Blockchain Secured Distributed Computing for Low Resource Edge Devices in IoT Systems

Manthan Maheshwari
Electronics and Instrumentation Engineering
SRM Institute of Science & Technology
Chennai, India

***Abstract*—Edge Devices in IoT systems have limited computing power, run on primitive operating systems, and mostly communicate through typical wireless IoT network protocols. IoT systems may require high computation power, large bandwidth connection, and rely mostly on the third party for centralized database and computation which causes latency and exposes the system to security loopholes. With the increasing adoption of IoT systems, there is a need to localize computing and reduce the need for large bandwidth connections. This paper proposes a model for distributed computing in low-resource edge devices connected in an IoT system secured with a decentralized private blockchain. The model uses a unique identity key for node identification, a public key for blockchain, and proof-of-work. The main objective of the proposed model is to create an IoT system that does not require third-party computation, database, and securing IoT system from illegal access control, as well as allowing for secure communication between local IoT devices.**

Keywords—Edge Devices, Blockchain, Internet of Things, Distributed Computing, Mesh network

I. INTRODUCTION

An Internet of Things (IoT) system refers to an application of IoT technology that is formed by connecting devices, sensors, actuators, computing power, data storage, and networking framework. These systems can provide application on a tiny scale and a global scale and lead revolutionary changes that improve our lives.

IoT systems may consist of IoT devices that have limitations such as limited computing power, limited power resource, etc. These devices can provide information such as sensor data but data is not enough to complete an IoT system. Information gathered needs to be processed and sent to the required devices to complete the IoT system. IoT systems need to get the data in real-time and through a secured medium.

IoT systems use cloud infrastructure for reliable communication between devices, computing power, and data storage. Cloud infrastructure possesses challenges like high latency, the need for large bandwidth, and a centralized database. To overcome the problem of latency we have some solutions like mobile cloud computing, fog computing, and edge computing. These solutions create another layer between devices and the cloud for data processing and storage. This brings data processing near to end devices and requires additional infrastructure for the middle layer. Resource-constrained edge devices can’t be used in the edge layer to attain the required benefits of the architecture.

In this paper, I have designed a model to use low-resource devices taking part in IoT systems in the application layer or edge layer. This model reduces dependency on cloud infrastructure and the need for centralized computational resources and large bandwidth.

The main features of this model are mentioned below:

- This model utilizes limited computing power offered by IoT nodes and resource-constrained devices that are used to provide a gateway for cloud connection. These devices are connected in a network in a mesh topology.
- A private, permissioned blockchain structure is proposed in this model to create a decentralized IoT system that provides secured communication between devices, secured access control, and distributed computational resources without significant overhead.
- The model distributes computational tasks to the participating devices with help of smart contracts [4] and maintains a world states for the processes. It also provides access controls and task creation to participating devices through consensus.

This model improves the IoT system’s security and meets the requirements of resource-constrained IoT devices. It is a generic, lightweight, and scalable solution that can be applied to various IoT applications.

The remainder of this paper is structured as follows: Section 2 provides a brief overview of the blockchain and smart contract technology in IoT. Section 3 discusses the related works in IoT systems. The proposed architecture is explained in-depth in Section 4. Finally, Section 5 concludes the paper and discusses future work.

II. OVERVIEW OF BLOCKCHAIN & SMART CONTRACT TECHNOLOGY

Blockchain & its use in IoT: Blockchain is a data structure that stores transactions in an ordered way and is linked to the previous block, serving as a distributed system of records. This structure is divided into two parts, header and transactions, and stores detailed information about the transactions it contains. It can associate a transaction with its source and destination address. Each block has a unique ID generated from a cryptographic digest as explained in the previous section. The header has a field that stores the hash of the immediately preceding block so that we can establish a connection, a “link,” between the blocks. For this reason, this structure was called Blockchain. Blockchain is an authenticated, synchronized distributed ledger saved and self-maintained by each node in the network [2]. It provides a peer-to-peer network without the interference of a third party. Without a central manager in the network, the blockchain nodes complete the verification of a transaction in various ways of consensus such as PoW (Proof of Work), PoS (Proof of Stake), DPoS (Delegated Proof of Stake), PBFT (Practical Byzantine Fault Tolerance), Raft [3].

