Decentralised Access Control Framework using Blockchain: Smart Farming Case

Normaizeerah Mohd Noor[1], Noor Afiza Mat Razali[2]*, Sharifah Nabila S Azli Sham[3], Khairul Khalil Ishak[4], Muslihah Wook[5], Nor Asiakin Hasbullah[6]

Faculty of Defence Science and Technology, National Defence University of Malaysia, Kuala Lumpur, Malaysia[1, 2, 3, 5, 6]
Center of Cyber Security and Big Data, Management and Science University, Selangor, Malaysia[4]

***Abstract*—The convergence of farming with cutting-edge technologies, like the Internet of Things (IoT), has led to the emergence of a smart farming revolution. IoT facilitates the interconnection of numerous devices across different agricultural ecosystems, enabling automation and ultimately enhancing the efficiency and quality of production. However, the implementation of IoT entails an array of potential risks. The accelerated adoption of IoT in the domain of smart farming has amplified the existing cybersecurity concerns, specifically those pertaining to access control. In extensive IoT environments that require scalability, the conventional centralized access control system is insufficient. Therefore, to address these gaps, we propose a novel decentralized access control framework. The framework applies blockchain technology as the decentralization approach with smart contract application focuses on the application scenario in smart farming to protect and secure IoT devices from unauthorised access by anomalous entities. The proposed framework adopted attribute-based access control (ABAC) and role-based access control (RBAC) to establish access rules and access permissions for IoT. The framework is validated via simulation to determine the price of gas consumption when executing smart contracts to retrieve attributes, roles and access rules between three smart contracts and provide the baseline value for future research references. Thus, this paper offers valuable insight into ongoing research on decentralized access control for IoT security to protect and secure IoT resources in the smart farming environment.**

Keywords*—Access control; role-based access control; attribute-based access control; blockchain technology; internet of things; smart contract; smart farming

I. Introduction

The integration of the Internet of Things (IoT) technology into smart farming infrastructure has the potential to revolutionize the agricultural industry by enabling the collection and analysis of vast amounts of data from various sources such as sensors, drones, and cameras. IoT technology can provide real-time information on farm operations, allowing farmers to remotely monitor and control equipment and make data-driven decisions for fast response to issues, minimize impact and reduce costs [1]. However, the adoption of IoT devices in smart farming also presents several challenges that need to be addressed. One of the primary concerns is the risk of IoT security, which arises due to the use of numerous heterogeneous devices in the system. Another critical issue is the management of resources, which can become complex and require a high level of coordination and integration. Furthermore, as smart farming systems grow in size and complexity, scalability becomes an increasingly important factor that must be considered[2],[3],[4]. Thus, to address the challenges posed by the adoption of IoT devices in smart farming, it is crucially needed for the enhancement of access control to ensure authorized access will be granted to legitimate devices while also being scalable to accommodate future expansion. An effective access control system can help mitigate the risks associated with IoT security and improve the overall scalability and management of resources in smart farming systems. Nevertheless, conventional centralised access control has brought about several problems and remains as a complicated issue since it includes single point of failure and incapability of addressing dynamic and diverse access control requirements for future IoT ecosystems [5] [6]. Therefore, the new framework must be designed with the aim of shifting from a centralised approach to a decentralised approach for eliminating trusted third parties in access control and achieving optimum management of IoT resources. Thus, this paper proposes a decentralisation approach using blockchain technology as a suitable solution since it provides an open, transparent and distributed ledger without the need for a third party [7]. It also has strong security features for securing IoT resources in the form of hashing ledger which guarantees high system reliability and integrity. This paper is structured as follows. Section II describes the background study including the IoT infrastructure, security issues, access control, blockchain smart contract and its application in smart farming. Section III highlights the related works to this study. Section IV discussed the proposed decentralised access control framework for IoT security enhancement using blockchain technology. Section V describes the evaluation procedure. Section VI presents the contribution for this work and Section VII discusses the conclusion for this study.

II. Background Study

According to the United Nations (UN), population growth is steadily increasing along with food consumption and production demands, which are anticipated to increase up to 70% by 2050 [10]. To fulfil these demands, conventional agriculture must shift to smart farming which combines internet connection and modern technology like IoT. This will provide numerous benefits, including accurate data collection for data-assisted decision-making [11], [12]. Such a scenario will enable remote monitoring, thereby contributing to the reduction of production costs. This will lead to efficient and sustainable agricultural production that is more demand-oriented and resource-efficient.

*Corresponding Author: FRGS/1/2021/ICT07/UPNM/02/1

A. IoT Architecture and Security Issues in Smart Farming

In smart farming, the integration of IoT sensors with any farm equipment and machinery for monitoring temperature, humidity, pressure, etc., will enable systematic data collection. The data can be remotely sent from different locations to a centre for monitoring and decision-making. These devices and sensors have their roles to play according to the different techniques used, their functionality and implementation, which can help farmers provide information in real-time. Farming techniques can be improved based on the collected information [13]. For instance, the roles include crop management, water management, soil management, livestock management, smart greenhouses and agriculture drones[11]. Smart irrigation systems, for instance, use temperature and soil sensors to maintain and control water wastage as well as to improve crop quality by monitoring the humidity of the soil and only watering at the right time. Thus, the management of heterogeneous IoT devices and sensors must be efficient and reliable.

The IoT architecture illustrated in Fig. 1 displays the key layers in smart farming, which are: the physical layer, the network layer, the edge or fog layer and the application layer. The physical layer can be any type of device (such as actuators and sensors) connected to the IoT network. The network layer is responsible for data transmission from the physical layer to the data processing system. The data transmission may use any wired or wireless device, such as a router, access points, 4G or 5G network, Wi-Fi, Bluetooth, etc. The network layer has a high possibility of security flaws if there is connectivity via the internet. The probable attacks (such as identity theft, bullying or controlling/hacking) can be countered by implementing identity management and encryption schemes [14]. The next layer is the edge or fog layer consisting of various resources with computer processing capabilities. This layer can store a small amount of data and process that data. It can also be used for decision-making and security features. The edge or fog layer includes the in-out interface and the gateway used to manage the entire collected data from the sensor without transmitting it to the cloud. The application layer is the communication protocol and interface that provide services to users and data visualisation from the sensor network.

It is important to protect and safeguard connected devices in the IoT environment [2]. According to [14], the security protocols that should be applied in smart farming IoT security solutions are access control, authentication, firewall, anomaly detection system and cryptography. However, before applying those security protocols, we must address the security issues and potential attacks in each layer of smart farming. Study [15] has developed various security protocols and arranged them into different categories (access control protocols, authentication protocols, key management protocols and intrusion detection protocols) to support various IoT applications that suffer from possible attacks.

