Improving IoT Security Using Blockchain

Khalid AlJemy
College of Engineering and Architecture
AlYamamah University
Riyadh Saudi Arabia
khalidaljemy@gmail.com

Mohammed AlAnazi
College of Engineering and Architecture
AlYamamah University
Riyadh Saudi Arabia
mohammedfarhan2030@gmail.com

Mohammed AlSofiry
College of Engineering and Architecture
AlYamamah University
Riyadh Saudi Arabia
m_alsofiry@hotmail.com

Adeel Baig
College of Engineering and Architecture
AlYamamah University
Riyadh, Saudi Arabia
a_baig@yu.edu.sa

2019 IEEE 10th GCC Conference & Exhibition (GCC) 978-1-5386-9477-0/20/$31.00 ©2020 IEEE 10.1109/GCC45510.2019.1570521015

***Abstract*—Internet of Things (IoT) is the new technology that will change the Internet era in the coming years. IoT can be used in many fields, such as digitally connected factories, facility management, production flow management inventory management, safety , security, health care, logistics, supply chain optimization, and so on. However, due to limited capabilities in IoT, it is impractical to implement extensive security solutions. Also, security standards for IoT are not mature enough. As a result, many security incidents affect IoT reliability. In this paper, we propose an effective security mechanisms that can be implemented to ensure data accuracy and integrity of IoT devices along with considering its limited hardware resources. We implement Blockchain in IoT to enhance IoT security. We apply our approach using the Ethereum protocol by having multiple Raspberry Pi devices connected as peers to the Blockchain. Also, we discuss the implementation and the expected outcome of this project, which is to implement the Access Control List mechanism using Blockchain to improve the overall IoT security and to connect IoT devices to Ethereum as light nodes.**

Keywords—IoT, Blockchain, ACL, PoA, Ethereum, Smart Contracts.

I. INTRODUCTION

The Internet of Things (IoT) stimulates a worldwide network of nodes-based communication without any human interaction to collect and analyze data. It is likely the beginning of the internet by enabling humans to communicate with each other and make them share or reuse data. Therefore, everyone has noticed that devices have become smaller and smarter as well as parts of our lives, such as the case with mobile smartphones, smart TVs, and smart cars. Cisco has mentioned that by 2020 smart devices will reach 50 billion all over the world where IoT becomes a means of communication in the society over the coming years [1]. The main components of IoT are security, middleware, and architecture [2]. Confidentiality, integrity, and availability are the major security concerns that IoT raise [2]. When the IoT concept was proposed in the late 1990s, security specialists have warned of the potential danger of a huge number of unsecured devices connecting to the Internet since then. [2]. According to Proof Point, "more than 25% of the botnet was made up of devices other than computers, including smart TVs, baby monitors and other household appliances [2]. There are many cryptography algorithms and cyphering techniques that can be used to maintain IoT information security. One of the new technologies that use these algorithms efficiently to due diligence IoT needs is the distributed ledger "Blockchain" concerning IoT limitations in processing power. The objective of this paper is to describe how IoT security can be enhanced by integrating it with a Blockchain mechanism using Ethereum. The main benefit of Ethereum is that it allows nodes in its network to publish smart contracts that include the rules or logic applied that need to be adhered to. Smart contracts will be used to implement our solution, which is an access control list for participants of an IoT network. The remainder of the paper is organized as follows: In section II, we discuss related work and the background of this paper, and in section III, we present our methodology. Section IV discusses the overall system architecture; in section V, we discuss implementation details; section VI is dedicated to a comparison, and in section VII, we provide a conclusion.

II. BACKGROUND AND RELATED WORK

A. Problem Statement

IoT lacks security efficiency, and in order to improve IoT security; we need to overcome hardware limitations such as heterogeneous capabilities in processing resources. Many IoT devices have no security protection software provided by anti-virus vendors, which is not applicable to IoT devices, considering their limitations. Also, scalability issues occur when a massive number of nodes are involved in the IoT network; therefore, network security where IoT devices are functioning needs to be addressed and improved.