Blockchain is used in IoT to create a fair authorization framework and to address issues in IoT systems like access control, secured communication, and single point failure which occurs in a centralized system. While most of the system uses public blockchains like Ethereum to secure authentication mechanism to provide trust account to the system, it is not suitable for IoT systems that consist of low-resource devices due to their high bandwidth, high computational power, and latency requirements. Therefore, I use private blockchain architecture in IoT which is lightweight, scalable, and reduces computational overhead.

Smart Contracts & their use in IoT: [1] A smart contract is a function or block of code that is stored inside a blockchain and is executed when specified conditions are satisfied. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without an intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met. The most common application of smart contracts in the field of IoT can be seen in supply management in which multiple untrusted parties are included.

III. RELATED WORK

A survey of the scientific literature shows multiple solutions that address blockchain-based decentralized architectures for the IoT. The most notable examples are:

- Maior et al. [5] present a theoretical description of a decentralized solution for energy management in IoT architectures. The solution is aimed at the application of smart power grids. They present 4 algorithms with analyses of correctness to describe the behavior of self-governing objects.

- Higgins et al. [6] propose a distributed IoT approach for electrical power demand management.

- Algarni, S. [7] designed an architecture that adopts a private hierarchical blockchain structure to improve which includes blockchain managers (necessarily a high resource device) to control access in IoT systems.

- Suzdalenko and Galkin [8] extend the approach by Higgins et al. [6] by allowing users to individually and in run-time join and part the environment.

- Baraka William Nyamtiga [9] proposed a framework for decentralized storage management for IoT systems and used private and public blockchain in a P2P network in the edge layer and among nodes.

- dSUMO [11] address the bottleneck in synchronization by proposing a distributed and decentralized microscopic simulation (the focus is on the data throughput and not so much the fault tolerance, the throughput is increased using decentralized setting).

- Ajayi, O.J [10] used hyperledger fabric and an open-source IoT Edge computing platform, which supports interoperability within heterogeneous devices to create a reliable decentralized system for IoT devices smart contract

- Al-Madani et al. [12] address indoor localization utilizing Wireless Sensor Networks (WSNs) relaying on publish/subscribe messaging model. The results show that the Simple Syndication (RSS) [13] format achieves acceptable accuracy for multiple types of applications.

Our proposed model differs from the previous contributions in two ways.

- Other solutions mostly use either public blockchain like Ethereum or permission blockchain service like hyper ledger fabric which requires membership or money for public blockchain, I argue that an IoT architecture with sensitive data requires completely free-of-cost private blockchain.

- The main contribution of this paper is the proposal of a model for low resource edge devices which can distribute computing among the devices with a minimum number of messages.

A related approach by Samaniego and Deters [14] suggests using virtual resources in combination with a permission-based blockchain for provisioning IoT services on edge hosts. They use blockchain to manage permissions only and therefore provide security using blockchain. But it needs a high-resource edge device to manage permissions. In contrast, our approach uses blockchain to store all information about service which makes it verifiable over time, while still providing security.

IV. PROPOSED MODEL

Communication between IoT devices using typical communication protocols like MQTT or CoAP have several security loopholes as stated in [15]. Even after using security layers like TLS IoT system is susceptible to data injection, passive reconnaissance, and malicious IoT nodes. Creating a distributed computing architecture for IoT by using only the protocol layer security may result in unauthorized intrusions in the network. The network can be brought down through Denial of Service (DOS) attacks.

In the model, I am using a private permissioned blockchain to create an additional layer of security in the IoT network. It facilitates secured communication of data between nodes, secured access in the network, and a cooperative way to distribute computing tasks. The traditional blockchain architecture i.e., public/permissionless blockchain cannot be used in our model because of the requirement of high computational power. In a public blockchain, anyone can become a node but requires high computational power to take part in consensuses. The data and nodes are pseudo-anonymous. The blockchain consensus mechanisms that are used traditionally also require high message frequency to maintain replicated state of the blockchain and verify transactions held from other nodes. This will increase computational overhead and make the low-resource device network very busy.

