BLOCKCHAIN MEETS IOT: AN ARCHITECTURE FOR SCALABLE ACCESS MANAGEMENT IN IOT

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

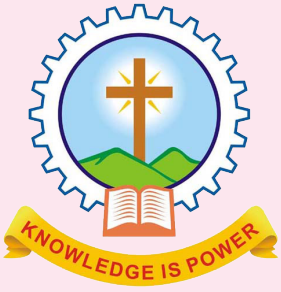
COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

AJAY AJITH



Department of Computer Science & Engineering
Mar Athanasius College Of Engineering
Kothamangalam

BLOCKCHAIN MEETS IOT: AN ARCHITECTURE FOR SCALABLE ACCESS MANAGEMENT IN IOT

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

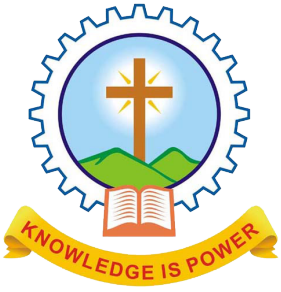
COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

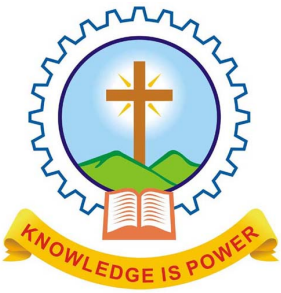
Submitted By

AJAY AJITH



Department of Computer Science & Engineering
Mar Athanasius College Of Engineering
Kothamangalam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM



CERTIFICATE

This is to certify that the report entitled **Blockchain meets IoT: An architecture for scalable access management in IoT** *submitted by* **Mr. AJAY AJITH, Reg. No.MAC15CS005** *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering Engineering from APJ Abdul Kalam Technological University for the year 2018 is a bonafide record of the seminar carried out by him under our supervision and guidance.*

.....	.....	.....
Prof. Joby George	**Prof. Neethu Subash**	**Dr. Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head Of Department***

Date:

Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the 'God Almighty' for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby george, Prof. Neethu Subash and Prof. Joby Anu Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

The Internet of Things (IoT) is stepping out of its infancy into full maturity and establishing itself as part of the future Internet. One of the technical challenges of having billions of devices deployed worldwide is the ability to manage them. Many access management technologies exist in IoT. They are based on centralized models which introduce a new variety of technical limitations to manage them globally. The idea of proposal is a new architecture for arbitrating roles and permissions in IoT. The new architecture is a fully distributed access control system for IoT based on blockchain technology. The architecture is backed by a proof of concept implementation and evaluated in realistic IoT scenarios. The results show that the blockchain technology could be used as access management technology in specific scalable IoT scenarios.

Contents

List of Figures

LIST OF ABBREVIATION

IOT	Internet Of Things
CoAP	Constrained Application Protocol
JSON	JavaScript Object Notation

INTRODUCTION

With a predicted 18 billion devices by 2022[1], Internet of Things (IoT) has become a technology with large influence across many vertical markets. It is foreseen that many IoT services will provide global reach across millions of simple and sometimes tiny devices. Besides that, the constrained capabilities of many IoT devices, as well as the current access control systems based on centralized and hierarchical structures, create new challenges in the IoT domain.

Centralized access control systems otherwise known as the client/server paradigm were designed to meet the needs of traditional human-machine oriented Internet scenarios where devices are within the same trust domain, which usually requires centralized access management. However, some IoT scenarios are much more dynamic than the traditional scenarios in which IoT devices may be mobile[2] and belong to various management communities during their lifetime. On the other hand, IoT devices can be managed by several managers at the same time. Moreover, many IoT devices and constrained managers will be too limited[3] in terms of CPU, memory and battery resources to be able to operate properly using the current systems. Henceforth, new ways of approaching the problem are needed

This paper aims at presenting a new architecture for managing IoT devices. The architecture provides a decentralized access control system connected to geographically distributed sensor networks. The solution is based on blockchain technology whereas the access control policies are enforced by adopting blockchain, this solution eliminates centralized access management. On the contrary, a single centralized access control server might become a bottleneck when access control queries and updates are frequent.

In contrast to other centralized system proposals[4], this approach brings the following advantages to access control in IoT:

1. Mobility : The architecture can be used in isolated administrative systems or domains. Thus, every administrative domain has its own freedom to manage the IoT devices while the access control policies are still enforced by the rules in the blockchain

2. Accessibility : In some IoT systems the constrained managers may use sleeping patterns that make it infeasible to constantly access them directly . This solution makes the access control rules available at any time. In addition, failures in some administrative servers do not ruin access to the information; all access control information is distributed.

3. Concurrency: A constrained device can have multiple managers at the same time, and all of them can access or modify the access control policies concurrently.

4. Lightweight: The IoT devices do not need any modification to adopt our solution. Besides, the communication between the managers and IoT devices happens through the blockchain network enabling cross platform communication.

5. Scalability: A constrained manager can still handle multiple IoT devices using our solution due to the fact that the IoT devices do not access the access control information directly from the managers. Furthermore, our solution supports numerous IoT devices connected through different constrained networks to a single blockchain.