B. Background

Blockchain is recently used in a wide variety of scenarios, such as financial systems, IoT, voting, storage and cloud, so on and so forth. We propose a Blockchain based solution to improve IoT security. We use Ethereum Blockchain that is based on Proof of Work consensus engine where miners race to solve a mathematical equation to mine a transaction. Ethereum Harmony platform is the graphical user interface that is based on Ethereum protocol. In this platform, nodes can start mining manually by activating the mining feature from Ethereum Harmony's terminal. In this scenario, miner activation will be initiated by a powerful node using an Ethereum full node protocol that works as a miner in the network. The processing power of IoT nodes would be reduced and security precautions are optimized when IoT nodes run the Ethereum light-node protocol that eliminates transactions mining and allows IoT devices to take benefit of the full-node. We use Proof of Authority protocol; by eliminating proof of work and replacing it with proof of authority where list validators create blocks and transactions in the Blockchain network; thus, miners are replaced with singers (authorities). Our paper provides different implementations in different consensus engines. We believe that our approach improves IoT security, considering the best practices applied.

C. Related Work

In this subsection we briefly present the existing work in this domain.

Authors in [3] present a framework of an authorization manager that makes users to control their data through "Fair Access" [3]. Fair Access adapts Blockchain into a decentralized access control manager, and it provides several transaction types, such as, delegating and revoking access [3]. In our opinion, Fair Access could use the Ethereum smart contract transcode for a more complex and granular access control model.

Authors in [4] present a solution to solve the high computational powers problem for IoT devices to communicate with Blockchain. The solution is based on centralized and decentralized architecture. There are centralized servers that act as intermediates with IoT devices and Blockchain [4] This solution uses smart contracts and it is based on Embark Blockchain. The updated ledger is saved in IPFS which can be a limitation for IoT devices. Thus, IPFS cannot talk to anything but browser nodes.

Authors in [5] present hashing data records collected from the drone in an IoT cloud-based application that registers drones to the Blockchain for solving crucial security issues and data resilience [5]. The solution this paper presents is to store data on the cloud and generate a receipt for each storage process and to the minimize power consumption of drones [5]. By moving Blockchain's ledger from the device to the cloud can cause undetected cyber-vandalism; thus, information can be changed before it reaches the cloud.

CapChain is an access control framework [6]". CapChain does not use smart contracts for access control. The abilities of CapChain treats access rights as a type of assets that could be transferred through transactions between users.

Another solution is discussed in [7], where, a light-weight access control solution, for IoT devices based on smart contracts BlendCAC [7]. BlendCAC eliminates the supervising authority for IoT devices and enables them to control their resources [7]. This solution is implemented on Raspberry Pi devices on a private Blockchain network as a proof of concept.

The "Flow Chain" distributed ledger system uses Proof of Stake (PoS) as its consensus engine [8]. The Flow Chain's data structure provides an efficient mining feature that restrict every node to mine blocks at its own branch [8].

The solutions presented above addresses IoT issues and solutions of security. In this work, we plan to explore the possibilities and propose effective solution for IoT security.

III. METHODOLOGY

In this section, we present our methodology.

The system provides a secure environment for data exchange among participants and secure data storage. The hardware at system end nodes will be different from one another. Some nodes will lack the processing power, and battery life will be limited, so the mining process will be eliminated from low-end nodes. A monitoring process should be there to ensure environment privacy. When the access control mechanism is implemented in Blockchain, it will minimize the chances of intrusion attacks to the system. Also, Blockchain will provide a ledger of activities to ensure non-repudiation. We integrate IoT with Ethereum Blockchain to implement ACL using Blockchain Ethereum smart contracts, where this smart contract contains nodes' information and assign manual ACL number at first then automate ACL number to maintain Blockchain privacy. We run the system using Ethereum protocol based on PoW in the Ethereum Harmony platform with a single mining node or multiple if needed. Nodes can join the network as peers, but miners must start mining functionality only manually by the admin inside the platform's terminal. Other nodes will maintain a distributed ledger of network transactions.

