Decision Tree for the Secure and Scalable Collaborative Edge Computing

Deepak Puthal*, Ernesto Damiani*, and Saraju P. Mohanty†
* Center for Cyber-Physical Systems and Department of EECS, Khalifa University, Abu Dhabi, UAE
† Department of Computer Science and Engineering, University of North Texas, Denton, Texas, USA
Email: {deepak.puthal, ernesto.damiani}@ku.ac.ae, saraju.mohanty@unt.edu

***Abstract*—Edge computing is an essential step of an IoT-Edge workflow, where the Internet of Things (IoT) devices sense or collect the data, and data are processed and evaluated at the edge data centers (EDCs) for near real-time evaluation. Further, data are processed in the resource-rich Cloud environment. There is always challenging to filter out the corrupted data at the resource constraint EDCs. To address this challenge, we proposed a method to create a collaborative edge computing environment and apply the Decision Tree (DT) to filter out the corrupted data and make the unchanged data available for data processing. Secondly, we developed a Decision Tree-based authentication mechanism to robust the Proof-of-Authentication (PoAh) process for seamless integration of Blockchain in IoT-Edge workflow for the secure end-to-end data transmission. Finally, we have experimented in a real-time testbed to validate both the proposed approaches.**

***Index Terms*—IoT-Edge Workflow, Collaborative Edge Computing, Decision Tree, Data Filtration, Blockchain.**

I. Introduction

As the technology advances, it's made even more accessible and inexpensive to integrate into everyday items such as washing machines, lights, ovens etc. This has provided everyday items the ability to connect to a network to send and receive data, referred to as the Internet of Things (IoT) [1].

Originally, the IoT devices (IoTD) were designed to send data over to the cloud for analysis/processing, this structure worked fine until the technology became inexpensive, low powered and easy to integrate. With the number of IoTDs around the world reaching 30 Billion it would require sending trillions of Gigabytes of data to the cloud for processing and would cause massive clogs in the pipeline [2]. To address this challenge, the IoT-Edge-Cloud structure, was developed (Fig 1). Based on the structure, the low powered IoTD can send the data over to an edge data center (EDC) in the local network for processing [3]. This reduces the load on the cloud significantly as most, if not all, of the processing can be achieved on the EDC.

The introduction of edge computing also reduced the latency in the network and enable faster response times as the IoTD don't necessary require an internet connection, but can simply be connected to a local network (Intranet) with the EDC also acting as a gateway to the internet. This workflow is most common in smart home devices [2], where all the IoTD in a house are controlled by a smart hub (edge device/data center) which is in turn connected to the cloud through internet. This enables the IoTD within the home network to work optimally even if the Internet connection is down as they are still connected to the smart hub (edge device) and can send and receive data/instructions [1].

A more important application of the IoT-Edge-Cloud workflow is in the industry sector, such as manufacturing, production lines, inventory management, shipping, etc. where many IoTD are used to manage the process involved and reduce downtime [4]. In these scenarios, the EDCs work along with the cloud platform for a more effective management of the industrial processes by performing edge based computation for a real-time analysis and only sending pre-processed important information to the cloud for high level analysis and management [5].

One of the most effective approaches could be establishing a collaborative environment in the edge layer to achieve better performance [6]. Multiple EDCs can combine their resources based on their capabilities to serve the requests from sources, devices, or users. In parallel, taking care of the pure form of the data is vital to extracting information from these data. Data security in the end-to-end process is essential to make the data filtration at the edge simplified or not overloaded. Considering the loop falls of the current security solution, i.e., single-point failure, Blockchain is one of the best solutions for the present time to secure the system [7]. The contribution of this paper is broadly in two parts and summarized as follows.

- A decision tree modeled solution is proposed to filter out the corrupted data in a collaborative edge environment.
- Decision tree-based blockchain consensus solution for the decentralized security in IoT-Edge workflow.