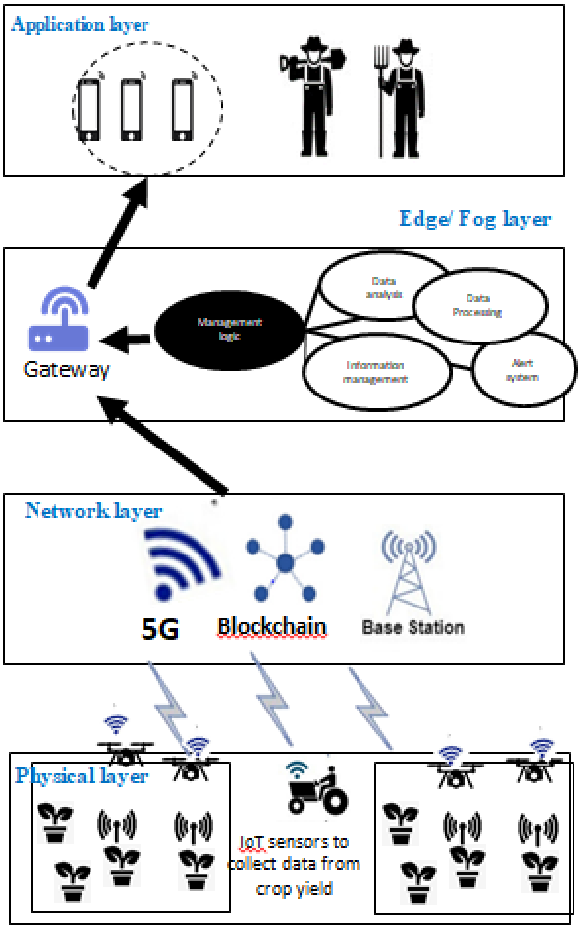


Fig. 1. Smart farming architecture.

Access control must be implemented to facilitate the process of data transfer in the physical layer. However, in IoT environment, security protocols (such as secure public key-based authentication and cryptography) are not suitable due to high computational power and storage capacity requirements [4]. Gupta et al. (2020) stated that edge layers may contain major security issues due to IoT devices and sensors that do not have their own security. This makes it easy for attackers to gain remote access to the system via unauthorised access, booting, flooding and signature wrapping. In the application layer, attacks have been categorised into two types: software attacks and encryption-based attacks. Software attacks generally use malicious software agents to acquire authentication credentials of users [14]. Encryption-based attacks apply extensive attacks to exploit the cryptographic protocols and mathematical models. Table I provides a summary of possible security attacks and issues in smart farming, along with the proposed countermeasure. In Table I, the type of attack is categorised according to the layers in the smart farming architecture.

TABLE I. SECURITY ATTACKS AND COUNTERMEASURES ACCORDING TO LAYERS

Layer	Security goals	Security attacks	Countermeasure
Application	Availability, Non-Repudiation, Privacy [17], Accountability and Integrity [2]	Data thefts, Sniffing, Access Control [4], [14] [17], Phishing attack, Malicious scripts, Deny services [4], reprogram attacks [17], Channel interference, DoS/DDoS, Cyberagroterrorism [2], Malicious Code Injection Attack, tempering privacy [14]	Access control, data encryption (cryptography and non-liner key encryption), Authentication, anti-virus, anti-spyware, firewall and ACLs [14]
Edge/Fog	Integrity, Authenticity, Confidentiality [17], [2]	Man-in-the-middle, Booting vulnerabilities, Unauthorised access, Signature wrapping, Forged control for actuators, Gateway-cloud request forgery, Forged measure injection [4], Flooding [16], cloud malware injection, SQL injection, Storage attacks, Side-channel attacks, Sybil Impersonation, Replay Session Hijacking [2], Interception of node communication [7]	Authentication, IDS, Anomaly detection system, access control [4]
Network	Availability [2], Confidentiality [14]	DoS/DDoS, Data transit attacks, Routing attacks, Autonomous system disruption, Signal disruptions [4], wormhole attack, traffic attack, jamming attack [18]	Identity management, encryption schemes, data privacy, authentication, hello flood detection, routing protocol
Physical	Confidentiality [2]	Random sensor incidents, Autonomous system hijacking, optical deformation, Irregular measurement, Sensor weakening, Node capture, Fake node, Sleep deprivation [4], social engineering, jamming attack [18], eavesdropping, malicious code injection [18], Facility damage [7]	Data privacy, secure booting, data integrity, risk assessment, device authentication, secure physical design

B. *Access Control in IoT*

The basic element of access control is the ability of the subject and object to perform an action that includes interaction in the right manner [19]. In the IoT environment, access control plays a crucial role in ensuring that all resources, including actuators and devices, are protected using selective restrictions that control access to IoT devices [20]. The object is defined as the system resource that contains or stores information on IoT devices, sensors, directories, programs, etc. An object is secured by a set of access policies consisting of conditions and requirements for an object's access to be granted. The subject can be defined as an entity (users or systems) that is capable of accessing an object. A subject must prove that it satisfies an access policy of a requested object before access is granted.

According to [21], four design components must be addressed based on the current access control problem in the IoT environment. Meanwhile, several approaches have been proposed for managing access control and associated privilege according to their access level in IoT systems [22], [23], [24], [25],[26],[27]. Discretionary Access Control (DAC), Mandatory Access Control (MAC), ABAC and RBAC are the most conventional models used in smart farming. Based on the literature, two commonly employed access control mechanisms for IoT are RBAC and ABAC due to their strong features and flexibility in supporting the IoT environment [28], [29], [30], [31], [32], [33], [34], [35].

1) Basic concepts, advantages and disadvantages of the ABAC model: ABAC uses pre-defined policies for access permission. The policies consist of three attributes: subject, object and environment [8], [30]. The attributes are used to authorise access permission with specified access policies using a target function that determines whether or not sufficient privileges are present for access[36]. Specific access policies with selected attributes must have good management [32].

The advantages of ABAC include the flexibility of policies based on changing dynamic attributes, such as location and time. With its flexibility and scalability, the ABAC model is more suitable for access control in IoT [32], [34], [36], [37], [38]. In addition, the use of access control marker language (XACML) as an extension of ABAC can be expressed as logical-based policies to define valid authorised access [21]. However, the drawback of XACML is the extensible markup language (XML), which makes it unsuitable for constrained devices, such as IoT applications. In ABAC, all attributes that have been defined must be managed and distributed to the right user for effective access management [32]. It can be a problem for IoT devices with less storage and computing power when the number of attributes and the number of users increase.

2) Basic concepts, advantages and disadvantages of the RBAC model: In RBAC, access control is based on the roles of subjects within an organisation who give permission. By associating the user with its roles and access permissions (e.g., read, write and execute), the roles are set to be active. They can be structured in hierarchal order where senior roles are more powerful and have more permission for access as compared to junior roles. Another important aspect of RBAC is constrained enforcement. A constraint can be applied at either the system level or the application level. Restrictions to RBAC states with or without being event triggered are known as invariant and precondition. These two restrictions are used as conditions when a role is assigned to a user in a user-role assignment and permission is assigned to a role in a permission-role assignment.