6. Transparency: The system hides the location of the IoT devices and how a resource is access

In particular, the seminar contributes to the design of a new decentralized access control architecture for IoT using blockchain technology. This approach differs from other solutions in the way that it applies a specific design to avoid integrating blockchain technology into IoT devices. This increases the usability of our solution in a vast number of IoT scenarios with limited capabilities. As opposed to other solutions, the design operates in a single smart contract, simplifying the whole process in the blockchain network and reducing the communication overhead between the nodes. Additionally, the access control information is provided to the IoT devices in real time. In summary, the adoption of blockchain technology in this

approach has been specifically designed to better scalability and to achieve better results than current solutions in lightweight IoT scenarios.

1.1 Blockchain

Bitcoin's public ledger the blockchain was first introduced in 2009 by Satoshi Nakamoto[5]. Bitcoin was the first widely used implementation of peer-to-peer trustless electronic cash. Thenceforth, many other forms of electronic cash (call cryptocurrencies) have been created using similar structures. At the same time, different applications using blockchain have been developed over the years to implement other scenarios beyond cryptocurrencies. New concepts, such as smart contracts and smart properties, have entered the scene. Smart contracts[6] are computer protocols that facilitate, verify, or enforce the negotiation or performance of a contract. They provide the ability to directly track and execute complex agreements between parties without human interaction. On the other hand, smart properties are agreements whose ownership is controlled via the blockchain, using contract

The potential uses of blockchain technology go beyond Bitcoin. Blockchain technology has the following properties:

1. Decentralized Control : A decentralized scheme in which no central authority dictates the rules

2. Data transparency and Auditability: A full copy of every transaction ever executed in the system is stored in the blockchain and is public to all the peers.

3. Distribute Information: Every network node keeps a copy of the blockchain to avoid having a centralized authority privately keep all that information.

4. Decentralized Consensus: The transactions are validated by all the nodes of a network instead of a central entity. This breaks with the paradigm of centralized consensus

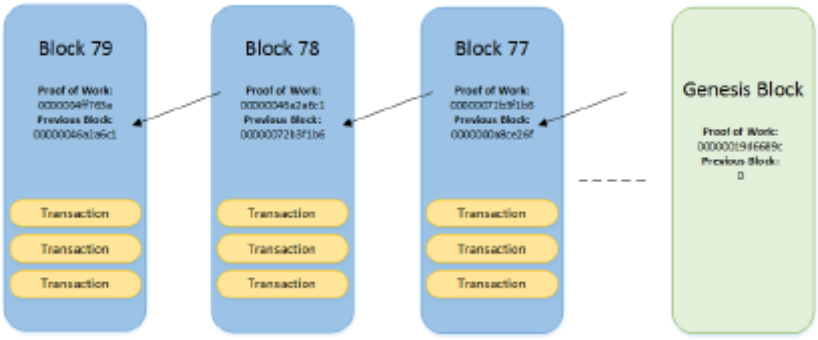


Figure 1.1: Blockchain Structure

1.2 Blockchain Technology

The blockchain[5] is a distributed database that does not need a central authority and eliminates the need for 3rd party verification. A blockchain contains a set of blocks, and every block contains a hash of the previous block, creating a chain of blocks from the genesis block to the current block. A genesis block is the first block in a blockchain. The genesis block is almost always hardcoded into the software. It is a special case in that it does not reference a previous block. For any block on the blockchain, there is only one path to the genesis block. Coming from the genesis block, however, there can be forks. Forks are generated when two blocks are created just a few seconds apart. When that happens, the latest block in the longest valid chain is always chosen. The longest valid chain is calculated based on the combined difficulty of that chain, not the number of blocks. The blocks in shorter chains are considered invalid chains and are often called orphan blocks

Blocks have a set of transactions. A transaction is a transfer of values between different entities that are broadcast to the network and collected into the blocks. All transactions are visible in the blockchain. The transactions are mined into a block by the so called pool miners or solo miners. The pool miners technique is a mining approach where multiple devices called miners contribute to the generation of a block. Pool miners or solo miners are entities that add transaction records into the blockchain. That process is called mining. Mining is intentionally

designed to be resource-intensive and difficult

Individual blocks must contain a proof of work (PoW)[7] to be considered valid in the blockchain. The PoW is verified by other miners each time they receive a block. The primary purpose of mining is to allow the nodes in a system to reach a secure, tamper-resistant consensus. Mining is also the mechanism used to introduce new cryptocurrency (e.g. Bitcoins) into the system. The miners are paid a transaction fee as well as a determined amount of newly created coins when they validate a block. This method serves the purpose of disseminating new coins in a decentralized manner as well as providing security to the system. The system automatically adapts to the total mining power of the network keeping it constant to a specific amount of time (e.g. 10 minutes in Bitcoin). The difficulty target of the PoW is also adjusted after every certain amount of blocks (e.g. 2016 blocks in Bitcoin) based on the network performance. A transaction takes time to reach all the nodes in the network, and the delay ensures that all the transactions are verified by all the nodes in the network, to prevent the so called double spending problem. Double spending is the result of using some cryptocurrency more than once at the same time.