The following section (i.e., Section II) provides the background studies and formulation of the problem. Section III presents the data filtration in a collaborative edge computing environment and experiment studies. In Section IV, we have prosed the decision tree-based secure blockchain consensus. The theoretical and experimental validation are discussed in Section V and Section VI respectively. Finally, we conclude the paper in Section VII.

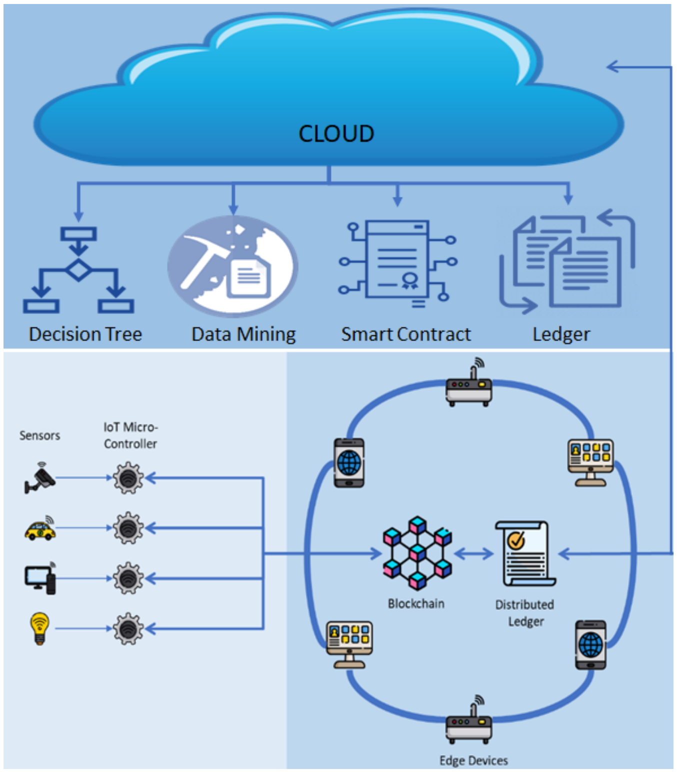


Fig. 1: End-to-end architecture of blockchain in an IoT workflow.

II. Background and Problem statement

A. Background

1) Collaborative Edge Computing : Collaborative computing is gaining a lot of interest in the computing domain. With the emerging Edge Computing, collaborative edge computing has become a key solution for the resource constraint environments, i.e., smart villages, agriculture, etc. [6]. There are a few related works as follows where authors have contributed to the collaborative computing in computing paradigm. A green and sustainable virtual network embedding framework for cooperative edge computing in wireless-optical broadband access networks [8]. Here authors have leveraged functionality to confirm backup EDCs and virtual network connectivity. In [9], Hou et al. proposed an online incentive-driven task allocation scheme to stimulate collaborative computing among EDCs and IoTDs. Here, EDCs manage the available resources dynamically combined with the neighbors. A collaborative edge data caching problem as a constrained optimization problem is solved in [10] to address the quality-of-service and data migration between the EDCs.

2) Lightweight Blockchain : Blockchain is no longer a new technology; instead, it is currently used in many applications domains for decentralized security. One of the primary challenges of Blockchain is to integrate with the application domain due to its properties. Where detailed rework on the blockchain consensus is preliminary. To address this flaw, Proof-of-Authentication (PoAh) is developed for the IoT and IoT-Edge workflow [11]. Where PoAh ignores the block validation process and introduces the authentication mechanism to make the overall process lightweight. It further extended to secure the devices in parallel to the data [12].

TABLE I: Notations Used in System Descriptions.

Acronym	**Description**
K_{PB}/K_{PR}	Public/Private Key of the SC
$E()/D()$	Encryption/Decryption
MAC	Message Authentication Code
$IoTD$	IoT Device
EDC	Edge datacenter
DT	Decision Tree
$MQTT$	Message Queue Telemetry Transport
Trx	Blockchain Transactions
Tr	Trust Value
Th	Threshold Value

B. Problem statement

From the above background discussion, it is clear that creating a collaborative environment in a resource constraint scenarios is vital in targeting the application domain, like smart villages, agriculture, etc. Again, filtering out the corrupted or modified data before processing or storing them at the edge is crucial. This article addresses the two critical issues in Edge computing scenarios, i.e., (i) filtering out the corrupted data before storing or processing at the edge and (ii) creating a collaborative edge computing environment for fast and scalable computing.