The advantages of RBAC are: a) the user can access resources based on the achieved tasks under suitable access mode and b) it is easier for the system administrator to redefine permissions for each user separately according to their roles [8]. The disadvantage of the RBAC mechanism is the inability to differentiate its role [28], leading to role-permission explosion problems in situations where the service-providing entities are unable to allow access permission to the user-role assignments of the role-providing entities due to a large number of objects [8]. Research [39] stated that service-providing entities must use an alternative to confirm if an unknown guest legitimately owns a certain role. The authors in [30] also noted that the disadvantages of the RBAC system are its lack of flexibility in adapting to changing users, maintain user-to-role assignment and role-to-permission assignments for dynamic applications or large-scale applications with a significant number of users or objects.

In summary, IoT has various limitations, including resource constraints, that prevent IoT from handling operations that require high computational power including managing complex access control [40], [41]. In [42], the authors use a combination of RBAC and ABAC models in the centralised environment. The authors proposed to divide the permissions assigned to a role according to their access actions. However, most research had proposed centralised decisions which can lead to the central point of failure and limited resources of IoT devices [43],[40]. Thus, the decentralised approach is more suitable for large-scale IoT environments.

C. Blockchain and Smart Contracts

Blockchain is formally described as a digital, decentralised and distributed ledger that communicates transactions or sensitive data without trusted third parties, removing centralised authority and intermediaries, and enabling two parties to communicate and conduct business quickly, securely and reliably [44]. This technology is different from the traditional system where the conventional approach is centralised. The structure of chain in blockchain is shown in Fig. 2.

In contrast, the blockchain system implements a decentralised system with many possible physically scattered nodes [45]. Blockchain also has strong security features for securing IoT resources in the form of hashing ledger which guarantees high system trustworthiness and integrity [32]. Based on the literature, blockchain has various unique characteristics such as decentralisation, transparency, autonomy, security, immutability, traceability, integrity and programmability [46]. Due to blockchain’s characteristics, its application is relevant for access control in smart farming since complex approaches are required. In the meantime, for a successful transaction on a blockchain network, verification is required through a consensus algorithm to reach an agreement on the transaction or a smart contract between two parties. The adoption of a consensus mechanism is dependent on the types of networks and the roles of nodes. Blockchain networks can be public or private networks [47], and the roles can be permissionless and permissioned. In permissioned or private networks, only invited nodes can participate in the network. The nodes will be divided and assigned to their roles. Only the selector miner node can perform transactions [48].

Meanwhile, smart contracts are self-executing contracts in which the terms of an arrangement between two parties are expressed in computer codes. When the requirements of a smart contract are met, it will self-execute to a blockchain, removing the need for trusted third parties [49]. According to [50], smart contracts are one solution that responds to the transaction sent by a user. The transactions use code logic which is the Solidity language [51]. Once users agree to the agreement based on the contract, this code logic will be incorporated into the blockchain network and all users in the network will have copies of the contract.

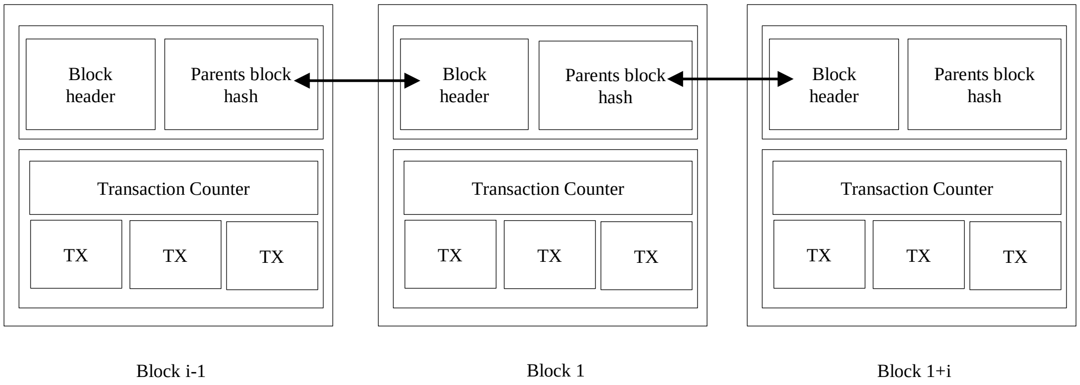


Fig. 2. The chain in blockchain.

III. Related Works

This section presents the most relevant literature related to our study and proposes a solution for access control in the IoT environment. Table II displays several existing solutions with a list of required elements for developing access control solutions in the IoT environment. The researchers in [52] proposed RBAC as a strategy for device management by considering the limited user access strategy and protecting the IoT network using Software-Defined Networking (SDN). The proposed study was able to manage network flow and provide dynamic access control when the access network rule needed to change. This proposal can overcome the problem of unauthorised device in the system by authenticating the server and providing information, such as a network device and network identification, to detect malicious activity. However, this proposal requires a system administrator to carefully manage user access control and network rule exposed to the risk of vulnerabilities due to careless configuration. The study [53] proposed a framework by using event-based solutions for access control mechanisms in an IoT environment. To handle the event process, the authors employed a processing module as a policy module for managing and controlling the movement of event operators and calculating data to prevent starvation of resources during the computation process.

Researchers also suggested combining the access control model and blockchain since blockchain has been widely used in several domains for promoting decentralisation, dynamic access control and tamper-proof [54]. ABAC was proposed to be the access control model with advantages such as scalability and flexibility for securing and protecting IoT resources with excellent features. The research [55] suggested a Policy chain integrated blockchain-based ABAC framework to address the problem of securing shared resources in decentralisation. In this framework, the authors utilised the JSON + Script format as the policy expression and devised new ways to apply policies using “script interpreter”. The interpreter was constructed according to three evaluations: evalScript, evalRule and evalPolicy as well as a consensus protocol for executing decisions faster. The research proposes that off-chain resources can be accessed by IoT devices using the pre-determined on-chain policy. The authors used consortium blockchain and two different nodes: full nodes and lightweight nodes. It was noted that the consensus algorithm can be used for validating and storing transactions in full nodes. The synchronisation of current state networks was accomplished in light nodes.

TABLE II. Comparison of Existing Solution Access Control in IoT Environment

Product	Secure policies	Trusted authorisation	Secure communication	Flexibility	Security	Scalability	Decentralised (D) / Centralised (C) / Partial (P)	Access control model	Domain
[34]	/	/	X	/	/	/	D	ABAC	IoT
[56]	/	/	X	/	/	/	D	ABAC	IoT
[52]	X	/	X	/	/	X	C	RBAC	IoT
[21]	X	X	X	/	/	X	C	ABAC	IoT
[60]	X	/	/	/	/	/	P	ABAC, RBAC, CBAC	Health-care (IoT)
[55]	X	/	X	/	/	/	P	ABAC/ XACML	IoT-ICS
[61]	/	/	X	/	/	/	D	CapBAC, ABAC	IoT
[62]	/	/	X	/	/	X	D	N/A	IoT
[58]	/	/	X	/	X	/	D	ABAC	Data sharing in IoT
[59]	N/A	/	X	X	/	/	D	N/A	IoT
[8]	X	X	X	/	/	/	C	ABAC, RBAC	Multi-domain
[53]	X	/	X	/	/	N/A	C	EBAC	IoT