Consensus is a fundamental problem in distributed systems that requires two or more agents to mutually agree on a given value needed for computational purposes. Some of these agents may be unreliable, and therefore the consensus process needs to be reliant. Blockchains can use various consensus algorithms. Some of them include proof of work (PoW), proof of stake (PoS), proof of storage[8], proof of burn, or proof of capacity[9] among others.

The PoW of every block guarantees a specific level of difficulty to generate a new block, and the decentralized consensus enforces the validity of every block in the blockchain. If there is consensus to accept a new block, the new block will be added into the blockchain and all the miners will have to start mining using that block as a reference. Each block is computationally impractical to modify once it has been added into the blockchain because the whole blockchain would also have to be regenerate

Proof of stake (PoS)[10] is a proposed alternative to PoW. PoS build on the notion that only those holding assets in the system may participate in the consensus process growing the blockchain. While the PoW method forces miners to repeatedly run expensive hashing algo-

rithms to validate transactions, PoS asks users to prove ownership of a certain

1.3 Blockchain Implementaion

Blockchain technology can be used in a myriad of ways rather than just as a digital currency system e.g. using blockchains as the underlying technology to build software. This section describes some of the popular blockchain systems and their salient features. The following systems mainly focus on building software applications on top of blockchain technology.

1. Bitcoin : Bitcoin[5] was the first blockchain to be conceptualized and implemented, and it is a cryptocurrency that serves as a digital financial asset. Bitcoin uses public key cryptography, peer-to-peer networking and proof of work to make transactions and verify them. The Bitcoin system is programmed so that a new block is created once every 10 minutes. If a fork is not a part of the longest computationally chain, it becomes a stale block.

   It is worthy to note that there are no balances in Bitcoin, or rather there are unspent transaction outputs (UTXO) in the blockchain. Whenever some bitcoins are received, they are recorded as UTXO. Thus, sending someone a bitcoin actually means creating a UTXO corresponding to the receiver's address. A transaction output typically consists of two fields, namely the amount and a locking script. The locking script sets out conditions that need to be fulfilled in order to spend the UTXO. A satoshi is the smallest denomination of the amount that can be sent.

2. Ethereum : Ethereum[10] was designed in 2013 by a Bitcoin developer Vitalik Buterin, who wanted to build a platform to facilitate the development of decentralized applications on top of the blockchain. Ethereum has its own cryptocurrency called ether and an internal currency to pay for computations and transaction fees called gas. The decentralized applications can be programmed with a built-in Turing complete language called Solidity. A Turing complete language refers to a programming language that can solve any computational problem if enough time and space are provided.

Ethereum uses PoW as its consensus mechanism, but it is soon switching to PoS. The basic build of the proof of work algorithms that Ethereum currently uses is a memory- hard hashing algorithm called Dagger-Hashimoto. The block creation time is significantly lower than in many other systems and amounts to approximately 12 seconds. Since a lower block creation time leads to a higher rate of stale blocks, the system uses GHOST protocol[11] to consider the heaviest computational chain as the main blockchain. The heaviest chain in this case includes the stale blocks as well.

3. Rootstock

   Rootstock2, a new open source platform, is very similar to Ethereum in terms of creating smart contracts on a Turing complete smart platform, except that it utilizes the Bitcoin ecosystem to do so. The advantages to this platform are that it exists as a Bitcoin sidechain and is backward compatible with the Ethereum virtual machine. This means that all Ethereum contracts can easily run on Rootstock. Their biggest advantage, however, is the fact that they can be merged-mined with Bitcoin, thereby making Rootstock as secure. A sidechain is a separate blockchain whose assets can be transferred to and from the main blockchain, i.e. the Bitcoin blockchain in this context.

4. Hyperledger : Hyperledger 3 is a project hosted by the Linux Foundation as a cross industry collaborative project. The system was designed with the enterprise architecture in mind with customizable networking rules that help different consensus protocols operate. It borrows the UTXO and script-based logic from Bitcoins, as described in I-C1, and uses practical Byzantine fault tolerant (PBFT)[12] consensus protocol instead of the proof of work algorithm. PBFT is known to process thousands of requests per second with a latency increase of less than a millisecond.

RELATED WORKS

Developing solutions for the IoT requires collaboration among different technologies. A common theme in this paper is the combination of blockchain, access control and Internet of Things. This section explores the extent of those technologies in IoT by looking at previous research carried out for the topic.

2.1 Blockchain and Internet of Things

Conoscenti et al[18] conducted a systematic literature review on the blockchain for the Internet of Things. The survey described several papers that manage data collected by IoT devices. As an example, [19] describes a system to verify the identity of the data and [20] describes a method to preserve the data ownership of the IoT devices. Unlike the idea presented here, none of the papers in the survey propose an architecture where managers can manage the entire lifecycle access policies of the IoT devices regardless of their location or provenance.