When we use PoA consensus, different computationally powerful and pre-defined authorized nodes will generate the transaction of other nodes and the ACL mechanism. ACL mechanism will minimize the possibility of acquiring unauthorized access to the Blockchain. By having a topology with IoT devices and one powerful single node or multiple nodes for validating blocks, we believe that it is possible to implement Ethereum protocol in all nodes and run Ethereum Harmony on the network administrator. The implementation of the access control list mechanism in Ethereum Harmony platform through initiating an ACL smart contract. This contract will govern the topology of the network. In our case, we publish an access control list mechanism smart contract through Ethereum Harmony running a Proof of Work consensus with a full-node for miners and a light-node for IoT devices.

A light-node is a piece of software that interacts with full-nodes to collaborate with Blockchain. Ethereum light-nodes enable devices to save power, communicate with full-nodes, and eliminate reading and writing the Blockchain [9]. A light-node relies on full nodes for many operations, from requesting the latest headers to asking for the balance of an account; therefore, full-nodes are intermediates whereby light-nodes communicate with the Blockchain [9].

When we use PoA consensus, where only some authorized nodes will explicitly create blocks instead of solving

mathematical equations (PoW) [10]. In fact, using IoT with a PoA consensus eliminates complex mathematical problems by using authorized nodes. These are the only nodes allowed to initiate new blocks, so this will enhance Blockchain security. Blockchain's permanent record will be by the authorized nodes [10].

IV. Overall System Architecture

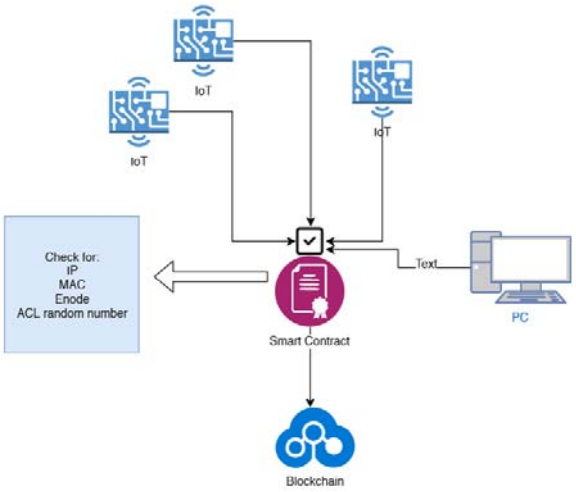

Fig. 1. Using Ethereum Smart Contract for access control list

As shown in Figure (1), before a node generates a block, it will check the information in the ACL smart contract, such as IP, MAC address, Enode, and the ACL random number (will be discussed in Algorithmic Component subsection). The ACL contract will be distributed to all nodes in the Blockchain network. Each time a new node is authorized to enter the network, the ACL smart contract will be updated and distributed to all nodes. The consensus engine in this scenario is PoW. Because this is a proof of concept and we test it for small scale networks. Since transactions mining starts manually in Ethereum's full node; IoT devices will run on Ethereum light-node; and as a result, the mining process will be eliminated from IoT devices. The overall system contains ACL on Blockchain that enables us to monitor nodes and governs the environment. Blockchain hashes transaction, time stamp them, then add a reference to them in the previous block, and therefore integrity is ensured.

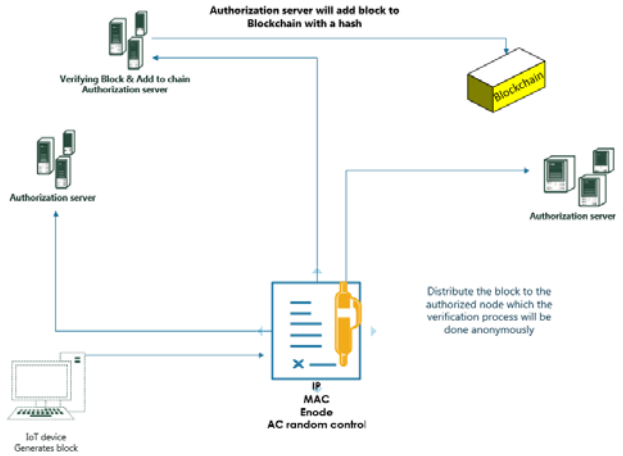

Fig. 2. Using PoA for generating the access control list

As shown in figure (2) PoW is replaced with PoA (Proof of Authority), where specifically authorized nodes will be chosen to generate new blocks; therefore, the block header extension in each generated block will contain information of authorized nodes in the network. Authorized nodes are responsible for adding new nodes to the network and updating the ACL smart contract.