Similarly, [56] proposed an access control framework based on the blockchain technology suitable for heterogenous IoT by evaluating attributes, operations and environments according to requests. In this proposal, the researchers used four smart contracts for executing access control mechanisms: access control contract (ACC), subject contract (SC), object contract (OC) and multiple policy contracts (PCs). They ensure security and flexible access control in the IoT environment. The authors also utilised trust management for detection and evaluation of malicious behaviour from other devices. The authors [34] stated that the protection of critical resources in IoT can be done by replacing conventional centralised access control, which is insufficient in large-scale IoT environments. They suggested the Attribute-Based Distributed Access Control (ADAC) with a smart contract system. ADAC was proposed to manage and access attributes of IoT devices by using three smart contracts: ACC, OC and multiple PC. ADAC development was inspired by the ABAC model which can determine authorised users based on subject attribute, object attribute, environment attribute and policies. The study [57] also offered to solve the single point of failure issue by combining Accountable Subgroup Multi-Signature (ASM) algorithm with the ABAC model and smart contract policy in order to achieve fine-grained and reliable data access control. This paper uses access policies to specify whether users with certain subject attributes are permitted to perform certain actions on data with certain object attributes in a certain environment. The access policies consist of Subject Attributes Policy (SAP), Object Attribute Policy (OAP), Attributes Authorise Policy (AAP), Environment Attributes Policy (EAP) and Result. For policies, evaluation is based on the required attributes that meet with policy, and the result consists of three elements: permit, deny and not applicable. ABAC model for IoT-integrated blockchain technology to tamper-proof, store the attribute and eliminate a single point of failure were utilised in a study. For accessing data, the author applied four smart contract mechanisms implemented on the Ethereum blockchain: ACC, object attribute management contract (OAMC), subject attribute management contract (SAMC) and policy management contract (PMC). They are responsible for storing and managing access policy information that consists of specified actions regarding the subject and object which must have their access request verified. However, the proposed framework lacks security and privacy protection of IoT data due to unauthenticated edge nodes which have no access decision at the edge. The researchers in [59] proposed the BorderChain application which allows IoT owners to authorise selective IoT services and devices that permit access at the IoT gateway before opening the endpoint to others via smart contracts. After the IoT owner grants access, an access token will be generated which can be used by legitimate IoT services and users to query IoT resources in IoT domains. This solution can convince IoT domain owners that the system will only authorise IoT requests that they approve. For scalability goals, the authors implemented off-chain (outside blockchain) which is cheaper and more efficient during the process of signature verification mechanism. The study [9] combined elements of access control methods, such as ABAC, RBAC and Capability-Based Access Control (CBAC), to establish fine-grained policy decisions in the healthcare environment. This framework reduces the number of policies by using the attribute to define roles as well as capabilities to provide only single attribute expressions that can access multiple resources. However, this framework is partially decentralised and stores access policy in a single database server based on a policy language (XACML) as well as policies generated by administrators. The blockchain will only allow if it reaches an agreement in the smart contract/consensus algorithm. It was also noticed that the development of security policies in access control mechanisms can be achieved via smart contracts where all users in the blockchain network will acquire a copy of policies and store them in blockchain. Flexibility and scalability can be achieved when using the combined access control model in an IoT environment since it can be utilised in heterogeneous IoT devices, further reducing the use of storage capacity for storing access policies in IoT devices.

Based on the literature, we identified that there is significant advantage for decentralised access control with blockchain technology integrated with the RBAC and ABAC models. RBAC can provide strong security by conducting role hierarchy and constraints to give permission, whereas ABAC is very flexible in granting access permission based on the three attributes. Therefore, in this paper, for our framework development, we propose the use of blockchain technology as a decentralised solution for managing and storing access policy information of subject and object that must verify their access request. We also utilise smart contracts for the automation of access decisions. For access policy development, we propose to implement a combination of the RBAC and ABAC models as an access control strategy. Our proposed framework is aiming to close the gaps for the access control focusing on enhancement of security and resource management using decentralized IoT mechanism that also considers the scalability factor.

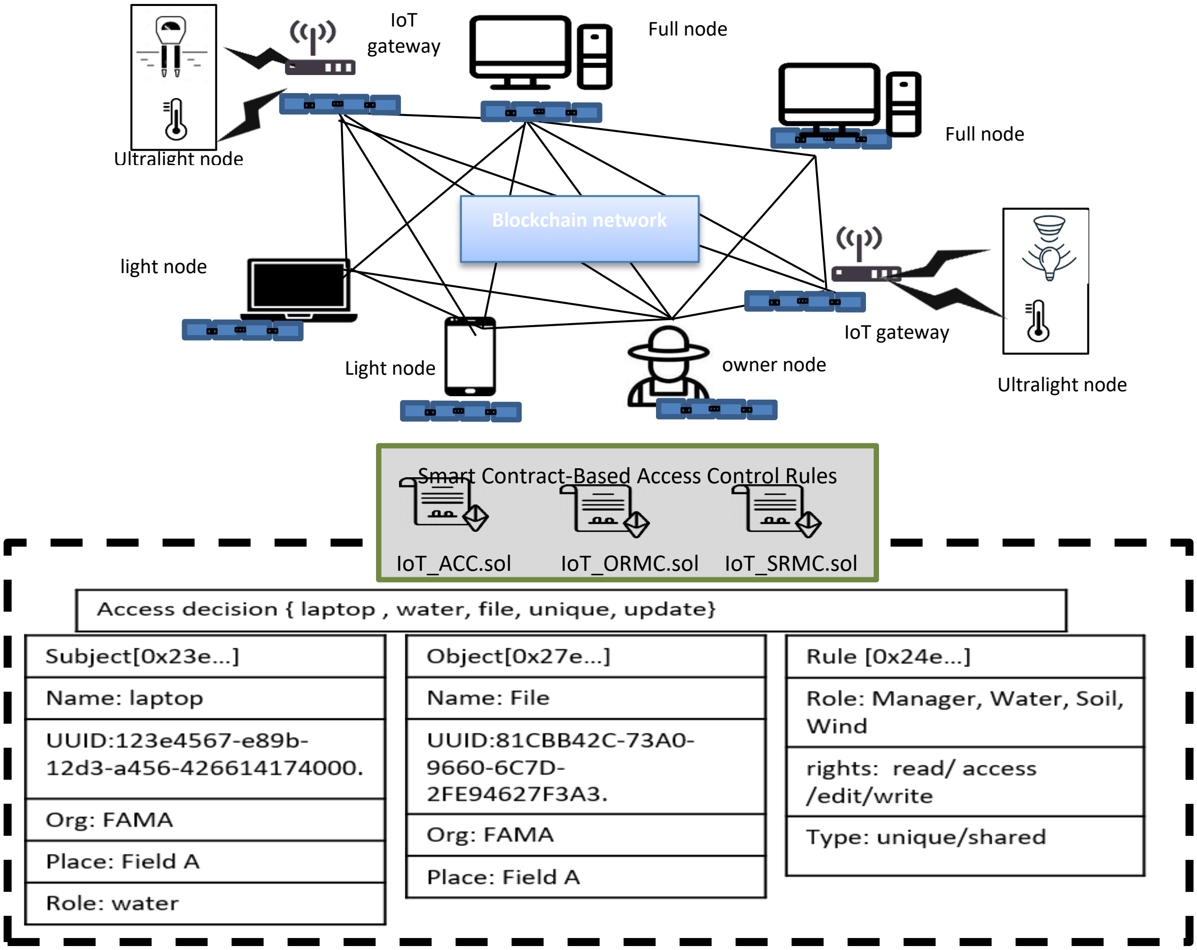
IV. Decentralized Access Control Framework for IoT Security Enhancement using Blockchain Technology in Smart Farming