The only previous work related to our solution is [21], which describes a cryptocurrency blockchain-based access control framework called FairAccess. However, there are several differences between them. This work focuses on creating a single smart contract to define the policy rules of the management system. The access control policies are defined creating transactions towards that smart contract. In contrast, the work in [21] creates a different smart contract for the access control policy of every resource-requester pair. Second, [21] includes the IoT devices in the blockchain. We focus on a wider number of IoT devices that do not have the capabilities to run the blockchain technology in their systems. Third, this system provides the access control information to the devices in real

time.

2.2 Access Control and IoT

Reference [22] provides an extensive review of different access control solutions in IoT. The survey identifies the current access control mechanism used in IoT and argues that commonly used Internet protocols cannot, in every case, be applied to constrained environments.

Based on its extensive literature review, [22] identifies and lists three decentralized authorization and access control solutions: DOAuth (Decentralized Open Authentication), FairAccess[21], [23], and the IBM Adept (Autonomous Decentralized Peer-to-Peer Telemetry)[15] framework. However, the same reference classifies the OAuth-based access control solutions as a heavyweight protocol for IoT scenarios due to its communication and processing overheads. Then again, the IBM Adept solution provides a messaging and file sharing framework to build IoT applications but it does not yet implement an access control mechanism. FairAccess, to the best knowledge of the authors, has some similarities with our solution but, as explained in the previous section, our solution is much more broad focusing on devices with limited capabilities to support the blockchain in their systems. Figure 7 shows a quantitative evaluation of the different access control solutions based on the evaluation method defined in [22]. The IBM Adept framework is not shown in the figure since it is not yet implementing any access control system. As the figure shows, our approach brought better results than the existing ones.

PROPOSED WORK

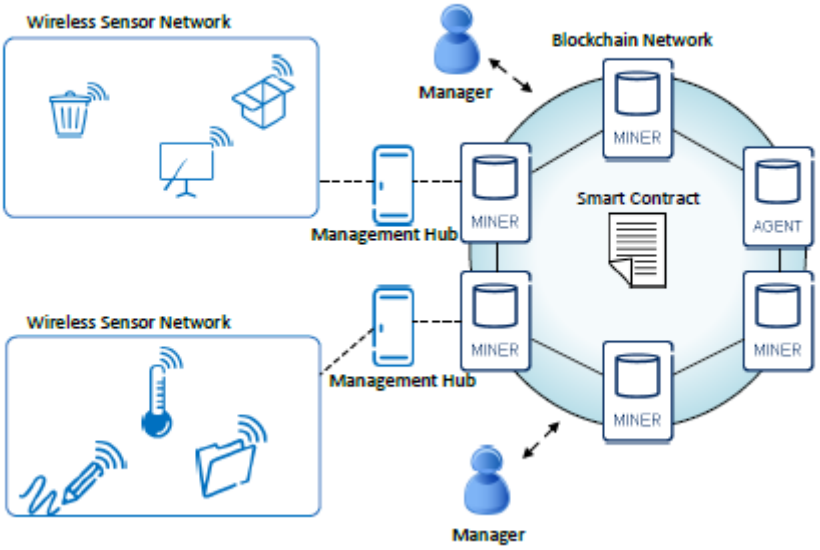


Figure 3.1: Decentralized access management system

The architecture proposed describes a new decentralized access management system where access control information is stored and distributed using blockchain technology.

All the entities will be part of blockchain technology except for IoT devices and management hub nodes. Nodes in a blockchain network must include a copy of the blockchain. The blockchain can be considerably large in size and will keep increasing over time. The majority of IoT devices will not be able to store blockchain information due to their constrained nature. Consequently, our architecture does not include IoT devices in the blockchain and, alternatively, defines a new node called management hub that requests

access control information from the blockchain on behalf of the IoT devices.

In addition to that, the solution involves a single smart contract that defines all the operations allowed in the access control system. That contract is unique and cannot be deleted from the system. Entities called managers interact with the smart contract in order to define the access control policy of the system.

3.1 System Architecture

The architecture can be divided into six different components:

3.1.1 Wireless Sensor Networks

A wireless sensor network is a communication network that allows constrained connectivity in applications with limited power and light requirements. Further, the IoT devices belonging to the wireless sensor network are limited in their computational power, memory, and/or energy availability.

IoT devices do not belong to the blockchain network. Consequently, one of the requirements of our architecture is that all the devices will have to be uniquely identified globally in the blockchain network. Public key generators can provide a feasible solution for the problem producing acceptably large and unique random numbers. Typically, using the existent IoT cryptographic technologies would automatically create a public key for every device. Hence, enforcing encryption connections will ensure unique identifiers. In fact, current IoT communication protocols such as CoAP[13] already support secure channels through DTLS[14].

3.1.2 Managers

A manager is an entity responsible for managing the access control permissions of a set of IoT devices. Normally, managers are considered lightweight nodes in our system. Lightweight nodes do not store the blockchain information or verify the blockchain's

transactions as the miner nodes do. As a result, constrained devices can also become managers in our system without representing an impediment to their hardware limitations. In addition, managers using our approach do not need to be constantly connected to the blockchain network, which helps to decrease the usage of their hardware resources.