V. Implementation Details

In this section, we present our implementation details. Due to space limitation, we present the details briefly.

A. Hardware Implementation

The hardware is a high processing PC, and nodes can be PCs or/and low processing devices, such as Raspberry Pi connected via LAN. We selected the Raspberry Pi in our project precisely because it can act as a resource-constrained device. Moreover, it is a flexible device that runs a partial OS, and it will contribute to our project the concept of secure limited resource IoT. Raspberry Pi features provide us with a small portable computer that can perform multiple tasks for different purposes, such as encrypting the data for storage or acting as a wireless security camera or a sensor followed to connect Raspberry Pi to the Blockchain:

- We configure the Ethereum protocol inside the device to initialize the unique Encode.
- Then we add the device to the Blockchain's network by the "Encode" followed by the device IP.
- The miner will add the device inside the platform terminal as a peer. If the adding peer process succeeds; a "*true*" condition will be presented inside the platform terminal.
- We configure Geth for Raspberry Pi as an Ethereum light-node.

B. Implementation using Proof of Work

1. Software Implementation

We use Ethereum Harmony which is based on Ethereum Protocol. This software allows us to control the environment of the network where we can add smart contracts, start the mining process, distribute the contracts among participants, and add nodes to the Blockchain network by Ethereum Harmony.

2. User Interfaces In this subsection, we will present the user interface of the Ethereum Harmony platform and explain the options that the platform offers with details.

We discuss in figure (3) Ethereum Harmony GUI as follows:

1. Home: This option previews an overall look of the network features whether if it is private, public, or customized. It also previews information about the genesis block, the exact date and time of joining the Blockchain, the type of synchronization being used, and some general nodes' information.

2. Ethereum peers: This option enables us to the nodes' node ID, IP, geographical location of the Blockchain network nodes, ping latency and reputation, and offline nodes.

3. System Logs: This option is the ledger of the Blockchain, where most of the activities of the Blockchain

network previewed here, such as transactions, synchronizations, pings, peers' status, etc. Each activity is hashed, recorded and saved.

4. JSON RPC Usage: This option provides a list of commands and guidelines that used in the operating system's terminal, such as adding peers, preview the node information, preview the number of peers, start and stop mining, etc.

5. Terminal: This option is very similar to the JSON RPC Usage option, although all these commands can be entered and used inside the local host terminal of the platform.

6. Wallet: The wallet contains nodes' and contracts' addresses. This option is essential for contract publication.

7. Contracts: After verifying contracts in the platform's terminal; this option allows us to publish contracts in the Blockchain network to the registered nodes in the wallet.

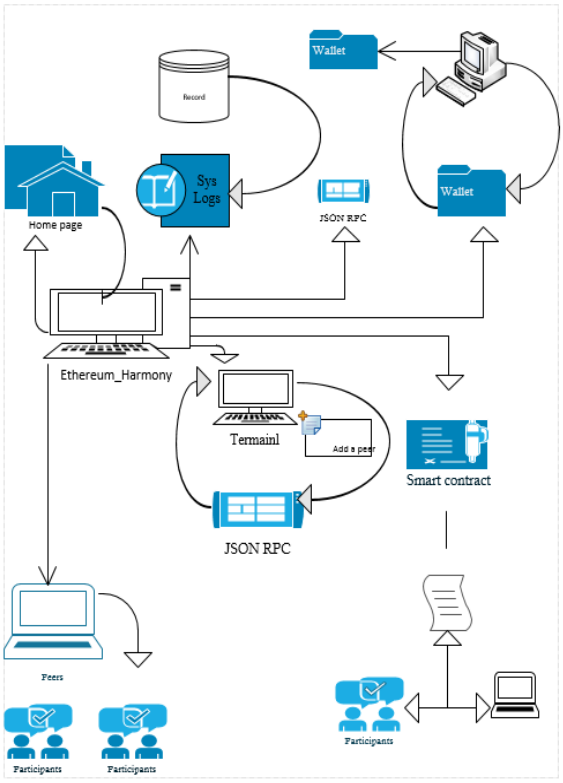

Fig. 3. Ethereum Harmony GUI

3. Algorithmic Components

The access control algorithm exists inside the Blockchain, and it will function as a smart contract.