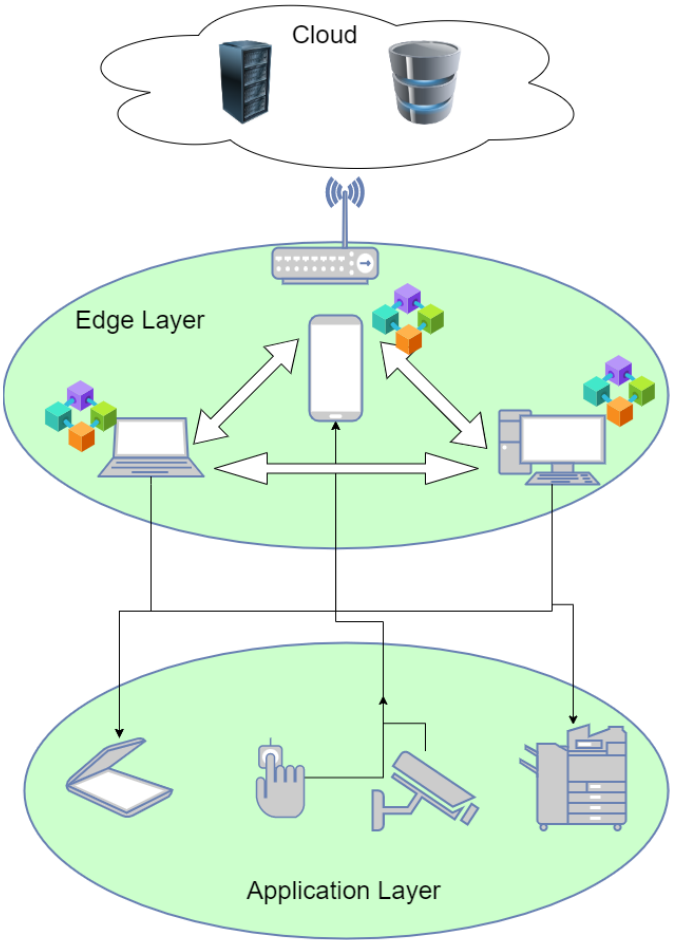

Figure 1: IoT system consisting of three layers with devices connected in a mesh network, devices in the edge layer sharing the same blockchain.

This model focuses on the edge layer in an IoT system which is composed of three layers identified as the cloud layer, the edge layer, and the application layer as illustrated in Figure 1. Devices or nodes that can take part in computational tasks and are connected in the same network are considered to be part of the edge layer connected in a mesh network. This layer is responsible for short-term data storage, real-time data processing, analytics, and handling communications for various data and messages exchanged among the different nodes. These devices may connect the system to the cloud through a gateway. All communication between edge layer devices and communication between edge layer devices and application layer devices happens through the MQTT protocol. Nodes connected in a mesh network increase the speed of communication between devices which helps in completing computational tasks with the least latency. During the creation of this model, we have assumed that the number of edge devices is more compared to the number of tasks in the IoT system and each task occurs in a period greater than the time required to compute it. Also, the nodes in the edge layer have to be programmed for each task and each task can be divided into synchronous subtasks.

Edge devices use asymmetric RSA key cryptography for access control, node identification, and for encrypting messages from the network. Each device maintains a set of topics as a subscriber. All devices in the edge layer subscribe to all the topics in which the application devices can publish. They maintain a set of topics according to its state in the network (it has registered a task or not, etc.) to avoid unnecessarily listening to unneeded messages.

The computational work for each task in the IoT system is divided into subtasks and each device has all the subtasks of all the tasks of the IoT system in form of asynchronous functions that can be executed when needed. The tasks correspond to the set of application devices from which data generates and the end device which needs the data. An end device can also be cloud storage so that data can be stored for future reference and decrease the storage need from edge devices.

The edge devices maintain a copy of the blockchain which contains device information and computational transactions made between devices. The blockchain is designed in such a way that the size of each block is very minimal and consensus mechanisms are selected to keep the computational overhead as low as possible. Traditionally used consensus mechanisms like Proof of Work (PoW) cannot be used due to the requirement of huge computational power.

The computation process begins when the application layer devices or IoT end-users from whom data originates broadcast the data to all edge devices. All edge devices save the data along with a timestamp and the topic to which the data has arrived. This data is saved until the computational process has been done with that data and the whole process is registered in the blockchain. At the time of receiving data, some devices may be free of any tasks and some may be busy doing a task with previously received data. For starting a task, a leader is elected among the devices which are free of tasks at that moment and registered in the blockchain log.