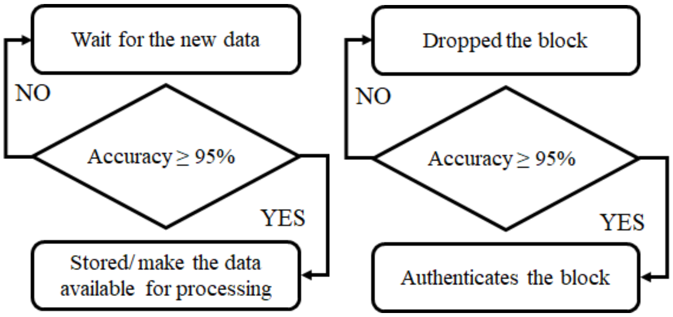
Further, only filtering out the corrupted data at the edge is not a scalable solution; our focus should be on securing the end-to-end IoT-Edge workflow. Single-point failure is a more significant challenge in current security solutions, and lightening the Blockchain is another issue while making the security solution decentralize. Targeting the PoAh, there is a scope to make the authentication process robust and application-specific. Our focus is to make the PoAh process robust by utilizing learning-based authentication, not just taking into account the cryptography.

III. Learning-based Data Filtering at Collaborative Edge Computing

In an IoT-Edge workflow, computational resources are either in the Cloud or Edge. Edge computing facilitates to processing of data or analysis in near real-time. An edge computing device can detect events in real-time by analyzing the data sent from the sensors and taking immediate actions compared to waiting for the Cloud to explore, causing a delayed response. However, the Cloud can study the long-term behavior of all the equipment in the manufacturing process by analyzing the information provided by EDCs to predict failures and provide recommendations to source queries.

The properties of the EDCs, such as limited computational power, limited storage capacity, limited energy supplies, etc., make it clear that an EDC cannot serve all the user/source requests without other EDCs. To resolve this issue, multiple EDC were deployed in an IoT workflow, as shown in Figure 1.

```
Algorithm 1: Data processing at the EDC
Data: Data blocks
Result: Accept or Reject the Data at the Edge
$\text{IoTD}\{\text{DATA}\} \rightarrow$ EDCs;
EDCs creates a collaborative environment;
Organize for applying DT (Ref: Figure 2a);
Apply DT;
Compare with training to find accuracy;
if $accuracy \geq 95\%$ then
    Stored/use for analysis;
else
    Drop the Data;
```



(a) Data filtering

(b) Block authentication

Fig. 2: Decision Tree Processes

Current research allows load balancing [13] [14] and service queue [15] to process service requests faster to build a scalable edge computing environment. The most traditional way is to put the data in a queue and process then either in a sequence or in priority [15]. Another way is the load balancing, where the requested EDC looks for other EDC's resources if it is overloaded [13] [14]. Source EDC communicates with other EDCs to find the best one by considering available resources and energy to process the incoming requests. However, collaborative edge computing brings the concept that all EDCs come together to serve the incoming requests.

There are high possibilities that data are modified or corrupted during the transit and before reaching the EDCS. It is an unavoidable step to detect the corrupted data and supply only original data for further processing and storage. This paper implemented the process and analysis to filter out the incoming corrupted data before making it available for data storage or analysis. We are applying the Decision Tree (DT) concept, as shown in Figure 2a. We are keeping the minimum accuracy percentage to 95% to detect the corrupted data. Detected corrupted data are dropped immediately and will not make available for further processing. The complete process of using DT in the data filtering is as shown in Algorithm 1.

Upon arriving at the data at the Edge, available EDCs come together to create a collaborative environment and apply DT (Algorithm 1) to filter out the corrupted data from the IoTD.

