KARYA ILHAM PUBLISHING
CONNECTING MINDS, BRIDGING IDEAS, INSPIRING READERS

Journal of Advanced Research in Computing and Applications

Journal homepage:
https://karyailham.com.my/index.php/arca/index
ISSN: 2462-1927

Integrity-Based Protection of Access Control Tokens using Keccak-256 in Smart Contracts for IoT

Noor Afiza Mohd Ariffin[1,*], Abdirahman Mohamed Aboubaker[1]

[1] Department of Computer Science, Faculty of Computer Science and Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 23 June 2025
Received in revised form 23 September 2025
Accepted 30 September 2025
Available online 2 October 2025

Keywords:
IoT; smart contract; access control; encryption

ABSTRACT

The Internet of Things (IoT) is transforming the world into a smarter and more adaptive system by seamlessly integrating the digital and physical realms. IoT devices are now utilized across diverse industries, ranging from smart homes, automotive services, and network sensors to more complex sectors like healthcare systems and smart cities. To address access control challenges in IoT systems, a notable scheme is the decentralized and trustworthy token-based capability access control structure (capBAC). This scheme leverages Ethereum smart contracts, creating a dedicated smart contract for each IoT device or object. The token manages access rights and actions, granting permissions to users and objects. CapBAC offers fine-grained, flexible management capabilities to ensure consistency between delegated information and token data. However, one significant limitation of CapBAC is the lack of integrity for access rights. Since tokens are stored in the blockchain without hashing, anyone can read the access rights and identify the object owner, posing critical challenges for integrity and authentication. To address this, we propose using the Keccak-256 hash algorithm to hash capability access tokens and safeguard their integrity. Keccak-256 is the recommended hashing algorithm in Ethereum, standardized for mobile telephony (TUAK) and the NIST standard. In the proposed solution, after an object owner creates and delegates a capability access token, the token will be hashed before being stored in the blockchain. This approach prevents attackers from reading blockchain storage and discovering token contents and access privileges, enhancing integrity and security.

1. Introduction

The Internet of Things (IoT) has grown so rapidly and caught everyone by surprise. IoT made non-traditional computing devices such as light bulbs and sensors, into an effective low-cost computing device, these devices can connect to the Internet for transmitting data, collecting instructions, or executing instructions. There are various devices that fall under the umbrella of the Internet-connected "smart" model's appliances, i.e., smart TVs, speakers, lighting systems, connected printers, HVAC systems, smart thermostats, weather forecast sensors, temperature detector sensors,

[*] *Corresponding author.*
E-mail address: noorafiza@upm.edu.my

https://doi.org/10.37934/arca.40.1.5062

and even military devices, these devices have become essential to this era and cannot be relinquished.

Furthermore, a cybersecurity firm based in Helsinki, Finland, performed a study on IoT rapid spread and discovered that, over the following two years, the number of IoT devices reaching homes would increase dramatically from nine devices to five hundred devices by 2022. So, IoT devices are tied into pretty much every product. As a result, the public ought to accept the change and adapt to it. Nevertheless, although these devices make our life easier and convenient, they still come with quite many challenges and risks, that companies will have to address them sooner than later. (IoT) devices are vulnerable to unauthorized access by malicious users, posing potential threats to user personal and property data, e.g., malicious users might know the contents of private conversations within homes by accessing certain fragile and weakly secured appliances.

One of several issues associated with IoT technology is that tech manufacturers design these devices carefree without giving managing the security risks involved with the devices any consideration. In facts, many of these gadgets and IoT products do not get enough updates, and in some cases, they do not get updated at all. This result in a device that had once been believed to be as secure and reliable when consumers first purchased it, to becomes insecure and gradually vulnerable device to attackers and other security issues. Even worse, the manufacturers use unsupported legacy Linux kernels which result in exposing their customers to potential attacks and virus, due to the outdated hardware and software.

The current increase of ransomware attacks would pose a significant security threat to IoT vendors and services. Attackers target IoT devices because of their poorly secure mechanisms and their complexity design, which makes it very challenging to propose a standardize security mechanisms for all IoT devices. Furthermore, the vast number of enterprises and manufacturers cater the IoT market. which be a catastrophic disaster for multiple businesses if the IoT is corrupted or compromised.

Brute-force attacks, and default passwords issues are another challenge associated with the IoT appliances. Many companies ship their devices with default passwords and not asking consumers to change them as soon as they obtain their devices. There are several policy, regulations, and report that advice vendors against selling IoT devices that have default passwords, like using "admin" as username or passwords. With that been said, these guidelines are now nothing more than suggestions, and there are no legal ramifications for urging manufacturers to drop this dangerous practice. In fact, poor authentication and login data leave almost all IoT devices vulnerable to hacking and passwords brute-force attack.

In October 2016, a hacker discovered a loophole to a particular security camera model. Nearly half a million Internet of Things (IoT) digital cameras recorders have been used as a botnet and targeted various social network platform and brought down Twitter and some other high-profile platforms for nearly two hours. This kind of attack is just an example of the important of securing these IoT devices. Several integrity incidents have been reported as well regarding connected cameras and devices in hotels and Airbnb's houses. In 2019, a hotel in-room robot Tapia, used as human personnel, have been hacked and used to spy on room guests. In a similar incident, Airbnb came under scrutiny in 2019 as well, after guests reported secret cameras filming them in the Airbnb houses, they rented out. As a result, four suspects were detained for taking hidden videos of visitors to motels and live streaming them to their audiences.