Any entity can be registered as a manager. However, the devices registered as IoT devices have to register under a manager's control. That is done to avoid managers from registering to devices under their control without the permission of the devices. In addition, all registered IoT devices in the system have to belong to at least one registered manager. Otherwise, nobody would be able to manage that device. A registered IoT device can belong to multiple managers at the same time. After registration of the IoT device under the manager's control, the managers can define specific access control permissions for them.

3.1.3 Agent Node

The agent node is a specific blockchain node in our architecture responsible to deploy the only smart contract in our system. The agent node is the owner of the smart contract during the lifetime of the access control system. Once the smart contract is accepted into the blockchain network, the agent node receives an address that identifies the smart contract inside the blockchain network. In order to interact with the smart contract, all the nodes in the blockchain network need to know that smart contract's address.

3.1.4 Smart Contract

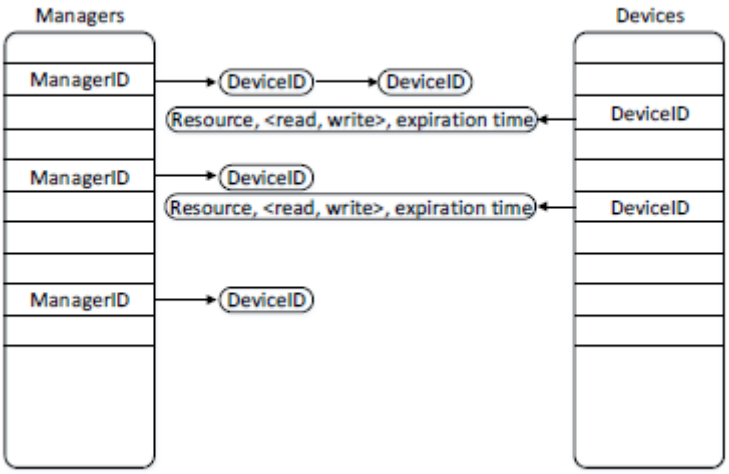


Figure 3.2: Decentralized access management system

The access management system described is governed by the operations defined in a single smart contract. This smart contract is unique and cannot be deleted from the system. Hence, all the operations allowed in the access management system are defined in the smart contract and are triggered by blockchain transactions. Once an operation is triggered through a transaction, the miners will keep the information of the transaction globally accessible. The smart contract and its operations are also globally accessible.

In addition to that, it has to be taken into consideration that managers are the only entities with the ability to interact with the smart contract in order to define new policies in the system.

3.1.5 Blockchain Network

The blockchain network in this architecture is a private blockchain for the sake of simplicity. A private blockchain is chosen because all the elements of the prototype are more

dimensioned, providing more reliable results when evaluating the system. However, in a real scenario, a public blockchain should be used to facilitate the adoption of the solution

Private blockchains are those that can be read by anyone but only written by private nodes. The miners in the network help keep the network secure and stable by approving transactions and keeping copies of the blockchain. Nodes can use the blockchain interface to store and globally access the access control policy of specific devices. The information is fully decentralized and tamper-proof.

3.1.6 Management Hub

As mentioned before, IoT devices do not belong to the blockchain network. The majority of IoT devices are very constrained in terms of CPU, memory and battery. Those limitations restrict IoT devices to be part of the blockchain network. Being part of the blockchain network implies keeping a copy of the blockchain locally and a track of the network transactions. Even though there are lightweight solutions that do not keep the entire blockchain information locally and rely on other nodes[15], all those lighter solutions are still too heavy for the majority of IoT devices. In consequence, we opted to use a node called management hub. The management hub is an interface that translates the information encoded in CoAP messages by the IoT devices into JSON-RPC messages understandable by the blockchain nodes. The management hub is connected directly with a blockchain node, for instance, a miner. Multiple sensor networks can be connected to a management hub node and multiple management hub nodes can be connected to the same blockchain node. IoT devices will only be able to request access information from the blockchain using the management hub.

Management hub nodes cannot be constrained devices. Such devices need high performance characteristics to be able to serve as many simultaneous requests as possible from the IoT devices.

In the simplest case where authentication is not needed, any IoT device will be able to connect to any management hub directly and access the blockchain network. However,

in many situations an access control is needed. In such a case, the IoT devices will only be able to connect to some specific management hubs. After an IoT device is added into the system, the manager node of that device will have to inform the specific management hub node about the credentials of that device, as well as, inform the device about the location of the management hub node.

3.2 System Interactions

This section explains the different interactions between the different components of our architecture. As shown in Figure 3, the interactions can be divided into four different phases: network

During this phase, the access management system is created in the blockchain network. Upon the creation of the blockchain network, the agent node deploys the smart contract into the blockchain network. This single smart contract defines all the operations of the access control management system. Once the smart contract is accepted into the blockchain network, the agent node receives the address of the smart contract. The address is used to identify the smart contract in the access management system and other components of the blockchain network need the smart contract's address to interact with it. For instance, all the managers in the system will interact with this single smart contract to register as managers or modify the control access rules of the IoT Devices.