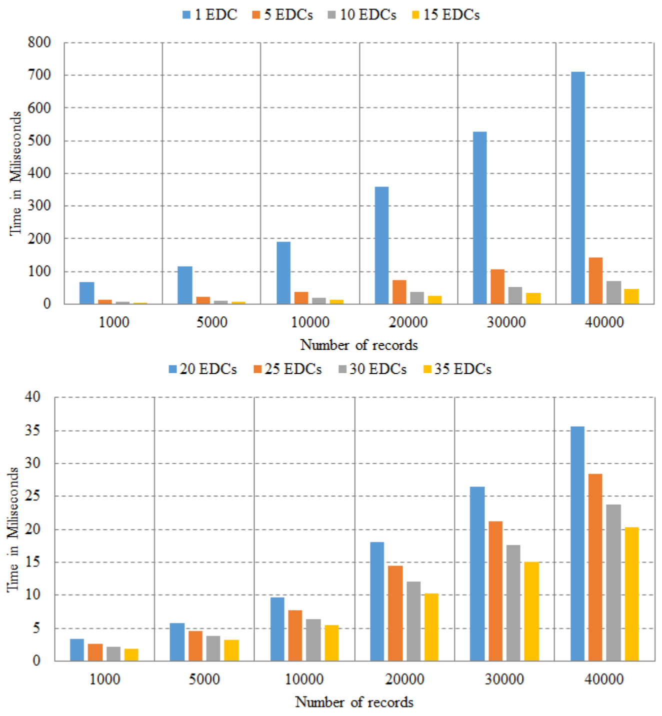


Fig. 3: Performance comparison edge computing in a collaborative environment

A. Implementation

For a real-time testbed implementation, we used a Raspberry Pi as the IoTD to gather the raw sensor data and many Raspberry Pis for creating a collaborative edge environment. This number of Raspberry Pis varies from 1 to 35 to complete the experiment and give comparative studies. The real-time data streamed from UO of Newcastle University acts as the sensor data for the experiment [16]. IoTD extracts this data from the repository and performs the operations. All the communication between the entities is through Message Queuing Telemetry Transport (MQTT). MQTT works with three mechanisms: published, subscribe, and broker [17]. It is a lightweight messaging protocol for small sensors and mobile devices having low power and memory, providing higher latency and reliability.

The performance of DT-based data filtration in the collaborative edge computing environment is shown in Figure 3. The experiment is started with one EDC to get the performance of a regular Edge computing environment. Subsequently, we increased the number of edges in the edge layer of the testbed, i.e., 5, 10, 15, 20, 25, 30, and 35 EDCs, to create collaborative edge computing scenarios. For all the experiments, the number of records as data streams ranges from 1,000 to 40,000 records. We have experimented with all possible combinations to get the performance and the performance of collaborative edge computing as shown in Figure 3. Individual experiments run three times and find the mean value to plot the results graph.

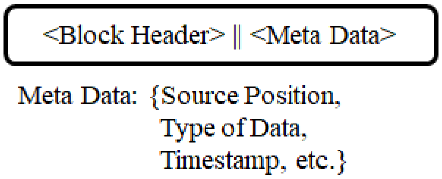


Fig. 4: Modified Blockchain Header.

IV. Decision Tree-based Secure and Robust Blockchain Consensus

Blockchain is the solution to make the end-to-end security model decentralized. Looking into the application domain, i.e., IoT-Edge workflow, we need a scalable solution to make the overall Blockchain process lightweight. Looking into the process of blockchain consensus, usually, it takes 10 minutes to validate a block and two houses of energy used in one-day [18], which is impossible for any IoT application domain. To address this PoAh mechanism is developed with a lightweight process, where PoAh introduces the authentication instead of block validation [11]. So, PoAh is best for distributed systems with resource constraint devices and works perfectly for any network.

At the beginning of the process, IoTD sense the data, and data are formatted to transactions (Trx). The transactions are formatted as data, source, and destination address. Further, the transitions are assembled to form the Block, $i.e., Block = \{Trx1, Trx2, \ldots, Trxn\}$. The structure of the block header is as in Figure 4.