Different schemes were proposed to overcome these issues, one notable and interesting scheme is a decentralized and trustworthy token capability access control structure (capBAC) the scheme uses Ethereum smart contract which creates a smart contract for each object and IoT device. The token manages the access rights in units and grants access rights or actions by object and user. This scheme achieves fine-grained and flexible management capabilities to ensure cohesiveness in both information that has been delegated and the information in the tokens. However, one of the limitations and issues of (CapBAC) is law integrity of access rights. Since all token are stored in the block without hashing mechanism in place, anyone can read and find out the access right of each token and the owner of the object, and that itself creates a critical and crucial challenge in terms of integrity and authentication.

Thus, we proposed to use keccak 256 hash function to hash the tokens that are stored in the blockchain. The reason for using Keccak256 hash algorithm instead of other hash algorithms, is because, keceak 256 is the recommended hash algorithm in Ethereum. Keccak256 has been standardized for mobile telephony (TUAK) as well as the NIST standard. So, the idea of the proposed solution is that after the owner of the object creates a capability access token and delegated it to other users, the token will be hashed before storing it into the blockchain to ensure its integrity. By doing that, we can prevent attackers from reading through the blockchain storage and discover the token contents and access privileges.

Smart contract is a set of codes and data that resides and run programs on the Ethereum blockchain. Smart contract code is stored at a specific address on the Ethereum blockchain which implies that it could send a transaction over the network. Nevertheless, the code is not managed or supervised by third parties since all smart contract codes are computer automated and enforced agreement, some sections do require human input and control, to improve the efficiency of transactions. Nonetheless, most of the smart contracts are self-executed and does not require any user's interference. Smart contracts incorporate protocols and user interfaces to formalize and protect relationships across computer networks. Smart contracts are created as scripts and deposited on the blockchain each smart contract has its unique address in the blockchain. There are many security benefits from using the smart contract. One notable benefit is the autonomy the user makes the agreement without relying on third parties or broker, which knock out the danger of third-party manipulating user's data or sell them to advertisement agencies.

2. Research Background

At the beginning of our research, we studied and compared different access control mechanisms used in IoT ecosystems. The most popular and most used solutions are Blockchain and smart contract since both these technologies are very reliable and trustworthy.

Many of these mechanisms use different mechanisms but most noticeable and common mechanisms are attribute based access control (ABAC), rules-based access control (RBAC), and capability access control (CapBAC). The main different between RBAC and ABAC are the former assigns permissions based on roles, and the latter is attribute-based, which means granting access based on certain attributes that may change over time. As for (capBAC) it grants access based on the token that the subject has received form the owner of the object. The token contains the access right of the subject and what kind of access is granted i.e., (Read, Write, or Execute) For instance, if subject x wants to access a specific object the subject should have the token to access that object otherwise, he/she cannot access it. Our research focuses on the CapBAC method since it considered the most successful and popular solution.

ABAC has been developed to address the most critical government entities. ABAC has been used in numerous places, including the US Department of Defence (DoD), the UK Ministry of Defence (MoD), and quickly becomes a NIST standard. ABAC places the hashing and authentication mechanisms into the data itself, meaning that the hashing protects the data from being compromised and exploited outside of its intended user, even if stolen or flat-out hacked. ABAC uses different hashing method, most notable and common method is the usage of attribute-based hashing (ABE) Objects are hashed on the basis of attribute-based access policies. ABE consists primarily of two types ABE (KP-ABE) and ciphertext-policy (CP-ABE). The entity is hashed in KP-ABE based on the collection of attributes belonging to an object, which must pass a policy encoded in the user's key to verifyion the object to continue. CP-ABE is the opposite of KP-ABE by using an attribute-based policy to encrypt an object and providing a user key consists of a collection of attributes relevant to that object User.

RBAC on the hands, is the second oldest form of access control method after ACL (access control list). As we explain earlier, RBAC is based on assigning rules to the employees so, the administrator will be determining the access privileges of each employee in the organization and whether or not the particular employee or set of employees need a modification right or just restricted it to read only. One of the benefit of RBAC is that its more affective compared to ABAC, the roles can be added or removed easily. And new roles can be added effortlessly. Moreover, the access control of sensitive information and data can be controlled effectively and lower the risk of data breach by applying the least privilege concept in this way, the organizations can avoid any kind of cyber-attacks. Nevertheless, one of the issues of RBAC is that it does not have any hashing mechanisms in place. So, if an intruders expose the role of certain employee, the intruders can gain access to all sensitive information. Furthermore, it's pretty difficult for the admin to manage all type of roles that is needed for a particular team or project. Which ABAC has the edge over the RBAC thanks to its automated features.

CapBAC method is designed according to the Capacity Based Authorization Model (often time referred to as capability-based security). Capacity is a communicable, significant token of authority. It compares an attribute that uniquely refers to an object and a related array of access rights. The owner of the object creates the capability access token to be used by other users. The token's functionality is to Grant a particular set of users the right to interact and access the object in specific ways. However, one of the issues of capBAC is most of the time, the token is stored as a plaintext, which means the object ID and the access rights are exposed to attackers, and that can result in a data breach or even data modification. Nonetheless, capBAC is a new and promising access control method that many organizations and researchers start exploring and finding solutions to the limitation. The remining of the chapter, is going to discuss the different access control mechanisms proposed for IoT ecosystem.

Ouaddah *et al.,* [1] proposed a decentralized access control mechanism called Fairaccess. The mechanism uses a token for authorization. The token is a digital signature used to provide permissions for any requesters to interact with the certain object in a particular way i.e., execute right, modification rights and so on. The controller is responsible for determining access policies and generating access tokens based on the defined policies, and if the authorized users compile with these policies then they can interact with the object. The blockchain is utilized as a database that store all the access token policies and forming transactions. Furthermore, the blockchain is also used as a logging database, which means it can be utilized for auditing purposes. Nevertheless, the proposed system has some issues like only the owner can terminate or create new request which could causes a delay.