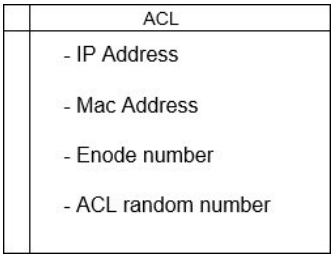

Fig. 4. ACL smart contract

It will store the mac address, IP, Encode of the device, and the access control number. The novelty of the access control number that we are presenting is that the admin should generate it for each node and distribute it, then it will be changed frequently after a while. This number can be considered as a certificate, and to update it; the old number must be provided to proceed for the new access control number. This ACL mechanism will ensure security and privacy of the network and nodes in a way that enables admin to authorize one who can participate, alongside the Blockchain time stamps, peer list, and distributed ledger. Although our approach uses PoW, we ran Ethereum light-node on IoT devices to eliminate the concept of racing to validate block; thus light-nodes will contain time stamp updated ledger. Ethereum full-nodes will race to mine the generated blocks.

C. Implementation using Proof of Authority

1. Software Implementation

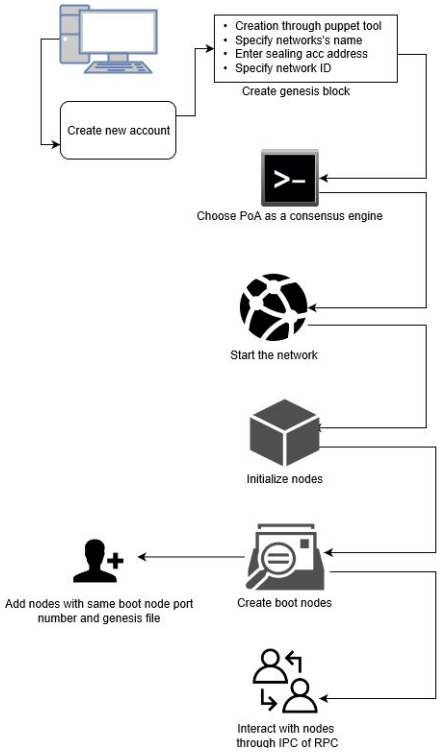

Fig. 5. Implementation using PoA consensus

In this subsection, we address implementation of Geth using PoA consensus. In figure (5) we discuss software implementation using PoA consensus. The software we use is a platform that is functioning on Ethereum protocol. We follow these steps in the implementation process:

1. **Create a new account:** We first create two accounts in Geth, one for the sealing node and the other for deploying the smart contracts.
2. **Create the genesis block:** We create a genesis block based on Puppeth tool provided by go-Ethereum. In genesis block creation we specify the name of the network and enter the sealing account address or any pre-funded account. Finally, we define the network ID, and we save the file.
3. **Selecting the consensus**: During the command-line interaction with Geth, it asks "which consensus engine to be used?"; we choose "Clique - proof-of-authority".
4. **Start the network**: We initialize the new chain from the genesis block and run it.

5. **Initializing nodes**: Each node must be initialized with the same genesis file.
6. **Create a boot node**: The boot node purpose is to help nodes to discover each other.
7. **Add nodes:** When adding nodes, they must use the same port of the boot node; the genesis file must exist in the node we want to add.
8. **Nodes Interaction**: Attach Geth Java Script Console to one of the nodes and interact with nodes locally through IPC (Inter-Process Communication), or RPC (Remote Procedure Call) over the Internet through HTTP.

2. User Interfaces

For the graphical user interface, we use the Mist browser for deploying and interacting with smart contracts and account management; although we can connect to Mist locally through IPC or RPC through the Internet HTTP.