Figure 3 shows how a manager and a management hub discover the address, querying the Agent node. Typically, that would be one possibility to obtain the information. However, in our implementation, that information is hard-coded into those components for simplicity.

The management hub will connect with the nearest available node in the blockchain network, a miner node in Figure 3. That miner hosts a personal copy of the blockchain. In addition, it enables the RPC port for listening for requests and lets management hubs connect to it. Management hubs also have to have a way to find the available nodes next to them. That information could be obtained from a centralized system in Internet, but in our particular implementation, is set manually in every management hub.

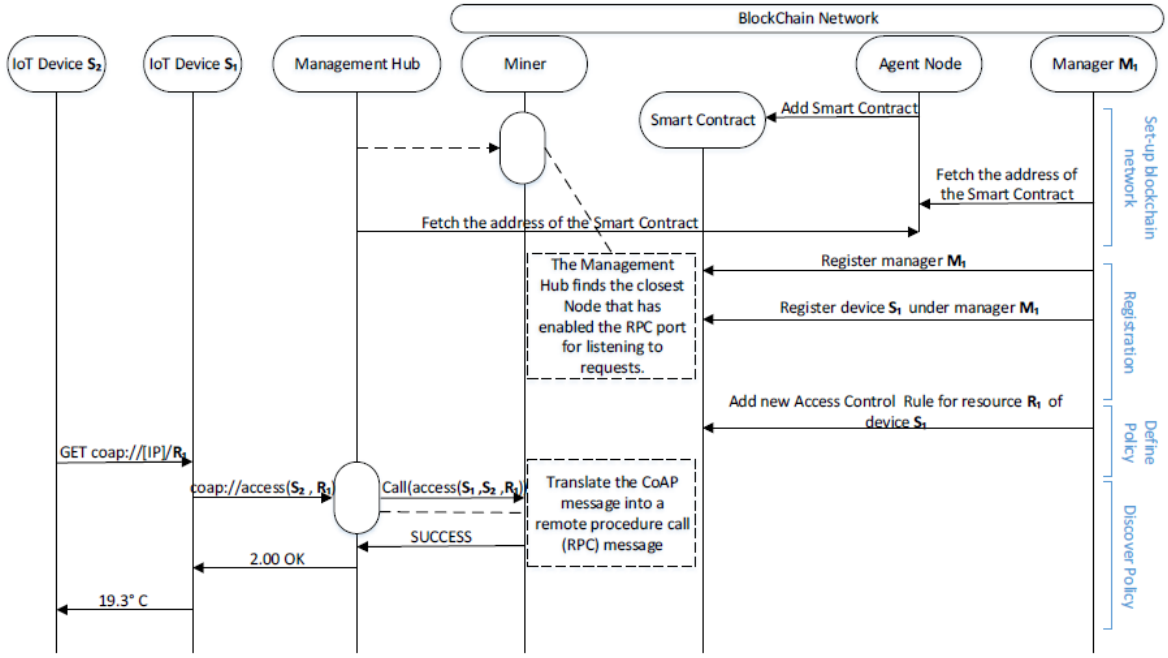


Figure 3.3: Network Set-up, Registration, Definition and Discovery of the Policy

3.2.1 Registering the managers and IoT devices into the system

Any blockchain node in the access management system can be registered as a manager. In order for a blockchain node to register itself as a manager, it needs to know the address of the smart contract. Once it obtains that information, it can register itself sending a transaction to the function defined in the smart contract. Thereafter, the manager will receive the address of its registration once the transaction is successfully accepted into the blockchain. That address will identify the manager in the access management system.Manager nodes can also register IoT devices under a manager's control. There is no limitation of the number of managers an IoT device can have. Thus, an IoT device can have several managers at any time. As in the previous case, the manager will receive an address of the registered device that will be used to identify the device into the access management system. IoT devices should be able to verify the registration under a manager before the operation is accepted in the blockchain. Otherwise, any manager could register any device under its control. For the sake of simplicity, our implementation eludes that verification. That assumption makes our system substantially insecure in case of a

malicious manager, but our goal was to prove the feasibility of the architecture rather than the security.

3.2.2 Management Modification

As explained before, every IoT device has to belong to at least one manager. In addition, our system supports a multiple number of managers controlling the same device. There are multiple ways in our system to transfer the management control from one manager to another or to add or delete several managers from the system.

In our prototype, we choose one of the simplest options in which every manager node in the system can remove itself as a manager of the devices it controls. On the contrary, managers cannot delete other managers from the system. The system will always let a manager to remove itself from an IoT device as long as the IoT device is under the control of at least another manager node. Otherwise, the smart contract will not allow the operation and will be canceled.

On the other hand, only the manager nodes that control an IoT device can register other managers under the control of that device. One of the advantages of our solution is that transferring the management control of an IoT device is a simple process due to the fact that all the operations in the system are defined and enforced using a single smart contract and managers do not need to interact with each other. The managers only need to know the device's address and the blockchain address of the smart contract to be able to modify the management relationships.