After the creation of blocks, blocks are broadcasted to the trusted networks. All the broadcasted blocks are encrypted and signed using the source private key K_{PR}, i.e., x, and make the public key K_{PB}, i.e., y, available to everyone. Where the set of trusted nodes creates a mesh network, and this network is reachable from any part of the network. The ElGamal cryptosystem is used for encryption and decryption in this proposed model. Where $y = g^x$ *(mod p)* is the process, x is the private key K_{PR}, and y is the public key K_{PB}. Here, the large prime number p and the generator g is publicly known to all the devices in the network.

The trusted network contains a set of trusted nodes; a node's trust value (Tr) is initialized at the deployment phase. With false block validation, the one trusted may lose their trust values and become a normal sensing device of the network. In contrast, a normal node can be part of the authentication validation phase to gain the trust value. In both scenarios, a node is eligible to become part of the trusted network/authentication process if the trusted value is greater than equal to the threshold $(i.e., Tr \geq Th)$, as shown in Figure 5. For an example and the values in the implementation is initiated as, initial trust value to 10 $(i.e.Tr = 10)$, normal node trust value to 0 $(i.e.Tr = 0)$ and the threshold value to 5 $(i.e.Th = 5)$. As block authentication continues, the trust value got updated and always compared with the threshold before becoming part of the authentication. The process is as shown in Figure 5.

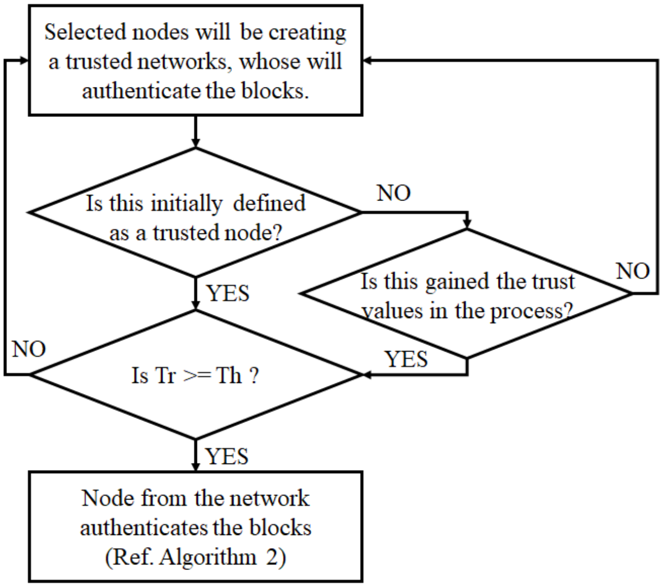

Selected nodes will be creating a trusted networks, whose will authenticate the blocks.
Is this initially defined as a trusted node?
NO
YES
Is this gained the trust values in the process?
NO
YES
NO
Is Tr >= Th ?
YES
Node from the network authenticates the blocks (Ref. Algorithm 2)

Fig. 5: Trusted node processes for the PoAh.

In the authentication process, the trusted node receives the blocks for the authentications and extracts the source's public key, i.e., y, for signature verification. According to the concept of the asymmetric cryptosystem, one cannot compute the private key (i.e., x) with the help of the public key i.e., y. After successful verification, the trusted node verifies the source MAC as the second verification level. However, the most crucial step is to apply the DT only after these two initial evaluation steps. The DT will be applied to the received Block, and the process is as in Figure 2b. The minimum accuracy implemented for this model is 95% to verify that the Block is authenticated. An accuracy of less than 95% will drop the Block and wait for other Blocks to come in for the authentication. Subsequently, all the authenticated blocks will be verified before broadcast to add to the chain. The end-to-end process of the consensus mechanism is as shown in the Algorithm 2.