Zhang *et al.,* [2] proposed a collaborative attribute-based access control using blockchain. The proposed scheme deals with the unauthorized access issues in IoT devices. Blockchain generate the addresses to be utilized as a storage for access policies. The policies are used for authorization purposes as well as ensuring the trusted among the users in the platform. The system uses a verifiable controlled collaboration mechanism to detect any malicious and misbehaviours activities and also asks for additional information to be used for authorization for certain action i.e., modification. The system also uses Authority Node (AN) which is utilized as a constructed for computation tasks and to query or invoke the access rights. The researchers claimed that the security analytics shows that their scheme provides a reliably authorization access control. Moreover, the researchers have also constructed a prototype as a proof-of-concept that their scheme is reliable and suitable for different IoT use cases.

Alphand *et al.,* [3] proposed a decentralized access control system called IoTChain for IoT ecosystem. IoTChain integrates the object security architecture (OSCAR) framework with the Blockchain to allow owners to produce keys for users so that the owners could secure their resources from unauthorized access. The system also utilizes a structure called Authentication and Authorization Constrained Environment (ACE) which provide an authorization solution for requesters to access the IoT recourse pool. In IoTchain the owner defines certain access control privileges using smart contract, the smart contract is a self-execute code, that by calling upon it will create an access token to the requesters as long as the requesters meet the smart contract requirement defined by the owner. So, when the user makes a request to access an IoT recourses, the smart contract will check the user authorization token against the predefined policies, once the user is authenticated, he/she could verify the recourses and downloads it. However, the proposed system got few limitation regarding scalability and assessment performance.

Zhang *et al.,* [4] merged the access control with machine learning to propose a system that provides dynamic access control for detecting misbehavior activity from users. The proposed system contains three types of access controls namely, access control contracts (ACCs), Judge contract (JS) and register contract (RC). For a user resource pair, each ACC determines one access control mechanism and applies functions to update access policies. There is a revocation list for each resource to detect any misbehavior activities to the resources and stored in contract. This list outlines the user’s misconduct, such as a DDOS attack, when the user sends many requests in a short period. So, if the JS detect any user misconduct, the JS decided to give a penalty to the user. Thus, whenever a malicious user requests access to the resources, the ACC will be executed and reports it to the JS contract then, the JS would detect the malicious activity. Which then specifies the associated punishment, i.e., restricting user access for within a period of time, depending on the misbehavior judging process the user could even be banned permanently from accessing the resources. Lasty, the task of RC is to maintain the contracts between JC and ACCs. However, the system has not been applied in a real world, so we cannot determine how scalable it can be or whether the system is suitable to different IoT use cases.

Xu *et al.,* [5] proposed a system called BlendCAC that is a capability-based access control based on blockchain and smart contract. The resource owner would determine the access policies and service controls. The cloud service provider (CSP) is responsible for overseeing and controlling the access control and the resource owner's policies. Thus, when the user requests access to the resource pool, a capability access token will be created by the resource owner. The created capability access token will be stored in the smart contract. Once the user receives the token and uses it, the CSP will check the access token's legitimacy and decide whether the user deserves the access. However, the drawbacks of BlendCAC are that the revocation features can only be permitted and used by a manager level entity, so regular users cannot perform the revocation features, which could cost many delay and insecurity. Furthermore, BlendCAC cannot be considered a fully decentralized system since the resources owner relies on the CSP to handle the authorization process.

Fotiou *et al.,* [6] presented an IoT access control management based on Ethereum. The main idea behind the proposed scheme is that the users do not communicate directly with the IoT devices, nor do they communicate with the gateway. Instead, they communicate solely with the Ethereum network. Therefore, when the user requires access to an IoT device, he/she should have a minimum of one token to access the IoT device. So, the more access token the user acquired the higher access privileges he/she can obtain. In other words, it's like a video game, the higher the user level reach, the higher skills he/she gets. The gateway verifies the legitimacy of the access token and the recourse location against the predefined policies, while the smart contract checks the number of access token the user obtained to perform a specific action, e.g., modification. The main drawback of this scheme is that it is very likely for the intruders to perform the infamous 51% attacks on the system, which means the intruders can have many legitimate access tokens to fully control the scheme or cause severe damage to the recourses.

Di *et al.,* [7] proposed an approach to generate an access control management model. The policies that defined the access resources are store in the blockchain. There are two classes of transactions used in the access model. the first class is the Policy Creation Transaction (PCT). The resource owner establishes PCT to create and pass an access right if the user attributes meet access policies. In order to accept the access request, the resource owner makes an access decision when the owner assigned the users to perform actions on specific resources e.g., write permission. The second class is the Right Transfer Transaction (RTT). It is used by transferring access rights from the existing user to another user through a blockchain transaction. This mechanism is called access delegation, which is beneficial in a collaborative system. This approach is implemented based on the (XACML) standard using the bitcoin platform to establish a blockchain network for controlling users' access rights and an external authorization system.

Patil *et al.,* [8] Presented a lightweight access control system for smart greenhouse farms. The system utilizes a private blockchain and has a policy header which is accountable for handling all the ingoing and outgoing traffic. The proposed system assembles all the IoT nodes into one group, which then designate a random cluster head. The selected head would control the overall of the network. Thus, if a user needs to access a particular IoT device, he/she can do so but, all the outgoing and ingoing transaction are monitored, validated, and authorized prior been executed. Since the system utilizes policy header, the owner defined an access management list and based on the predefined policies. The access would be permitted or denied. Policy header is a great mechanism to cope and overcome the Proof of Work (PoW) high-cost computation challenges. The policy header continuously got renowned and updated every time a new block added to the chain. Moreover, if the owner adds a new policy to the list, the policy header would store in the blockchain will be updated simultaneously, which provide a highly secured and fine-grand access control management mechanism. Nevertheless, the primary concern of the system is scalability. If the system consistently receives enormous requests from users, the system might not handle all these vast requests, and a malicious user would be able to slape through and gain access to the blockchain network.