3.2.3 Policy Definition

Managers can define access control rules for the resources of their IoT devices. The permissions can be defined in many ways. However, the permissions in our implementation list the devices entitled to access a particular resource. Thenceforth, managers not only need to know the address of the devices under their control but also the address of the devices authorized to access their IoT devices. Managers can enforce the policy creating a transaction towards the smart contract with all that information.

Similar to the policy definition, Managers can modify and delete policies at any time.

The method is similar to the method described in the policy definition. If a manager adds an existing policy using the an operation, that policy gets modified automatically

3.2.4 Discovering the policy

When device S2 in Figure 2 wishes to access a resource hosted by device S1, S2 sends a CoAP message requesting the resource information of S1. S1 consequently, can request the access control information of S2 through the management hub. Before an IoT device can connect to the closest management hub, the device first needs to discover the hub's IP address. There can be several mechanisms for discovering the closest management hub node but the method used in our implementation assumes a default location for every device.

The management hub then translates the device's message into an RPC message[16] and sends it to the miner in the blockchain network attached to it. The operation queries the information from the blockchain stored in the miner. Essentially, that means that the operation is not a transaction and is not stored in the blockchain. As a result, the operation is processed immediately and does not incur any fee. Once the miner informs about the access policy of S1 to the management hub, the management hub translates the answer back to S1. S1 acts accordingly depending on the information received by the management hub. Figure 3 shows a successful answer, and therefore, S1 sends the information of the resource to S2

SYSTEM LIMITATIONS

The adoption of the blockchain technology in our solution improves the way IoT devices can be managed. However, the blockchain technology has some technicalities that could limit the solution proposed in this paper. This section explains those technicalities and describes methods to overcome them.

4.1 Cryptocurrency fees

Cryptocurrency fees are a fundamental part of blockchain-based computing platforms. All the transactions include a fee, and miners are awarded with certain amount of cryptocurrency money if they successfully manage to include one of their mined blocks into the blockchain. In some public ledgers there is a minimum fee amount required for a transaction to be accepted. That is a method used by some systems to avoid unwanted spam transactions.

In this architecture, IoT devices do not belong to the blockchain network themselves. The management hub nodes translate the messages from the devices into RPC messages and forward them to the blockchain network. In fact, querying information from the blockchain does not incur any fee and, therefore, the management hub can request information freely from any device. However, only the manager nodes will be able to create transactions on behalf of the devices and will also have to pay the transaction fees on behalf of them.

4.2 Processing Time

It is a fact that transactions take time to get accepted into the blockchain. As of now, BitCoin's[5] transactions can take up to 10 minutes and 12 seconds in Ethereum[10]. However, as stated before, the management hub nodes do not need to use transactions to request the information from the blockchain network. The management hub can query the information

immediately from the blockchain's node attached to it. Thus, the management hub can provide the information to the devices in real time. However, the transactions created by the manager nodes might incur long delays. In some situations, it might be inadequate to wait for such time. For instance, in a situation where a manager node grants or denies access to a particular resource in a device. An unauthorized attacker could gain access to the restricted information before the revoked operation by the manager node is spread and accepted by the majority of the miners.

One possible solution to overcome that disadvantage is the introduction of an expiration date parameter in the operations of the smart contract. This way, the policy rules can expire automatically after a certain time. Since time is not really specified in the blockchain, the expiration time can instead be translated as the number of accepted blocks into the blockchain. Therefore, a certain rule can expire after a specific number of mined blocks. This solution does not solve the case in which a rule in a smart contract has to be revoked immediately before its expiration time. Those specific cases are still very hard to solve using the blockchain. The best way would be to include higher transaction fees for the revocation operations to minimize the time spent adding the revocation operation into the blockchain network.

SECURITY ANALYSIS

Security is of paramount significance in any management system and our system should not be an exception. Even though the design of our system aims to facilitate the access control of resources in constrained scenarios, the solution should provide a satisfying level of security. In this section, we identify the main possible threads in our architecture and provide solutions to guarantee the best level of security.

5.1 STRIDE model

To identify the threats in our system, we use the STRIDE model[17] by asking whether one or more of the thread types apply. STRIDE classifies the threats into six categories, and its acronym derives from them: spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privileges.

As shown in the table below, each of the elements of our architecture is susceptible to a set of threats

	S	T	R	I	D	E
Management Hub	X	X	X	X	X	
Manager	X					
IoT Device	X					

Figure 5.1: STRIDE model diagram

As a result, even though the blockchain technology provides a certain level of security such as integrity and reliability of the data, IoT devices are not part of the blockchain network and have to rely their access control decisions on the management hub nodes. A malicious management hub could spoof (impersonate a management hub), tamper (modify the access control

information sent to the IoT devices), repudiate (claiming to have not performed an action), DoS (degrade the information sent to an IoT device) or disclose unauthorized information of the IoT devices. Signed certificates could solve that issue. The management hub nodes could get signed certificates from a certificate authority and the IoT devices could verify the authenticity of the management hub nodes through them.