V. Theoretical Validation

After integrating the DT in $PoAh$, the consensus process will be robust, reducing the probability of false authentication. However, it is nearly impossible for an attacker to change the context and data of a block to authenticate a malicious block in the process of DT. An attacker may fool the process of digital signature and MAC verification. Further, the authentication process is not just limited to number theory grames but also secures the system with supervised learning. In our approach, DT combined to evaluate the Source ID, Source position, Type of Data, Timestamp etc., to authenticate the Block. This approach sets the bar higher for the attackers to overcome the authentication process.

Along with the above validation, there are more claims proved in the PoAh model description [7], such as (I) PoAh

```
Algorithm 2: DT enabled PoAh
Data: All nodes generate and collectes data to create
      blocks.
Result: Authenticate or drop the blocks
$(Trx^{+}) \rightarrow Block$ ;
$Block\ Header \parallel Meta\ Data$ (Ref. Figure 4) ;
Sign $\Rightarrow K_{PR}(Block) \rightarrow$ Broadcast ;
if $Verified$ then
    $Verify \Rightarrow MACChecking$ ;
    Verify Sign $\Rightarrow K_{PB}(Block)$ ;
    Apply DT for authentication (Ref. Figure 2b) ;
    if $Authenticated$ then
        $Block \| PoAh(Block)$;
        Ref. Algorithm 1 ;
        Add Block to the DLTs ;
    else
        Do not add the Block to the DLTs;
else
    Drop the block;
```

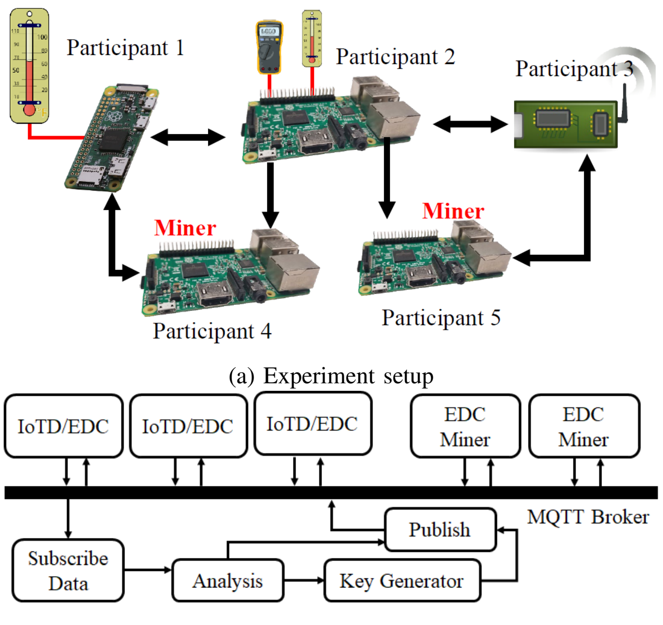
utilizes minimal resources for block validation, (ii) PoAh requires minimal time compared to PoW without compromising security threats, (iii) PoAh provides substantial security while integrating a blockchain based decentralized security solution to the IoT, and (iv) PoAh provides a better platform for IoT compared to other blockchain consensus algorithms.

VI. Experiment and Evaluations

This section detailed the experient setup and performance analysis of decision tree-based secure and scalable blockchain consensus for IoT-Edge workflows.

A. Experiment Setups

The experiment is set up in a lab testbed as shown in Figure 6a. We have used five Raspberry Pis in total, where three Pis are generating blocks, and two are for the block authentication process(mining process). Further, they are creating a mesh network for communications. The starting point of the process is from IoTD, which sends the data through MQTT as shown in Figure 6b. As mentioned earlier, MQTT is a lightweight messaging protocol and works with three mechanisms; they are publish, subscribe and broker [17] [19]. There can be multiple $IoTD$ up to $EDC(Trusted)$. They are the mining nodes that collect or generate data from various sources. The process uses an asymmetric key and hence a limited block of data sent at a given time. They send the encrypted data and signature as well. Some specific EDC acting as verifier $EDC(Trusted)$ receives the encrypted data and signature from $IoTD$. Upon receiving the data, $EDC(Trusted)$ decrypts it and uses the signature to verify the data received from the $IoTD$. If $EDC(Trusted)$ verifies the signature, it encrypts the data and sends it to other $EDC(Trusted)$. Every entity has its public/private key pairs. Further, the proposed DT will apply to perform the final stage of authentication as shown in Figure 2b and Algorithm 2. The Block will be accepted if the accuracy is greater than equal to 95%. Upon receiving data from specific EDC, the EDC decrypts it, creates a hash $(SHA-256)$ value for the data and saves it to a file. $EDC(Trusted)$ could communicate the data to other $EDCs$.