Novo [9] has presented a completely decentralized access control system for the IoT ecosystem. The system is based on blockchain technology, so the concept behind the proposed system is that the IoT devices are not included in the blockchain to avoid any network overheads. The system consists of three primary entities, managers, management hub, and node agent. Managers' responsibilities are to create and determine the system's policies, while management hubs can request permission from the blockchain on behalf of the IoT devices. The management hubs can do so by utilizing a function called "call method". Lastly, as the name suggests, the node agent's responsibility is to deploy the smart contracts in the system. The system employs a sole smart contract to execute all the necessary transaction requested from the IoT device. Thus, when a request is initiated from an IoT device, the management hub would request the blockchain on behalf of the requested IoT device to grain the access. However, before granting the access permission, the smart contract validates the IoT device and distribute the request to the blockchain network to approve it. If the blockchain network approves the IoT device, it then grains the permission to add the device to its network. Otherwise, it will disapprove the request. Nonetheless, the proposed system encounters a critical issue. If the manager is malicious or been compromised, the entire system would be in jeopardy.

Ren *et al.,* [10] proposed a mechanism that consist of five network layers. The first layer is WSN controls for the manufacturing process. The second layer is the ad hoc network which is used to collect manufacturing data and analyze them in local center. The third layer include edge center to handle access control privilege for users. As well as, building a decentralized blockchain network that holds the access control. In the fourth layer is used to validate the transactions. Last layer is IoT devices and it used to read access control policies from the previous layer. The mechanism utilizes an emerging access control and identity management in decentralized blockchain network for IoT devices to provide more secure, robust, and confidential.

3. Methodology

The research concept is to find a way to apply hash-based integrity verification to the capability access control for IoT devices. That is because many limitations have been exposed to the capability access control mechanism that has been proposed in numerous papers. Furthermore, storing the token in plaintext is not recommended, and many researchers and security experts are against such a practice. Thus, we first looked up the existing capability access control mechanism that uses blockchain and smart contracts in IoT ecosystems and found out their limitation and whether they included an hashing mechanism for the token in their proposal. Nevertheless, we must make sure that the proposed method would not affect the functionality of capability access control and cause any delay or overheated. So, In Figure 1 illustrate the process we going to go through in the research.

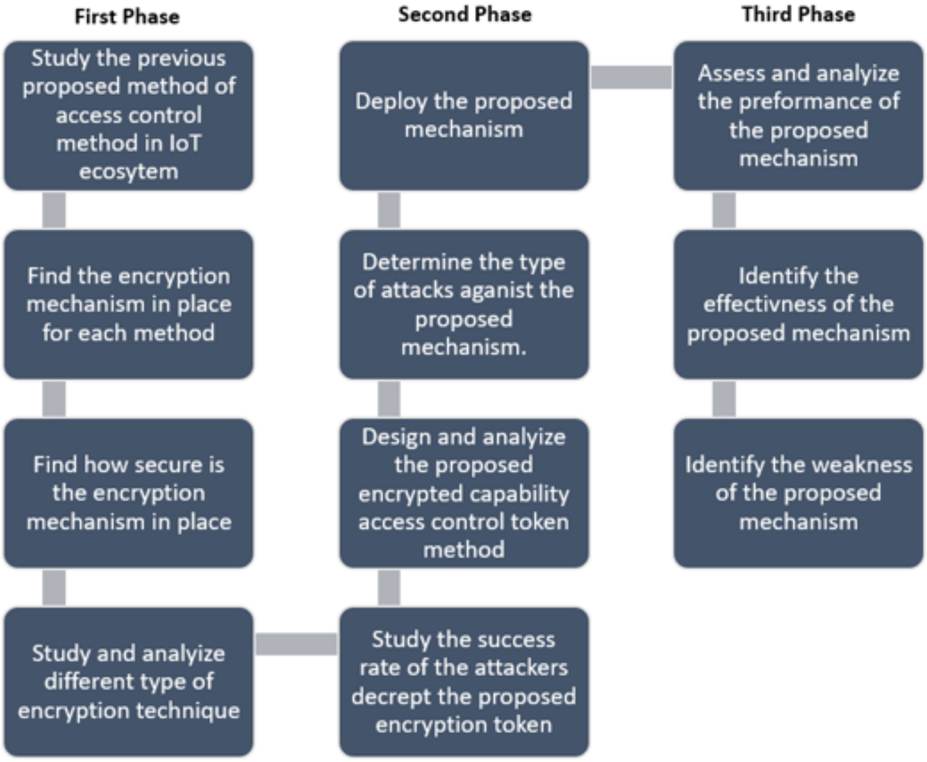
First Phase
Study the previous proposed method of access control method in IoT ecosytem
Find the encryption mechanism in place for each method
Find how secure is the encryption mechanism in place
Study and analyze different type of encryption technique
Second Phase
Deploy the proposed mechanism
Determine the type of attacks aganist the proposed mechanism.
Design and analyze the proposed encrypted capability access control token method
Study the success rate of the attackers decrept the proposed encryption token
Third Phase
Assess and analyize the preformance of the proposed mechanism
Identify the effectivness of the proposed mechanism
Identify the weakness of the proposed mechanism

Fig. 1. Research methodology

3.1 Comprehending and Analyzing Different Type of Access Control Methods

At the beginning of our research, we studied and compared different access control mechanisms used in IoT ecosystems. The most popular and most used solutions are Blockchain and smart contract since both these technologies are very reliable and trustworthy. Many of these mechanisms use different mechanisms but most noticeable and common mechanisms are attribute based access control (ABAC), rules-based access control (RBAC), and capability access control (CapBAC). The main different between RBAC and ABAC are the former assigns permissions based on roles, and the latter is attribute-based, which means granting access based on certain attributes that may change over time. As for (capBAC) it grants access based on the token that the subject has received form the owner of the object. The token contains the access right of the subject and what kind of access is granted i.e., (Read, Write, or Execute) For instance, if subject x wants to access a specific object the subject should have the token to access that object otherwise, he/she cannot access it. Our research focuses on the CapBAC method since it considered the most successful and popular solution.