This section discusses our proposed decentralised access control framework for IoT security enhancement using blockchain technology. First, we present an extensive overview of our proposed framework, as illustrated in Fig. 3. This framework was developed with the primary aim of achieving security, while also efficiently managing resources and ensuring scalability to cope with the increasing demands of smart farming. This framework was developed by adapting the FRABAC model where the combination of RBAC and ABAC models with user-role permission and attributes are employed for the user, admin and resource owner through smart contracts [8]. The integration of blockchain and access control models is the novel element that can reduce the redundancy of several roles and rules of permission. It has a unique access without creating or implementing special roles or rules reserved for each user/device. This framework can help address the role permission explosion or role-explosion problems, which have complex role structure (hierarchy) and a large number of roles. Most of them have the same access permissions. Our framework includes a blockchain-based smart contract and P-2-P network. The network consists of

IoT node owner, full nodes, lightweight nodes and extra lightweight nodes which have their own responsibilities based on the ability to execute access control according to computing power and storage capacity that considered based on smart farming scenario.

In this framework, we propose the adoption of smart contracts for access permission request, access control rule management and for verifying the permitted decision by fulfilling the requirement of access rules.

Smart contract is also responsible for updating attributes and roles. Access permission provides transparent access permission and traceability since all nodes have a copy of the smart contract. In FRABAC model there are Access Control Contract, Object-Rule Management Contract and Subject-Role Management Contract. These three concepts were adopted as IoT_ACC, IoT_ORMC and IoT_SRMC in our framework. In the smart farming environment, to address heterogeneous IoT device authorisation matters, we propose that every device must authenticate itself by describing and identifying its own credentials including its attributes, such as address name, identification number, location and role. Thus, all authenticated devices must interact through smart contracts for access control execution which contributes to tamper-proof access rules. A set of rules was developed to define access permission that can be executed by a subject (IoT devices) to access the object (resources). This access decision is processed by checking the matching rule with the list of all attributes that meet the requirement. The rules consist of i) identifier role, ii) type of access request, iii) identifier access action and iv) the list of attributes. The attributes must have the same attribute values in the resource, rs, and the requestor, known as IoT devices, u. We defined V as value of attributes which can be presented as follows: V rs_i(r,att) =V u_i(u,att).



Access decision { laptop , water, file, unique, update}

Subject[0x23e...]
Name: laptop
UUID:123e4567-e89b-12d3-a456-426614174000.
Org: FAMA
Place: Field A
Role: water

Object[0x27e...]
Name: File
UUID:81CBB42C-73A0-9660-6C7D-2FE94627F3A3.
Org: FAMA
Place: Field A

Rule [0x24e...]
Role: Manager, Water, Soil, Wind
rights: read/ access /edit/write
Type: unique/shared

Fig. 3. Decentralized access control framework for IoT security enhancement using blockchain technology in smart farming.

The flow of access control in the proposed framework was described by illustrating the interaction of network nodes with smart contracts, as illustrated in Fig. 4. The IoT owner nodes are responsible for managing access rules. The access rules consist of four factors: type of access, role, access action and a matching list of all attributes. Once rules are mined, they will be stored in IoT_ORMC (step 0.1). The IoT owner node is also responsible for storing and managing all attributes that consist of three different nodes, including full nodes, light nodes and ultralight nodes. The IoT owner node must assign a role for each node based on its responsibility and it will be stored in IoT_SRMC (step 0.2). When IoT_ACC obtains a request from any node, such as light nodes (laptop), to access resources (step 1), the request will be evaluated by IoT_ACC. Evaluation is accomplished by obtaining the access rules from IoT_ORMC (step 2) and acquiring all attribute information as well as the roles from IoT_SRMC (step 3). Finally, IoT_ACC verifies the access request by matching the access rule in IoT_SRMC and the attribute in IoT_ORMC (step 4). If the request is sufficient for access privileges, the requestor (laptop) can access the resources based on its roles.

A. Blockchain Nodes

In this framework, public and permissioned blockchains are adopted where nodes will be added and removed from the network with their identity verification. Since every node has a role and permission, the blockchain nodes require more CPU processing power and memory requires significant storage space to maintain the ledger copy [59]. We propose using four different nodes that allow heterogeneous IoT devices to access the blockchain network. Two nodes used for access control are categorised according to their capabilities, storage capacity and computing power [63]. The types of nodes and their responsibilities are categorised in the blockchain network, as follows:

- IoT requestor node is a requester that runs smart contracts for requesting access to resources. In the smart farming scenario, the requestor nodes are IoT devices or IoT sensors. Each node requestor is added and authenticated to the blockchain network by the IoT owner node before requesting access to resources. Requestor nodes represent three different nodes: full node, light node and ultralight node.

Full nodes are devices that have sufficient computing power and storage capabilities such as computers, laptops and servers that can perform full transactions. Light nodes are devices that have limited storage capabilities and computing power and can only store blockchain headers and support services for themselves. Mobile phone is one example of light nodes. Ultralight nodes are devices that have insufficient storage capabilities and computing power. Sensors and actuators are examples of ultralight nodes that require connection from the IoT gateway to P2P networks through communication technologies, such as Wi-Fi and ZigBee.

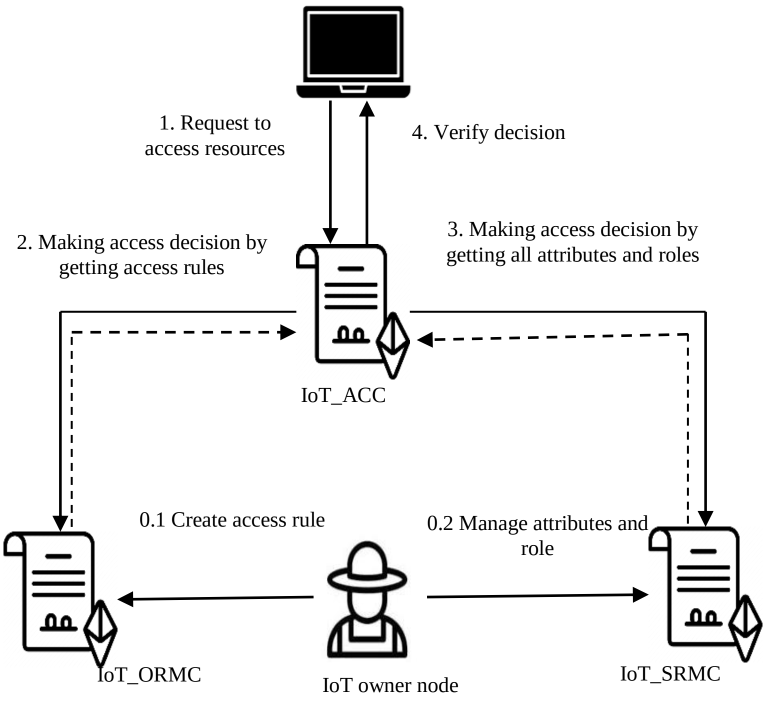

1. Request to access resources
4. Verify decision
2. Making access decision by getting access rules
3. Making access decision by getting all attributes and roles
IoT_ACC
0.1 Create access rule
0.2 Manage attributes and role
IoT_ORMC
IoT owner node
IoT_SRMC