Moreover, since the queries from the IoT devices to the management hub nodes are not transactions on the blockchain for obvious performance benefits, the blockchain loses the ability to verify which access control rules are implemented properly by the management hub nodes. That information could be stored locally in every management domain where the IoT devices reside. Since the IoT devices could be part of different management domains during their lifetimes, that information would be spread among many nodes making it difficult to audit and track it. For critical access control systems, the blockchain could force the management hub nodes to use transactions instead of queries. This solution incurs a performance penalty but could increase security in the system in some particular cases.

On the other hand, the discovery of the closest management hub node in a network should be reliable, as should the discovery of the address of the smart contract by the managers and the management hub nodes. That information could be queried from the agent node or stored privately on a trusted network accessible storage.

Furthermore, once an IoT device registers in the system for the first time, a malicious manager in the blockchain could claim control of that device. However, an IoT device should verify the registration under a manager before the operation is accepted in the blockchain. Otherwise, any manager could register any device under its control. A similar threat could occur when a malicious IoT device impersonates another device. In this case, the communication between the devices is done through DTLS which prevents spoofing as well as eavesdropping and tampering.

CONCLUSION

This study address the scalability problem of managing access to billions of constrained devices in the IoT. Certainly, centralized access control systems lack the ability to deal with increased load efficiently. The paper introduces a new access management system that mitigates the issues associated with managing numerous constrained IoT devices. The solution is fully decentralized and based on blockchain technology. Since the majority of IoT devices are largely constrained to support blockchain technology directly, the IoT devices in our design do not belong to the blockchain network which makes easier the integration of the current IoT devices to adapt to our system. The goal of this paper The goal of this paper was to provide a generic, scalable, and easy-to-manage access control system for IoT and to implement a proof of concept (PoC) prototype that proves our design. According to our implementation and evaluation, our solution scales well due to the fact that numerous constrained networks can be connected simultaneously to the blockchain network using specific nodes called management hub nodes. Additionally, the versatility of having different management hub nodes distributed around the whole blockchain network and connected in different ways to the constrained networks provides a considerably high flexibility to our solution. In general, our solution is able to adapt to various IoT scenarios confirming that blockchain technology can embrace IoT technology at its fullest.

REFERENCES

[1] S. K. Datta et al., "Vehicles as Connected Resources: Opportunities and Challenges for the Future", *IEEE Computer*, vol. 12, no. 2, 2017, pp. 26–35.

[2] H. Li et al.,"Engineering Searchable Encryption of Mobile Cloud Networks: When QoE Meets QoP,"]*IEEE Wireless Commun.,* vol. 22, no. 4, Aug. 2015, pp. 74–80.

[3] N. Kshetri, "Can Blockchain Strengthen the Internet of Things? "*IT Professional,* vol. 19, no. 4, 2017, pp. 68–72.

[4] A. Lei et al., "Blockchain-Based Dynamic Key Management for HeterogeneousIntelligent Transportation Systems," *IEEE Internet of Things J.,* 2017.

[5] N. Z. Aitzhan and D. Svetinovic, "Security and Privacy in Decentralized Energy Trading through Multi-Signatures, Blockchain and AnonymousMessaging Streams, " *IEEE Trans. Dependable and Secure Computing,* 2017.

[6] P. K. Sharma et al., "DistBlockNet: A Distributed Blockchains- Based Secure SDN Architecture for IoT Networks, " *IEEE Commun. Mag.,* vol. 55, no. 9, Sept. 2017, pp. 78–85.

[7] K. Zhang et al., "Mobile Edge Computing for Vehicular Networks: A Promising Network Paradigm with Predictive Offloading, "*IEEE Vehic. Tech. Mag.,* vol.12, no. 2, 2017, pp. 36–44.

[8] Z. Zhou et al., "Software Defined Machine-to-Machine Communication for Smart Energy Management, " *IEEE Commun. Mag.,*vol. 55, no. 10, Oct. 2017, pp. 52–60.

[9] X. He et al., "A Novel Load Balancing Strategy of Software- Defined Cloud/Fog Networking in the Internet of Vehicles, " *China Commun.,* vol. 13, Supplement 2, 2016, pp. 140–49.

[10] K. Sasaki et al., "Vehicle Control System Coordinated Between Cloud and Mobile Edge Computing, " *Proc. 55th Annual Conf. Society of Instrument and Control Engineers of Japan,* 2016, pp. 1122–27.

[11] X. Hou et al., "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures, "*IEEE Trans. Vehic. Tech.,* vol. 65, no. 6, 2016, pp. 3860–73.

[12] Y. Zhang et al., "Securing Vehicle-to-Grid Communications in the Smart Grid, " *IEEE Wireless Commun.,* vol. 20, no. 6, Dec. 2013, pp. 66–73.

[13] J. Kang et al., "Enabling Localized Peer-to-Peer Electricity Trading Among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains, " *IEEE Trans. Industrial Informatics,* 2017

[14] J. Ren et al., "Serving at the Edge: A Scalable IoT Architecture Based on Transparent Computing, " *IEEE Network,* vol. 31, no. 5, Sept./Oct. 2017, pp. 96–105.

[15] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System, "*https://bitcoin.org/bitcoin.pdf,*2008