ABAC has been developed to address the most critical government entities. ABAC has been used in numerous places, including the US Department of Defence (DoD), the UK Ministry of Defence (MoD), and quickly becomes a NIST standard. ABAC places the hashing and authentication mechanisms into the data itself, meaning that the hashing protects the data from being compromised and exploited outside of its intended user, even if stolen or flat-out hacked. ABAC uses different hashing method; most notable and common method is the usage of attribute-based hashing (ABE) Objects are hashed based on attribute-based access policies. ABE consists primarily of two types ABE (KP-ABE) and ciphertext-policy (CP-ABE). The entity is hashed in KP-ABE based on the collection of attributes belonging to an object, which must pass a policy encoded in the user's key to verify the object to continue. CP-ABE is the opposite of KP-ABE by using an attribute-based policy to encrypt an object and providing a user key consists of a collection of attributes relevant to that object User.

RBAC on the hands, is the second oldest form of access control method after ACL (access control list). As we explain earlier, RBAC is based on assigning rules to the employees so, the administrator will be determining the access privileges of each employee in the organization and whether the particular employee or set of employees need a modification right or just restricted it to read only. One of the benefits of RBAC is that its more affective compared to ABAC, the roles can be added or removed easily. And new roles can be added effortlessly. Moreover, the access control of sensitive information and data can be controlled effectively and lower the risk of data breach by applying the least privilege concept in this way, the organizations can avoid any kind of cyber-attacks. Nevertheless, one of the issues of RBAC is that it does not have any hashing mechanisms in place. So, if an intruders expose the role of certain employee, the intruders can gain access to all sensitive information. Furthermore, it’s pretty difficult for the admin to manage all type of roles that is needed for a particular team or project. Which ABAC has the edge over the RBAC thanks to its automated features.

CapBAC method is designed according to the Capacity Based Authorization Model (often time referred to as capability-based security). Capacity is a communicable, significant token of authority. It compares an attribute that uniquely refers to an object and a related array of access rights. The owner of the object creates the capability access token to be used by other users. The token's functionality is to Grant a particular set of users the right to interact and access the object in specific ways. However, one of the issues of capBAC is most of the time, the token is stored as a plaintext, which means the object ID and the access rights are exposed to attackers, and that can result in a data breach or even data modification. Nonetheless, capBAC is a new and promising access control method that many organizations and researchers start exploring and finding solutions to the limitation.

Our proposed improvement is on the proposed method of capability access control (capBAC). The system is implemented using Ethereum smart contract. The functionality of the system is that the owner of the object or the IoT device would first create an Ethereum Blockchain address we are going to call it *address A* and register a smart contract to be used to store and manage the capability access token. The owner afterwards creates a new capability access Token by executing a command in the smart contract. The newly created token will have a read, write, and execute access privilege to the address A since the smart contract does not know that address A is the actual object owner. Thus, the owner should first define himself as the object's owner/master before delegating the access tokens to other subjects. Therefore, the owner's capability access token states that *address A* has full privilege to the object or the IoT device. The owner can also delegate some of his privileges to other subjects. For instances, Address A can delegate read access token access right to another address or subject. Furthermore, the owner can revoke privilege from other users as well. Figure 2 illustrates the functionality of the system.

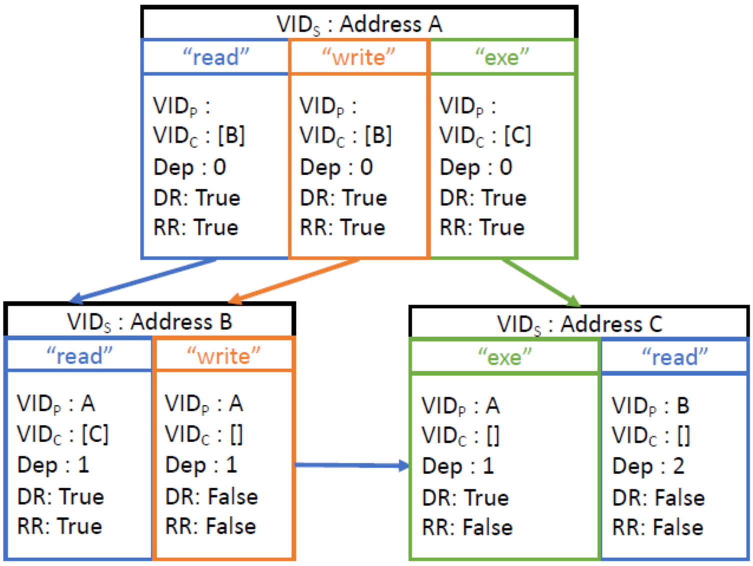

Fig. 2. Capability access control token (capBAC)

This is a simple example of access control token functionality. In the above figure, there are three addresses namely (*address A, address B, and address C*). just like we said earlier *address A* is the owner of the object or the IoT device and has three access tokens right (read, write, and exe). So, *Address A* each token has a certain information which are VID_p which indicates the ID of the patent or root of the subject, VID_c indicate the ID of the child address, *Dep* indicates whether the owner of the subject can delegate its token to other subject, while *RR* indicate whether the subject can revoke the delegated token. So, let us assume *address B* want a read and write access right, so address A (the parent) will delegate his write and read access token to *address B*. and similarly, *address B* can further delegate his write access token to *address C.* This approach makes the system very scalable and flexible. However, the proposed method's problem is that the access token is stored as a plaintext, which makes it insecure and intercepted by intruders.