Fig. 4. The interaction between IoT requestor node with smart contracts to access a resource.

- IoT owner node represents the owner of the IoT resources and devices in smart farming. An IoT owner node defines and deploys access control rules and permits requestor devices to enter the network through smart contracts. It can also add new IoT devices, manage node attributes as well as assign their roles that represent their responsibility.

B. *Smart Contract-based Access Control Rules*

In this research, for the verification mechanism in the blockchain network, we designed smart contract-based access control rules where the admin has the privilege to register devices to the blockchain network for the first time. Afterwards, all devices that request access to the network will be self-authenticated. This is particularly significant since smart farming consists of heterogeneous sensors that require self-administration to obtain access to the system at any time. In our smart contract design, we propose three smart contracts to avoid complexity. The smart contracts are: IoT_ACC, IoT_SRMC and IoT_ORMC. IoT_ACC is responsible for enforcing rules and making access decisions, IoT_ORMC is responsible for managing and updating the rules and resource attributes and IoT_SRMC is responsible for managing and assigning device attributes and the role of IoT devices.

1) Access Control Contract (IoT_ACC): IoT_ACC evaluates requests and provides access decisions made by access requests from IoT devices and sensors (subjects) to access resources (objects) in the system. This contract is executed by IoT devices when checking the pre-condition rules. The pre-condition rules will be matched based on the rules made from ORMC to the ACC to determine whether the subject has the right to perform actions on the object. To evaluate an access decision, this contract has two steps:

Step 1: Identification step

In this step, IoT_ACC will identify the type of request by IoT devices, either unique or multiple, and verify whether the IoT devices have sufficient requirements for acquiring access privileges. If the devices identified have access privileges, then the request will be saved and evaluated, otherwise, the request will be ignored. This step consists of two functions:

- typeReq(): used for the identification of type request. The request must have three things: user, ui, resources, ri, that want access and access action, acci. In this identification process, the user, ui, must contain identification, userID, that uses an Ethereum account and the list of user attributes, userAtt (e.g., location, time). Meanwhile, resource, ri, must have object identification, objectID, identification of resources belonging, refer_to, the list of resource attributes (e.g., location, type), resourceAtt, and access action, acci.

- requestAction(): used for deciding the type of request. The request typeReq() will pass value where a decision is made based on the object refer_to attribute.

Step 2: Evaluate the request

After successfully identifying the type of request, the evaluation process request is accomplished by retrieving precondition and evaluation constraints. To evaluate the access request, IoT_ACC must recognise whether or not the requestor is an active role and has rules. There are two functions in this step:

- activeRole(): to identify active role by checking the subject/IoT device via registering all attributes of subject/IoT devices in blockchain.

- getRule(): to retrieve rules that are specified in the form of tuple (rolei, typeAcci, accModei, attribute index list, attribute user, attribute resources). We determined rolei as the identifier of the role, while typeAcci represents the type of access. 1 represents shared access, while 0 represents private access. accModei is set as the identifier of access action, and the list of attribute index is defined as matching values of attributes in the resource, ri, and the user, ui. In tuple, the attribute user defines the values of attributes of the user. The attribute resources represent the attribute values of resources.

After evaluating the rules and active roles, IoT_ACC is conducted to evaluate three constraints defined in RBAC and ABAC.

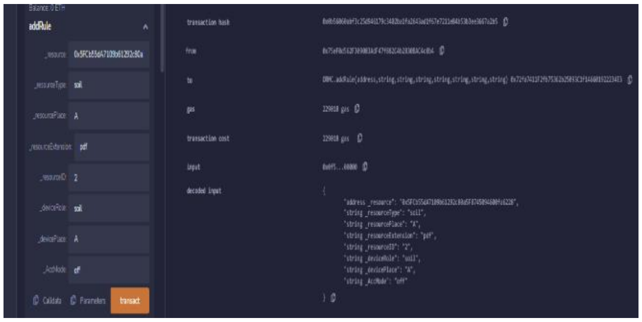
- User resource constraints were used to check whether the attributes in the object and user are the same values. If the values of attributes are the same, it will pass the value to currentRule() in the form of a Boolean function which is a true value.

- User constraints were used to check if attributes in devices are equal or the same values as the access rule. If the value of attributes is the same, it will pass the value to currentRule() in the form of a Boolean function which is a true value.

- Object constraints were used to check if attributes in resources are equal or same values as the access rule. If the value of attributes is the same, it will pass the value to currentRule() in the form of a Boolean function which is a true value.

After successfully validating the access request through several steps, the subject (the IoT device) verifies the results.

2) Object-Rule Management Contract (IoT_ORMC): IoT_ORMC specifies a policy by defining a set of access rules associated with each subject and resource based on two types of rules for resource access: shared access and private access, as shown in Fig. 5 the process of adding access rule. In this smart contract, only the IoT owner has the authority to execute the access rules. According to [8], these rules will be more efficient in reducing excessive permissions. Instead of checking user queries by using many rules, the model checks user queries by using only one rule. In this study, the set of access rules have four criteria: type of rule, access action, role and constraint, as shown in Table III.



```
_resource: 0x5FCb55dA7109b61292c80a
_resourceType: soil
_resourcePlace: A
_resourceExtension: pdf
_resourceID: 2
_deviceRole: soil
_devicePlace: A
_AccMode: off

transaction hash    0x0b5[illegible]
from                0x75e[illegible]
to                  ORMC.addRule(address,string,string,string,string,string,string,string) 0x72fa[illegible]
gas                 229818 gas
transaction cost    229818 gas
input               0x0f5...00000
decoded input       {
                        "address _resource": "0x5FCb55dA7109b61292c[illegible]",
                        "string _resourceType": "soil",
                        "string _resourcePlace": "A",
                        "string _resourceExtension": "pdf",
                        "string _resourceID": "2",
                        "string _deviceRole": "soil",
                        "string _devicePlace": "A",
                        "string _AccMode": "off"
                    }
```

Fig. 5. Process adding access rule function.