3. Algorithmic Components

There is a new concept in Blockchain called Proof of Authority (PoA), where only authorized nodes can create new blocks. These authorized nodes are pre-approved "Sealers" [11]. PoA doesn't change the core of data structure and block headers, and when validators create blocks, they will be stored in the 32 bytes extra-data in block headers. All the nodes within the network will be sharing the same access control list smart contracts; therefore, PoA nodes will maintain and control the nodes in the access control list smart contracts. PoA nodes generate the manual random number for the new node addition process. So, when nodes generate new blocks, it will be verified first to the ACL smart contract. PoA nodes distribute the updated ledger to all nodes. This process can help overcome the overhead that is caused by PoW. Proof-of-Authority is not at all suited for public networks where the trust should be as distributed as possible, but on the other hand, it is almost a perfect fit for private networks [12].

VI. COMPARISON

In this section we present different comparisons for Blockchain protocols, Blockchain platforms, and our approach. Also, we highlight benefits of using Ethereum Harmony.

TABLE I.
COMPARISON BETWEEN BLOCKCHAIN PROTOCOLS

Evaluation Criteria	Proof of Work	Proof of Stake	Proof of Authority
Mining Blocks	Everyone mine	Only select few "Validators"	Selected Authorized Validators
Data Position	Decentralized	Decentralized	Decentralized
IoT processing power	Expensive	Less expensive	Practically not expensive

Table I shows a comparison between different Blockchain mining protocols. Data is decentralized in all of the consensus engines. In PoW, mining blocks done by any node that participates in the Blockchain network, and it could be expensive for IoT in processing power. There are only a few validators are selected for mining purpose in PoS consensus, and it is less expensive for IoT than PoW. Mining blocks in PoA done by authorized selected validators and it is not expansive in processing power for IoT.

TABLE II. OUR APPROACH

Evaluation criteria	Ethereum with ACL	Ethereum without ACL
Access Management	YES	NO
Application Security	YES	YES
Blockchain Privacy	YES	NO
Data Security	YES	YES
Encryption	YES	YES
Governance	YES	NO

Table II presents an evaluation of an environment where IoT devices are integrated with a Blockchain network with or without ACL where Ethereum protocol runs on all nodes. As a result, by running ACL with Ethereum, we improved Blockchain's access management, privacy, and governance.

TABLE III.
COMPARING DIFFERENT BLOCKCHAIN PLATFORM FUNCTIONALITIES

Evaluation Criteria	FlowChain	Ethereumj	Kovan
Proof protocol	PoS	PoW	PoA
Miner Selection	By Admin	By puzzle/Admin	Selected Validators
51% attack immunity	Not possible	Depends on the Blockchain visibility	Not possible
Hash Algorithm	Double SHA-256	Double SHA-256	SHA-3
Support Smart Contracts	No	Yes	Yes
Transaction Mining	Everyone	Everyone/Nodes that starts mining	Selected Validators

In table III we present different platforms that use proof protocols and discuss each of them briefly.
Flow Chain uses Proof of Stake where the admin directs the miner selection. It is not possible to attack 51% of the Flowchain network. The algorithm Flow Chain uses is a

double hash algorithm which is SHA-256. Smart contracts are not supported in Flowchain; also, everyone in the Flowchain network can mine transaction.

Ethereum uses Proof of work where the miner selection is based on solving a puzzle, or the admin can select it. For the attack immunity, it depends on the blockchain visibility. The hash algorithm in Ethereum is double SHA-256. Ethereum supports smart contracts, and everyone within its network can mine transaction or the nodes that start mining.

Kovan platform uses Proof of authority where there is no miner. On the other hand, there will be validators selected by a higher authority. It is not possible to attack 51% of the Kovan network. The algorithm that Kovan uses is a double hash algorithm which is SHA-3. Kovan supports smart contracts; therefore, no one in the Kovan network can mine transaction, except the authorized validators "sealers".