3.2 The Proposed Method of Hashing the token in CapBAC scheme

In this phase, we explain the proposed method of hashing the access control token. We apt for hashing algorithm instead of hashing algorithm. Because of the issues associated with the hashing algorithms, there are two type of hashings methods namely symmetric hashing and asymmetric hashing. Symmetric hashing is the oldest and standard hashing method, it uses a single key to encrypt and verify the message, while asymmetric hashing uses two different keys one for hashing (Public key) and another for verifyion (Private Key). The problem with hashing is that There is no data protection along the way, any data that been hashed can be verifyed if the user applied a proper key mechanism. Which means the intruders can tamper with the content as well. An expert hackers can guess and discover the hashing mechanism and quickly figure out the verifyion method. Moreover, there are many tools and method hackers use to encrypt the messages. All the hacker needs to know is the hashing mechanism used in a particular system and the tool would do the hard work for him. So, in this proposed enhancement, we choose to utilize Keccak256 hash algorithm, which is the recommended hash algorithm in Ethereum. Keccak256 has been standardized for mobile telephony (TUAK) as well as the NIST standard. Keccak is based on a sponge construction approach, a comprehensive random permutation function that allows the input to be absorbed any amount of data, then output any amount of data. This result in significant security and flexibility. Figure 3 illustrates the flowchart of our proposed enhancement.

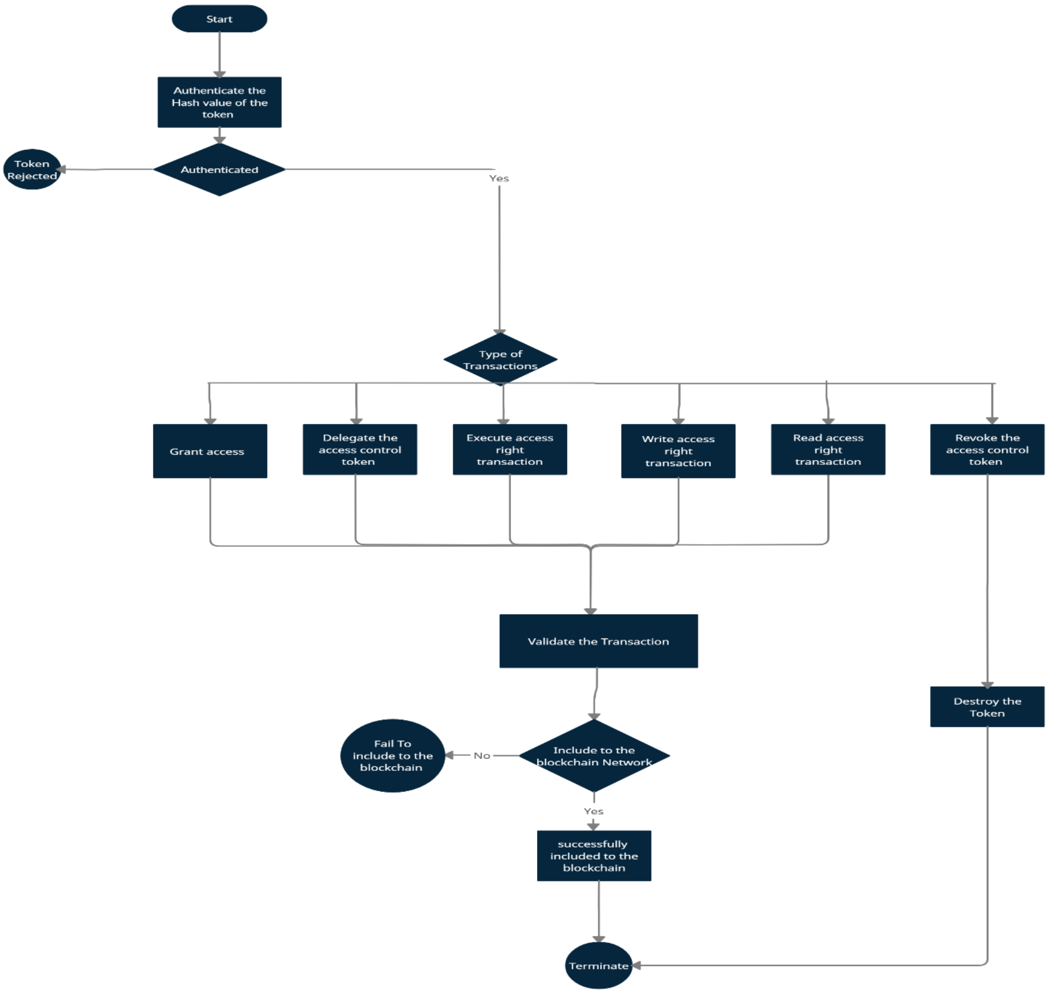
Start
Authenticate the Hash value of the token
Authenticated
Token Rejected
Yes
Type of Transactions
Grant access
Delegate the access control token
Execute access right transaction
Write access right transaction
Read access right transaction
Revoke the access control token
Validate the Transaction
Destroy the Token
Fail To include to the blockchain
No
Include to the blockchain Network
Yes
successfully included to the blockchain
Terminate