To elect a leader, there are some commonly used voting mechanisms for permissioned networks like Practical Byzantine Fault Tolerance (PBFT) [16], Proof of Elapsed Time (PoET) [17] or RAFT [18]. These algorithms are not suitable for our model because they require multiple messages to be sent through the network to commit a change. It results in additional overhead in the network. We use a leader election consensus algorithm that only requires one series of messages to commit to changes in the blockchain. The algorithm is based on a lottery ticket system that can be verified universally. A recent paper proposed construction of verifiable random functions (VRF) [19], where a node can compute a VRF f given an input value, and a private key to generate a random value and a universally verifiable proof that random value was indeed the output of the VRF using the public key without revealing the private key. The contesting nodes broadcast a random value and with their public key hash to other nodes which are free of tsks at that moment. Nodes receiving multiple values elect the node as a leader and save its public key in the transaction log.

The elected leader node changes its state and does not take part in upcoming elections unless the current task is completed. The current task is distributed to nodes as subtasks in the form of ID of the executable code or the subtask which is initially stored in the device. The leader node adds subtasks in the transaction log which is verified by the nodes by matching the public key hash of the leader and the leader hash in the transaction log. Leader randomly selects nodes for each task and assigns public keys to each subtask thus smart contracts are formed.

The state of each participating node is maintained in the same transactional log. Four states can be named as Idle, Electing, Leader, and Computing.

- Idle: A node is idle when it is neither involved in any system process nor is a leader of any task. It can take part in leader election, subtask distribution, and verification of transactions and new blocks.

- Electing: When a node is electing a leader for a task it can still take part in the verification of transactions and compute any subtask simultaneously.

- Leader: A leader of any task does not take part in computing other subtasks but can verify transactions and elect other leaders.

- Computing: When a node is selected for computing. It can take part in the election of other leaders but cannot verify transactions.

If any node which has a subtask assigned to it in the network goes out of the network or crashes before changing the state of the smart contract. The leader assigns the subtask to free nodes randomly. The leader detects the node crash if the state for the corresponding subtask does not change in the maximum specified computing time specified in the subtask. In this way, subtasks are computed faster in the case of node crashes. When all the subtasks for a task are completed, the leader processes the state outputs and perform the final subtask which is to publish the output and proposes the block to the block to other nodes with the computed transactions and block header which contains a timestamp, Merkle tree root hash as the block hash, previous block hash, and the task IDs for which transactions are stored in the block. Other nodes check the public key hash of the leader and the hash in the block to check that it is from the leader only. The data is not encrypted thus it can be retrieved for future reference using a timestamp and task ID in the blockchain.

In this case, if the leader node crashes in the middle of the computing process of a task, then the participating nodes re-elect a leader among them and add the public key of the selected node as a new leader and the already processed subtasks are added in the same transactions. The newly elected leader takes over the process and handles the task as stated above.

The access is controlled with asymmetric keys which are unique to each device. These keys are added to the blockchain in the genesis block. To add a new node in the network it is equipped with all the executable functions and public keys of all the existing devices. It makes a request from free nodes for the latest blockchain. This completes the distributed computing system with low-latency, computational load on a single device and maintains the IoT system in a decentralized secured system.

V. CONCLUSIONS AND FUTURE WORK

The issues related to the security and privacy of the IoT system are immense and require careful consideration. There are some pros and cons to both centralized and decentralized solutions. Centralized solutions are constrained by scalability, while decentralized approaches are bound by delays, computational overheads, and energy constraints. I proposed a voting-on-need model to provide lightweight, decentralized IoT access control security mechanisms.

The proposed model is a generalizable solution that can be applied to various IoT applications. Furthermore, IoT issues are not fully addressed in prior studies, as most studies focus on addressing access control issues specifically in IoT systems.

I understand that research evaluation is must be based on implementation and testing phases that specify the solution's applicability and effectiveness compared to related research. However, this research is still in progress and I believe that the results of these two phases should be published in a separate paper due to the expectation that a great number of details will need to be discussed, as well as new contributions.

In future research, the proposed model will be implemented in a real environment to measure the achievement of the fundamental security goals about integrity by applying the digital signature, authentication via asymmetric keys, consensus via a voting mechanism, and confidentiality via public key encryption. The RaspberryPI IoT device will be used for the deployment of the model. Further details will be included in future works.