We propose an effective security mechanisms for IoT devices using Ethereum Harmony platform based on the Ethereum core. Ethereum uses PoW as its consensus engine. We have presented some concepts of using PoA consensus as an effective solution. Therefore, implementing access control list mechanism in Blockchain with different choices either using PoA or PoW improves IoT security and eliminates mining blocks from IoT nodes. We use Ethereum Harmony for many reasons, such as:

- *Ethereum protocol:* Ethereum protocol is suitable for our environment as its benefits are mentioned in the sections above, we have tested Ethereum Harmony platform as an open source testing product that runs Ethereum protocol and functionalities as it should be. Also, Ethereum core allows us to use different consensus engines.
- *Smart Contracts:* Since one of the main benefits of Ethereum protocol is smart contracts, we have tested many open source contracts, some of them include an access control mechanisms, and we published them to nodes in the Blockchain network.
- *Peers in the Blockchain network:* Ethereum Harmony platform gives us the ability to add peers, check their activity in the log option as an immutable ledger. It provides the peers the ability to act as a miner, and it holds a wallet of nodes' addresses that will be used to publish the contract for these addresses in the wallet.
- *Access control list mechanism:* We implement the access control list mechanism using Ethereum smart contracts.

VII. CONCLUSION

In conclusion, the Internet of Things stimulates a world of nodes communication without any human interaction to collect and analyze data. It is likely the beginning of the internet by enabling humans to communicate with each other and making them share or reuse data. Therefore, everyone has noticed that devices have become smaller and smarter as well as parts of our lives, such as the case with mobile smartphones, smart TVs, and smart cars. Security is one of the three elements of IoT components, along with architecture and middleware. The objective of this paper is to enhance security in IoT by integrating it with Blockchain mechanism using Ethereum. Smart contracts enhance the implementation of access control list for the IoT network participants. Our paper provides different implementations in different consensus engines. We believe that our enhancement improves IoT security considering the best practices applied.

REFERENCES

[1] I. Yaqoob, E. Ahmed, I. Hashem, A. Ahmed, A. Gani, M. Imran, et al., "Internet of Things Architecture: Recent Advances, Taxonomy, Requirements, and Open Challenges," IEEE Wireless Communications, June 2017.

[2] C. Stergiou, K. Psannis, B. Kim, B. Gubta, "Future Generation Computer Systems," Secure integration of IoT and Cloud Computing, November 2016.

[3] A. Ouaddah, A. Elkalam, A. Ouhman, "FairAccess: a new Blockchain-based access control framework for the Internet of Things," Security and Communication Networks, February 2017.

[4] N. Rifi, E. Rachkidi, N. Agoulmine, N. Taher, "Towards Using Blockchain Technology for IoT data access protection", COSMO, IBISC Laboratory, University of Evry, France, January 2018.

[5] X. Liang, J. Zhao, S. Shetty, D. Li. "Towards data assurance and resilience in IoT using blockchain," IEEE 17th International Conference on Ubiquitous Wireless Broadband (ICUWB), December 2017.

[6] T. Le, M. Mutka, "CapChain: A Privacy Preserving Access Control Framework Based on Blockchain for Pervasive Environments," IEEE International Conference on Smart Computing (SMARTCOMP), July 2018

[7] R. Xu, Y. Chen, E. Blasch, G. Chen, "BlendCAC: A BLockchain-ENabled Decentralized Capability-based Access Control for IoTs," IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (Smart Data), April 2018

[8] J. Chen, "Flowchain: A Distributed Ledger Designed for Peer-to-Peer IoT Networks and Real-time Data Transactions,"IEEE Globecom Workshops (GC Wkshps), January 2017.

[9] Ethereum Light-node, Retrieved from: https://www.parity.io/what-is-a-light-client/

[10] Proof-of-Authority Chains, "Ethereum Documentation" Retrieved from Wiki.parity: https://wiki.parity.io/Proof-of-Authority-Chains.

[11] P. Szilágyi. "Clique PoA protocol". Retrieved from: http://eips.ethereum.org/EIPS/eip-225, March 2017.

[12] Enuma Technologies, "Rolling your own Proof-of-Authority Ethereum consortium", August 2017. Retrieved from: https://blog.enuma.io/update/2017/08/29/proof-of-authority-ethereum-networks.html