TABLE III. SET OF ACCESS CONTROL RULES FOR IOT_ORMC

Type of Rule	Access Action	Role	Constraint
unique	Update / write	Water	Extension rs: docx place rs: A type rs: water V u(u,att) = V r(rs, att).
multiple	Read	Water	Extension rs: pdf place rs: A type rs: water V u(u,att) = V r(rs, att).

- Type of rule: is the type to access resources. In this case, we divided access resources into two types: unique and multiple. Unique access is an editable resource, such as word (i.e., docx) and excel (i.e., xlsx). Multiple access is a non-editable source, such as portable document format (i.e., pdf), video (i.e., mp4, avi) and audio (i.e., wav, aif, mp3).
- Access action: is an action that performs by subject to access resources; for instance, read, write, view, control, etc.
- Role: is a character played by IoT devices (e.g., the device for watering plants is categorised under water group).
- Constraints: are access restrictions built on logical formula by donating the value function where the attribute value for the user is V u(u,att) and the attribute value for the resource is V r(rs, att). Constraints can also include other statements such as time or location which are environment attributes, V u(u,att).

3) Subject-role Management Contract (IoT_SRMC): In the process of access control, IoT devices can have their identity impersonated [64][64]. To address this security concern, IoT_SRMC is proposed to authenticate legitimate users who intend to access the IoT network by registering a new device in the IoT network. This contract adopts ABAC and RBAC models as a strategy for accessing control. It determines all attributes of IoT devices that can be used as valuable information to assess resources and assign roles to IoT devices. Each IoT device has a unique identifier (Ethereum account address) and multiple attributes associated with its ID, including location and role. This contract has functions for managing subject attributes and roles, such as adding, deleting and updating, which can only be performed by the IoT owner. In Table IV, all information about the IoT device is shown.

TABLE IV. SUBJECT REGISTRATION TABLE

device	deviceID	deviceType	deviceRole	devicePlace
Device A	0xA128F8 ……	laptop	water	field A
Sensor B	0xA134S8 ……	temperature	water	field A
Gateway A	0xA122A8 ……	gateway	soil	field A

C. Framework Flow

We present two types of form requests in this research. First part is the registration of new IoT devices and sensors; second part is the access request by IoT devices made through smart contracts. Fig. 6 illustrates the decentralised access control for IoT security enhancement. For the first part, the registration of new devices and sensors begin when an IoT owner issues a smart contract that implements a hybrid access control mechanism into the blockchain. Blockchain responds by issuing requests to the IoT owner and then creates a smart contract.

The IoT owner requests to register his own IoT devices and sensors, known as a subject, intended to authorise its device by providing all device and sensor attributes (i.e., name, location, identification, role, etc.). If no rule is made, the IoT owner must publish an access rule based on four criteria: i) types of access (shared, private), ii) role, iii) access action and iv) constraint. Lastly, after all access rules are complete, the transaction is stored in the blockchain.

For the second part, access is requested by IoT devices made through smart contracts where the IoT devices send a request for any service to access or update (i.e., data, file, storage unit) in the IoT network. Next, when the request of the subject is generated, IoT_ACC (main smart contract) is executed to control the overall access management. IoT_ACC will then obtain all information from IoT_ORMC and IoT_SRMC to match values between the access request, access rule and list of attributes to obtain the access decision for IoT devices. If all information shows the same values and authentication is successful, then access permission for IoT devices is complete. IoT_ACC then forwards back the return access result to IoT devices or corresponding objects. Finally, the result of access permission is stored in the blockchain network.

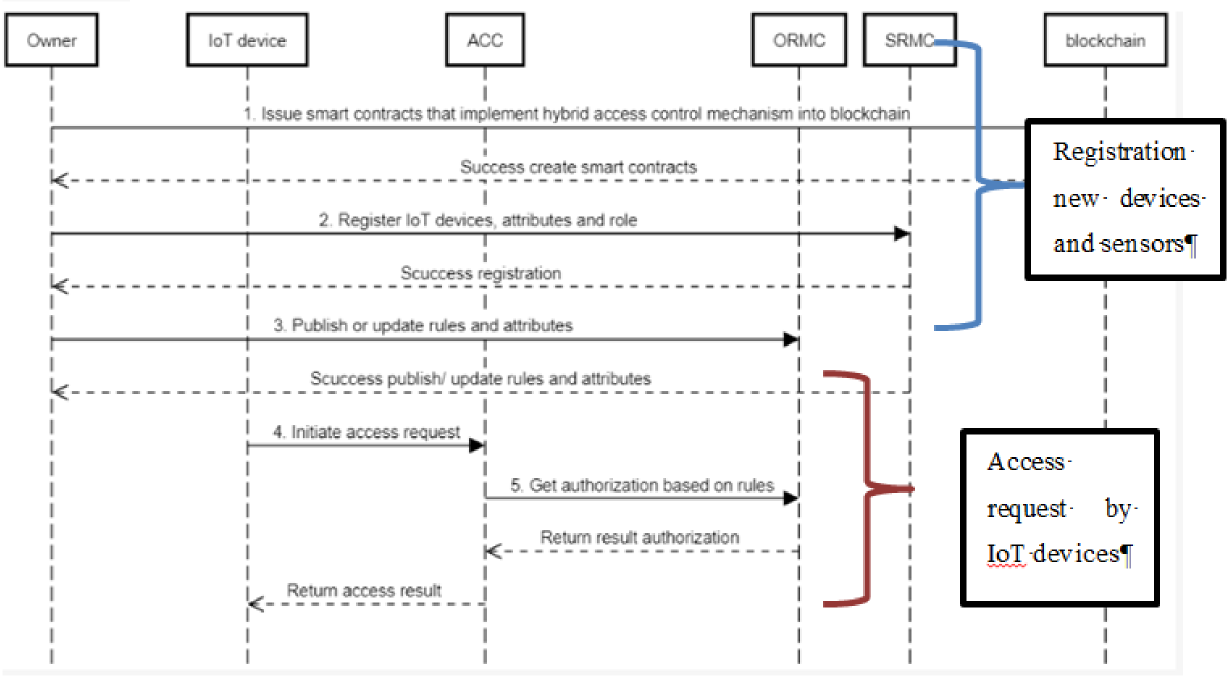

Owner
IoT device
ACC
ORMC
SRMC
blockchain
1. Issue smart contracts that implement hybrid access control mechanism into blockchain
Success create smart contracts
2. Register IoT devices, attributes and role
Success registration
3. Publish or update rules and attributes
Success publish/ update rules and attributes
4. Initiate access request
5. Get authorization based on rules
Return result authorization
Return access result
Registration new devices and sensors¶
Access request by IoT devices¶

Fig. 6. Access control flow mechanism in smart farming.

