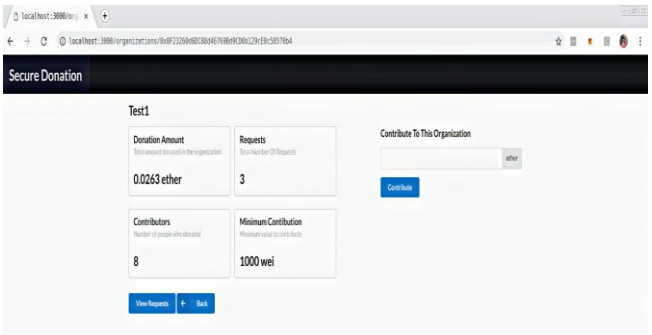

Fig. 10. Organization dashboard

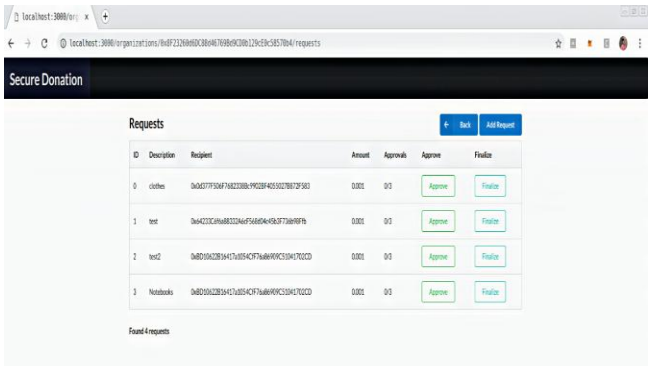

Fig. 11. List of requests allotted to a node in the network

VI. CONCLUSION AND FUTURE WORKS

In this entire system one question may arise as to how is it solving our problem? Let us consider that a rich person donates money to the NGO and the manager plans to return that money to the person by showing a fake transaction. In our decentralized application, he will first have to create a request. As the request is approved by the people, the possibility for the fake request getting approved is negligible. However, if by any chance the transaction is approved by the people, considering that the set of people who get the request approval rights for this request are themselves corrupted, the request is recorded as it is added to the blockchain. Hence, the person will have to pay tax for that money anyway. Thus, either way the motive of the corrupted person fails. This money cannot even be used for any other illegal activities, as same condition applies on every transaction. This system can be made efficient when we parallelize the transactions that are queued for getting processed on the network.

However, till now there is no particular way to speed up the transactions except the experiment of increasing resources to process the request faster. In future works, we will try to make this system efficient as per the speed of transactions is concerned. Our proposed system serves a very important purpose if implemented and accepted by the government. Hence, if the system is utilized to its maximum potential, it will help bring down corruption to a large scale.

REFERENCES

1. Z. Zheng, S. Xie, H. N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," International Journal of Web and Grid Services, 2018.
2. S. Doshi and M. Ranganathan, "Contesting the Unethical City: Land Dispossession and Corruption Narratives in Urban India," Annals of the American Association of Geographers, 2017.
3. Dhanpal Singh, "Different Dimensions of Corruption In India: Some Suggestions For Prevention," Research Journal of philosophy and social sciences, vol 42, No 1, 2016.
4. A. K. Koç, E. Yavuz, U. C. Çabuk, and G. Dalkiliç, "Towards secure e-voting using ethereum blockchain," in Proceeding 6th International Symposium on Digital Forensic and Security, ISDFS, 2018.
5. L. Luu, D. H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in Proceedings of the ACM Conference on Computer and Communications Security, 2016.
6. Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," IEEE Internet of Things Journal, 2019.
7. P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2017.
8. G. Zyskind, O. Nathan, and A. S. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in Proceedings IEEE Security and Privacy Workshops, SPW, 2015.
9. A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017, 2017.
10. A. Mavridou and A. Laszka, "Tool Demonstration: FSolidM for designing secure ethereum smart contracts," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018.
11. S. Aggarwal, "Modern Web-Development using ReactJS," International Journal of Recent Research Aspects, 2018.
12. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, 2013.
13. K. Ma and R. Sun, "Introducing websocket-based real-time monitoring system for remote intelligent buildings," International Journal of Distributed Sensor Networks, 2013.
14. M. Wuehler, "Infura Dashboard Update," Infura Blog, 2018.
15. N. Chhetri, "A Comparative Analysis of Node.js (Server-Side JavaScript)," Culminating Projects in Computer Science and Information Technology, 2016.

AUTHORS PROFILE

Anant Kakrania, belongs to School Of Computer Science and Engineering from Vellore Institute Of Technology, Vellore, Tamil Nadu, India. His research interests are Distributed Systems, Blockchain, Cryptocurrency and web Security. He has undertaken many research projects under the Web Development domain. He worked as a Web Developer Intern in Softech ERP Solutions Pvt. Ltd., Kolkata, India. He is currently the Technical Head in IEEE Computer Society, VIT. He is very enthusiastic about emergin computing technologies and works passionately towards the betterment of his skills in every possible technical domain.

Dr. Kakelli Anil Kumar, working as Associate Professor at School of Computer Science and Engineering (SCOPE), Vellore Institute of Technology, Vellore, Tamil Nadu, India. His research interests are secure protocol design in IOT and wireless sensor networks, secure cloud computing services, block chain and cryptocurrency and digital forensics. He has published over 25 research articles in reputed international journals and conference.