(a) Experiment setup

(b) Real-time system implementation architecture.

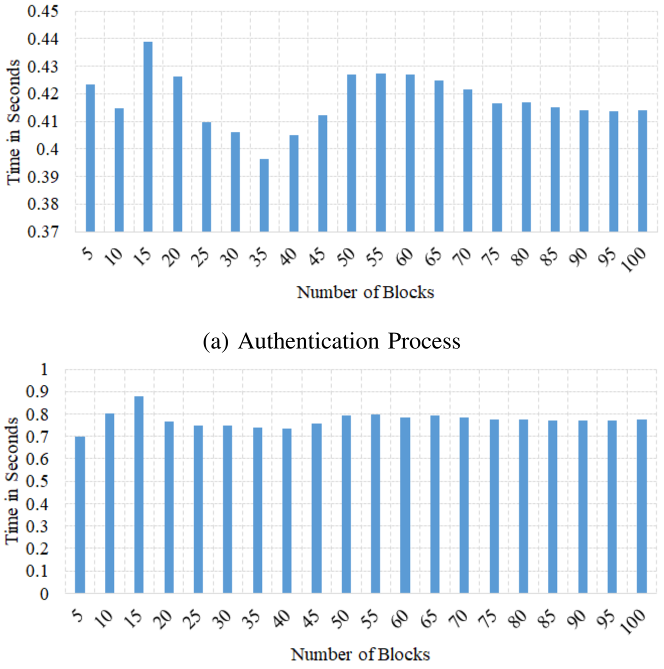
Fig. 6: Experiment setup and implementation architecture of DT-based PoAh

B. Result Analysis

This section deals with the time durations required for various processes. Here the benchmarking is done at two levels: First is the time duration between encrypting the message and creating the signature, decrypting and verifying the signature, and processing DT; second is the total duration to complete one cycle of sending data, encryption, and decryption with asymmetric key, creating hash and save them to files. The time duration here is the average time required to process data with the intervals of 5 blocks starting from 5 up to 100.

1) Authentication Process: It starts from the time when IoTD encrypts the data using K_{PB}. The IoTD then creates a digital signature using its own K_{PR}, and the data itself. The encrypted message and the signature are then sent to the Miner (Trusted Node). Using K_{PR}, Miner decrypts the data from IoTD, and verifies the signature using, the decrypted data, signature received, and K_{PB}. If the signature verification is valid, Miner applies the DT for the final authentication step and finally communicates the data to other entities. The time duration varies between 0.396 to 0.438 seconds. The experiment result is as shown in Figure 7a.

2) Overall Process: Here, we evaluate the proposed approach's end-to-end process, including the authentications phase. There are multiple encryption/decryption processes using asymmetric keys, creating and verifying digital signatures,



(a) Authentication Process

(b) Overall Process

Fig. 7: Performance of DT-based PoAh

communicating data to one or more EDC, creating the hash values of the data, applying DT at the Miner, and saving data into files. The experimenting model considers 5 EDCs. Hence the readings will vary by the increment or decrement of the EDCs participating in the processes. The overall time starts from data encryption in IoTD to storing data in the last EDC. The overall time ranges between 0.6 to 0.803. There is a consistent performance in the overall process, and it is pretty evident as the number of data blocks increases. The experiment result is as shown in Figure 7b.