Fig. 3. The flowchart of the proposed enhancement

The functionality of the system does not differ from Figure 2 accept that we added a hash function to the token itself. So, we will reuse the example in Figure 2. The object owner creates Ethereum address called address A and creates a smart contract to store and manage the capability access token. However, in this case, the difference is that the owner will encrypt the subject ID, object ID, and the access right, e.g., write, read, or execute. So, it will be stored in the blockchain hashed. If the owner wants to delegate one of his capability tokens to another subject will name is address B, another smart contract will be triggered to add the new subject ID to the token afterwards hashed the ID again. Therefore, when address B want to interact with the object. It Has to present the capability access token. The system will validate it based on the hash value of the token. If the hash value matches the one stored in the blockchain, then address B will be permitted to utilize the object based on the access right the owner gave it. If the hash value differs, then the access request will be denied.

4. Results

This section is to analyse the performance and measurement of the proposed research. The analysis will be based on the previous capBAC proposed mechanism measurement method. In that method, capBAC measured the system based on gas consumption. Gas is referred to as the internal pricing of running a transaction. Each operation or transaction that preformed in Ethereum network cost a specified volume of gas. Thus, the more complex the transaction is, the higher the gas cost. The closest terminology we could use here is the cooking gas. Let say five people live in an apartment, and each one of them cooks their food three times a day. As a result, gas consumption will increase drastically. The same thing is applied in the Ethereum. If the transaction is very complex, it will consume more gas, which will cost more tokens/money. However, there is one thing we need to notice. The transactions are paid on gas, not token, and that is because, if we use Ether cryptocurrency, the system might not be usable or stable, since Ether's price fluctuates based on the currency market value. So, Ethereum will have to update the pricing every second. Because, of that, Ethereum uses gas consumption per transaction instead of Ether or other cryptocurrencies. So, Table 1 to Table 2 show the gas consumption of capBAC.

Table 1
Create action gas consumption

Len(Op[])	Act	Gas
0	“read”	64,435
1	“edit”	51,774
2	“POST”	54,110
3	“exe2”	56,446

In Table 1 we can see that each action/token created consume different amount of gas, for instance, for the first action, *read* consumes 64,435 while *exe2* consumes 56,446 which is less than the first action. That is because, the smaller the task the less it consumes. Unless the *exe2* action contains complex computational power. Which in this case is unluckily. Table 2 shows the delegation gas consumption.

We have five addresses (users) namely *address A, address B, and address C address D, address E and lastly address F.* So, we assume that address *A* is the owner of the object. Thus, *address A* delegates the access token to *address B,* and that consumes 162,386 gas. And *address B* further delegates the access token to *address C* and so on. If we look closely, we notice that when address C delegate the access token to address E, it consumes only 147,386 gas. The 15,000-gas difference is because every time an address/ user delegates the access control, the smart contract creates a VID_{ch} storage to store all the delegated tokens, which cost more gas.

Table 2
Delegation action gas consumption

Delegator	Delegatee	Gas
addressA	addressB	162,386
addressB	addressC	162,386
addressC	addressD	162,386
addressC	addressE	147,386
addressC	addressF	147,386

Therefore, in our proposed enhancement, we expect to get the same gas consumption result as the above figures and if we get lower that would be significant and a milestone.

5. Conclusions

The Internet of Things is the idea of connecting different types of devices (things) to the internet. These connected devices can intercommunicate among each other to transfer or receive data. IoT has become an essential piece of technology in our daily life. Many researchers believe that IoT is the future and will never vanish from our life. Instead, we will depend on them, even more, to make our task more manageable and accomplishable. However, the primary concern of IoT technology is its fragile security mechanism. Many IoT manufacturers do not spend a lot of time on securing the IoT devices. Many IoT devices are shipped out with a very loose security solution. i.e., default password. Therefore, in this study, we emphasis on the integrity of the access control token. We explained the benefit and the purpose behind using tokenization as an authentication method. We also discussed the different mechanisms in place for IoT technology. And lastly, we discussed the method we used for conducting this research as well as the introduction of our proposed hashing algorithm. Furthermore, we also discussed the method of measuring and evaluating the efficiency of our algorithm.

Acknowledgement
This research was supported by Universiti Putra Malaysia (UPM). Special thanks from authors for financial support from Universiti Putra Malaysia.

References

[1] Ouaddah, Aafaf, Anas Abou Elkalam, and Abdellah Ait Ouahman. "FairAccess: a new Blockchain-based access control framework for the Internet of Things." *Security and communication networks* 9, no. 18 (2016): 5943-5964.

[2] Zhang, Yan, Bing Li, Ben Liu, Jiaxin Wu, Yazhou Wang, and Xia Yang. "An attribute-based collaborative access control scheme using blockchain for IoT devices." *Electronics* 9, no. 2 (2020): 285.

[3] Alphand, Olivier, Michele Amoretti, Timothy Claeys, Simone Dall'Asta, Andrzej Duda, Gianluigi Ferrari, Franck Rousseau, Bernard Tourancheau, Luca Veltri, and Francesco Zanichelli. "IoTChain: A blockchain security architecture for the Internet of Things." In *2018 IEEE wireless communications and networking conference (WCNC)*, pp. 1-6. IEEE, 2018.

[4] Zhang, Yuanyu, Shoji Kasahara, Yulong Shen, Xiaohong Jiang, and Jianxiong Wan. "Smart contract-based access control for the internet of things." *IEEE Internet of Things Journal* 6, no. 2 (2018): 1594-1605.

[5] Xu, Ronghua, Yu Chen, Erik Blasch, and Genshe Chen. "Blendcac: A smart contract enabled decentralized capability-based access control mechanism for the iot." *Computers* 7, no. 3 (2018): 39.