REFERENCES

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [4]. Papers that have been accepted for publication should be cited as "in press" [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

[1] Introduction to smart contracts | ethereum.org

[2] X. Wang, L. Feng and H. Zhang, "Human Resource Information Management Model based on Blockchain Technology", Service-Oriented System Engineering (SOSE 2017), pp. 168-173, 2017.

[3] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm", Usenix Conference on Usenix Technical Conference (USENIX Association), pp. 305-320, 2014.

[4] V. Dwivedi, A. Norta, A. Wulf, B. Leiding, S. Saxena and C. Udokwu, "A Formal Specification Smart-Contract Language for Legally Binding Decentralized Autonomous Organizations," in IEEE Access, vol. 9, pp. 76069-76082, 2021, DOI: 10.1109/ACCESS.2021.3081926.

[5] Maior, H.A.; Rao, S. A self-governing, decentralized, extensible Internet of Things to share electrical power efficiently. 2014 IEEE International Conference on Automation Science and Engineering (CASE). IEEE, 2014, pp. 37–43.

[6] Higgins, N.; Vyatkin, V.; Nair, N.K.C.; Schwarz, K. Distributed power system automation with IEC 61850, IEC 61499, and intelligent control. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 2011, 41, 81–92

[7] Algarni, S.; Eassa, F.; Almarhabi, K.; Almalaise, A.; Albassam, E.; Alsubhi, K.; Yamin, M. Blockchain-Based Secure Access Control in an IoT System. Appl. Sci. 2021, 11, 1772. https://doi.org/ 10.3390/app11041772

[8] uzdalenko, A.; Galkin, I. Instantaneous, short-term and predictive long-term power balancing techniques in intelligent distribution grids. Doctoral Conference on Computing, Electrical and Industrial Systems. Springer, 2013, pp. 343–350.

[9] Nyamtiga, B.W.; Sicato, J.C.S.; Rathore, S.; Sung, Y.; Park, J.H. Blockchain-Based Secure Storage Management with Edge Computing for IoT. Electronics 2019, 8, 828. https://doi.org/10.3390/electronics8080828

[10] Ajayi, O.J.; Rafferty, J.; Santos, J.; Garcia-Constantino, M.; Cui, Z. BECA: A Blockchain-Based Edge Computing Architecture for Internet of Things Systems. IoT 2021, 2, 610-632. https://doi.org/10.3390/iot2040031

[11] Bragard, Q.; Ventresque, A.; Murphy, L. Self-balancing decentralized distributed platform for urban traffic simulation. IEEE Transactions on Intelligent Transportation Systems 2017, 18, 1190–1197

[12] Al-Madani, B.M.; Shahra, E.Q. An Energy-Aware Plateform for IoT Indoor Tracking Based on RTPS. Procedia computer science 2018, 130, 188–195.

[13] Teh, P.L.; Ghani, A.A.A.; Chan Yu Huang. Survey on application tools of Really Simple Syndication (RSS): A case study at Klang Valley. 2008 International Symposium on Information Technology, 2008, Vol. 3, pp. 1–8. doi:10.1109/ITSIM.2008.4631980.

[14] Samaniego, M.; Deters, R. Using blockchain to push software-defined IoT components onto edge hosts. Proceedings of the International Conference on Big Data and Advanced Wireless Technologies. ACM, 2016, p. 58.

[15] S. Shapsough, F. Aloul and I. A. Zualkernan, "Securing Low-Resource Edge Devices for IoT Systems," 2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI), 2018, pp. 1-4, doi: 10.1109/ISSI.2018.8538135.

[16] Castro, M.; Liskov, B.; others. Practical Byzantine fault tolerance. OSDI, 1999, Vol. 99

[17] Chen, L.; Xu, L.; Shah, N.; Gao, Z.; Lu, Y.; Shi, W. On security analysis of proof-of-elapsed-time (poet). International Symposium on Stabilization, Safety, and Security of Distributed Systems. Springer, 2017.

[18] Ongaro, D.; Ousterhout, J. In search of an understandable consensus algorithm. 2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14), 2014.

[19] Micali, S.; Rabin, M.; Vadhan, S. Verifiable random functions. 40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039). IEEE, 1999.