VII. Conclusion

This paper proposed a solution to solve two problems in IoT-Edge workflows, i.e., (i) filtering out corrupted data in a collaborative edge computing environment; and (ii) robust blockchain consensus designing for the secure end-to-end system. We have applied a decision tree-based solution to filter out the corrupted data at the Edge layer and create a robust blockchain consensus (Proof-of-Authentication (PoAH)). Both these contributions are essential for an end-to-end system. Finally, the real-time testbed results validate the proposed approaches for the IoT-Edge workflows.

Acknowledgment

The statements made herein are solely the responsibility of the authors. This publication is based upon work supported by the Khalifa University under Award No. FSU-2022-018.

References

[1] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, 2017.

[2] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE access*, vol. 6, pp. 47 980–48 009, 2018.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[4] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.

[5] Y. Wu, "Cloud-edge orchestration for the internet of things: Architecture and ai-powered data processing," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 792–12 805, 2020.

[6] D. Puthal, S. P. Mohanty, S. Wilson, and U. Choppali, "Collaborative edge computing for smart villages [energy and security]," *IEEE Consumer Electronics Magazine*, vol. 10, no. 3, pp. 68–71, 2021.

[7] D. Puthal, S. P. Mohanty, V. P. Yanambaka, and E. Kougianos, "Poah: A novel consensus algorithm for fast scalable private blockchain for large-scale iot frameworks," *arXiv preprint arXiv:2001.07297*, 2020.

[8] Z. Ning, X. Kong, F. Xia, W. Hou, and X. Wang, "Green and sustainable cloud of things: Enabling collaborative edge computing," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 72–78, 2018.

[9] W. Hou, H. Wen, N. Zhang, J. Wu, W. Lei, and R. Zhao, "Incentive-driven task allocation for collaborative edge computing in industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 706–718, 2021.

[10] X. Xia, F. Chen, Q. He, J. Grundy, M. Abdelrazek, and H. Jin, "Online collaborative data caching in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 2, pp. 281–294, 2020.

[11] D. Puthal, S. P. Mohanty, P. Nanda, E. Kougianos, and G. Das, "Proof-of-authentication for scalable blockchain in resource-constrained distributed systems," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2019, pp. 1–5.

[12] S. P. Mohanty, V. P. Yanambaka, E. Kougianos, and D. Puthal, "Pufchain: A hardware-assisted blockchain for sustainable simultaneous device and data security in the internet of everything (ioe)," *IEEE Consumer Electronics Magazine*, vol. 9, no. 2, pp. 8–16, 2020.

[13] Q.-M. Nguyen, L.-A. Phan, and T. Kim, "Load-balancing of kubernetes-based edge computing infrastructure using resource adaptive proxy," *Sensors*, vol. 22, no. 8, p. 2869, 2022.

[14] D. Puthal, R. Ranjan, A. Nanda, P. Nanda, P. P. Jayaraman, and A. Y. Zomaya, "Secure authentication and load balancing of distributed edge datacenters," *Journal of Parallel and Distributed Computing*, vol. 124, pp. 60–69, 2019.

[15] M. Guo, L. Li, and Q. Guan, "Energy-efficient and delay-guaranteed workload allocation in iot-edge-cloud computing systems," *IEEE Access*, vol. 7, pp. 78 685–78 697, 2019.

[16] L. Smith and M. Turner, "Building the urban observatory: Engineering the largest set of publicly available real-time environmental urban data in the uk." in *Geophysical Research Abstracts*, vol. 21, 2019.

[17] N. Naik, "Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http," in *2017 IEEE international systems engineering symposium (ISSE)*. IEEE, 2017, pp. 1–7.

[18] 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)*. IEEE, 2017, pp. 618–623.

[19] D. Puthal, S. Wilson, A. Nanda, M. Liu, S. Swain, B. P. Sahoo, K. Yelamarthi, P. Pillai, H. El-Sayed, and M. Prasad, "Decision tree based user-centric security solution for critical iot infrastructure," *Computers & Electrical Engineering*, vol. 99, p. 107754, 2022.