[6] Fotiou, Nikos, Iakovos Pittaras, Vasilios A. Siris, Spyros Voulgaris, and George C. Polyzos. "Secure IoT access at scale using blockchains and smart contracts." In *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pp. 1-6. IEEE, 2019.

[7] Di Francesco Maesa, Damiano, Paolo Mori, and Laura Ricci. "Blockchain based access control." In *IFIP international conference on distributed applications and interoperable systems*, pp. 206-220. Cham: Springer International Publishing, 2017.

[8] Patil, Akash Suresh, Bayu Adhi Tama, Youngho Park, and Kyung-Hyune Rhee. "A framework for blockchain based secure smart green house farming." In *International Conference on Ubiquitous Information Technologies and Applications*, pp. 1162-1167. Singapore: Springer Singapore, 2017.

[9] Novo, Oscar. "Blockchain meets IoT: An architecture for scalable access management in IoT." *IEEE internet of things journal* 5, no. 2 (2018): 1184-1195.

[10] Ren, Yongjun, Fujian Zhu, Jian Qi, Jin Wang, and Arun Kumar Sangaiah. "Identity management and access control based on blockchain under edge computing for the industrial internet of things." *Applied Sciences* 9, no. 10 (2019): 2058.

[11] Abdi, Adam Ibrahim, Fathy Elbouraey Eassa, Kamal Jambi, Khalid Almarhabi, and Abdullah Saad Al-Malaise Al-Ghamdi. "Blockchain platforms and access control classification for IoT systems." *Symmetry* 12, no. 10 (2020): 1663.

[12] Anggorojati, Bayu, Parikshit Narendra Mahalle, Neeli Rashmi Prasad, and Ramjee Prasad. "Capability-based access control delegation model on the federated IoT network." In *The 15th International Symposium on Wireless Personal Multimedia Communications*, pp. 604-608. IEEE, 2012.

[13] Deebak, Bakkiam David, and Fadi Al-Turjman. "A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks." *Ad Hoc Networks* 97 (2020): 102022.

[14] Xue, Jingting, Chunxiang Xu, and Yuan Zhang. "Private blockchain-based secure access control for smart home systems." *KSII Transactions on Internet and Information Systems (TIIS)* 12, no. 12 (2018): 6057-6078.

[15] Nakamura, Yuta, Yuanyu Zhang, Masahiro Sasabe, and Shoji Kasahara. "Exploiting smart contracts for capability-based access control in the internet of things." *Sensors* 20, no. 6 (2020): 1793.

[16] Sultana, Tanzeela, Ahmad Almogren, Mariam Akbar, Mansour Zuair, Ibrar Ullah, and Nadeem Javaid. "Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices." *Applied Sciences* 10, no. 2 (2020): 488.

[17] Almiani, Muder, Alia AbuGhazleh, Amer Al-Rahayfeh, Saleh Atiewi, and Abdul Razaque. "Deep recurrent neural network for IoT intrusion detection system." *Simulation Modelling Practice and Theory* 101 (2020): 102031.

[18] Outchakoucht, Aissam, ES-SAMAALI Hamza, and Jean Philippe Leroy. "Dynamic access control policy based on blockchain and machine learning for the internet of things." *International journal of advanced Computer Science and applications* 8, no. 7 (2017).

[19] Alam, Tanweer. "Internet of things: A secure cloud-based manet mobility model." *Tanweer Alam." Internet of Things: A Secure Cloud-Based MANET Mobility Model.", International Journal of Network Security* 22, no. 3 (2020).

[20] Awaisi, Kamran Sattar, Shahid Hussain, Mansoor Ahmed, Arif Ali Khan, and Ghufran Ahmed. "Leveraging IoT and fog computing in healthcare systems." *IEEE Internet of Things Magazine* 3, no. 2 (2020): 52-56.

[21] Alam, Masoom, Naina Emmanuel, Tanveer Khan, Yang Xiang, and Houcine Hassan. "Garbled role-based access control in the cloud." *Journal of Ambient Intelligence and Humanized Computing* 9, no. 4 (2018): 1153-1166.

[22] Bokefode, Jayant D., Avdhut S. Bhise, Prajakta A. Satarkar, and Dattatray G. Modani. "Developing a secure cloud storage system for storing IoT data by applying role based encryption." *Procedia Computer Science* 89 (2016): 43-50.

[23] Ghosh, Smarajit, and Vinod Karar. "Blowfish hybridized weighted attribute-based encryption for secure and efficient data collaboration in cloud computing." *Applied Sciences* 8, no. 7 (2018): 1119.

[24] Ramu, Gandikota, and Appawala Jayanthi. "Enhancing medical data security in the cloud using RBAC-CPABE and ASS." *International Journal of Applied Engineering Research* 13, no. 7 (2018): 5190-5196.

[25] Saraswathi, M., and T. Bhuvaneswari. "A Secured Storage using AES Algorithm and Role Based Access in Cloud." *Int. J. Sci. Res. Sci. Eng. Technol* 3, no. 5 (2017): 511-515.

[26] Tian, Ye, Yanbin Peng, Gaimei Gao, and Xinguang Peng. "Role-based Access Control for Body Area Networks Using Attribute-based Encryption in Cloud Storage." *Int. J. Netw. Secur.* 19, no. 5 (2017): 720-726.

[27] Staff Contributor. 2019. "RBAC vs. ABAC Access Control: What’s the Difference? - DNSstuff." Software Reviews, Opinions, and Tips - DNSstuff. October 31, 2019.

[28] "Understand Ethereum Blockchain — Steemit." 2018. Steemit.com. Steemit. 2018.