V. Evaluation

The main component of the proposed framework is the smart contracts that functioned as the verification mechanism. The deployment of smart contracts on the blockchain and the execution of associated contracts require payment of fees to the miner who mines the block. Thus, to evaluate our proposed framework, the smart contract cost consumption was measured by calculating the gas used for a transaction execution for the specific functions in smart contracts. The complexity of the task determines the quantity of gas consumed, with more gas being used for more complex tasks and the price of gas fluctuates over time. The fee required to perform a task is calculated by multiplying the consumed gas amount and the gas price. In this study, we run experiments for evaluation in the Ethereum network and the gas limit is set by the transaction initiator that determines the level of computational resources to be utilized in executing the transaction. In Ethereum, a unit called gas was employed to quantify the amount of work required to complete a task when deploying a smart contract. The initiator pays a fee for the gas used, which can vary depending on the gas limit set. The higher the amount paid, the easier the transaction will be executed [65] [66]. In this study, we determine cost per transaction by multiplying the gas price per unit with the gas limit (gasPrice X gasLimit) as per calculation in [51]. If the gas limit does not exceed the gas used, the execution of the transaction will be successful and it will be added or dropped in the blockchain network.

A. *Simulation Setting*

The experiments were conducted during the end of March 2023 when the value of 1 Ether is at the average of 1 eth ~ 1000000000 gwei. In this study, the simulation environment was set up in two layers with the hardware setting and the software setting, as displayed in Tables V and VI, respectively.

TABLE V. Hardware Setting

Items	Description
Operating system	Windows 10
Processor	AMD Ryzen 7 3700U with Radeon Vega Mobile Gfx 2.30 GHz
Memory	8 GB RAM

TABLE VI. Software Setting

Items	Description	Details
Language	Solidity	Used to build prototypes of smart contracts based on object-oriented programming language
Platform	Ethereum Network	Used as a public blockchain network in the virtual environment (EVM)
Compiler	Remix ide (version 0.5.17	Used to compile smart contracts
Test network	goerli testnet	Used to test networks
Gas limit	3,000,000 units	Used to set the amount of gas initiator that will execute the transaction

B. *Gas Usage and Cost per Operation in Smart Contracts*

The costs and gas usage in different smart contract deployments of IoT_ACC, IoT_ORMC and IoT_SRMC are shown in Table VII. Deployment of IoT_ACC smart contract requires 1,487,367 gas units. The gas cost in IoT_ACC is less than other smart contracts since IoT_ACC smart contract is only used for enforcing access decisions. Meanwhile, in IoT_ORMC smart contract, 2,196,564 gas units are required to create access rules based on ABAC and RBAC that execute the checking steps to determine if the role is a one-to-one relationship and active. In the meantime, to deploy IoT_SRMC smart contract, 1,677,746 gas units are required. Then, using the experiment's result, the cost of executing the IoT_ACC, IoT_ORMC and IoT_SRMC functions is calculated using Eq. (1).

$$\text{TxFee} = \text{gas} * \text{gasPrice} * 10\text{^}{-9} \qquad (1)$$

where gas represents the amount of gas used by the transaction, gasPrice represents the price of each gas unit, and 10^-9 is a conversion factor to convert the result into Ether. Hence, in this study, we determine that the cost value of executing each of the proposed smart contracts for IoT_ACC, IoT_ORMC and IoT_SRMC are 0.007212, 0.020180 and 0.019921 ether. This value serves as a benchmark for the cost operation for the proposed framework application in smart farming settings.

TABLE VII. GAS USAGE AND COST OF DIFFERENT SMART CONTRACT FUNCTIONS: IOT_ACC, IOT_IOT_RMC

Smart Contract Deployment	Description	Gas used	Cost (ether)
IoT_ACC Function	Responsible for enforcing rules and making access decisions	1,487,367	0.007212
IoT_ORMC Function	Responsible for managing and updating rules and resource attributes	2,196,564	0.020180
IoT_SRMC Function	Responsible for managing and assigning device attributes and the role of IoT devices	1,677,746	0.019921

VI. CONTRIBUTION

This paper proposes a novel decentralised access control framework for IoT security enhancement by adopting blockchain technology with the combination of ABAC and RBAC access control models. The aim of this proposed framework is to enhance the efficiency of access control management and secure IoT resources [8][9] with the scope of this study being smart farming. This framework aims to reduce the redundancy of permission required to authenticate devices to authorise and at the same time provide capacity for scalability. The pre-determined study objectives are as follows:

- We developed a decentralised access control framework embedded with blockchain technology to secure IoT resources from being accessed by unauthorised entities and scalable to cope with future expansion.
- We employed smart contracts as a fine-grained access control strategy to assign role-permission-based attributes that include object, subject and environment.
- We adopted two access control models (RBAC and ABAC) as an authentication element, including device attributes (name, location, type) and device roles in the smart farming environment.
- For validation, we applied Ethereum blockchain smart contract functionalities to issue, revoke or modify roles corresponding to a user, resource attributes and role permissions. The resource owner can further grant or deny access to resources.
- We conducted a simulation experiment to evaluate the framework component which is the smart contracts using gas cost measurements in the Ethereum network.

VII. CONCLUSION

The integration of IoT devices in smart farming facilitates the modernization of information and communication, resulting in increased productivity within the agricultural industry. To maintain the integrity of IoT resources and achieve security while also efficiently managing resources and ensuring scalability, a framework of decentralised access control using blockchain technology was developed in this study. The framework was developed based on findings from a detailed analysis published by researchers. Our proposed framework was developed by adopting blockchain technology to authenticate and authorize user and IoT device access while facilitating efficient resource management and scalability in IoT-enabled smart farming. The implementation of smart contracts is proposed to enhance the trust facilitated by the implementation of a ledger that is transparent and immutable. This study also suggests the implementation of the principle of role inheritance to reduce unnecessary user groups from access permission that can control the separation of duties, the list of privileges and confidentiality. In the framework, we proposed to combine blockchain technology as a decentralised approach with a hybrid access control model that consists of RBAC and ABAC. The proposed framework utilises a set of rules consisting of roles, access types, lists of attributes and actions that can be executed to obtain access permission by roles. The rules are divided into two types of access which are unique and multiple access. The proposed framework can resolve the challenge of role explosion, simplify management tasks, and reduce the complexity associated with traditional access control methods that rely on a centralized design. By doing so, the framework offers an effective solution to the limitations of current access control methods. It can enhance the overall security and resource management in decentralized IoT environments, thus improving the performance and efficiency of the access control mechanisms. The framework's main component which is the smart contracts was evaluated by measuring gas usage to determine cost operation using simulation. The finding can be used as a benchmark for comparison with the execution in the Mainnet network environment.

In future work, we will further evaluate the proposed framework using the measurement of the transaction throughput and network latency in blockchain by conducting more experiments. Also, an exploration towards a tokenised-based accelerating process for communication between smart contracts and IoT devices will be delivered to understand the effect of various attacks, such as DDOS attacks or man-in-the-middle attacks that compromise the integrity of data entry in smart farming.

ACKNOWLEDGMENT

This research is supported by the Fundamental Research Grant FRGS/1/2021/ICT07/UPNM/02/1. The authors fully acknowledge the National Defence University of Malaysia (UPNM) and the Ministry of Higher Education Malaysia (MOHE) for the approved fund.

REFERENCES

[1] H. Mahajan and A. Badarla, "Cross-Layer Protocol for WSN-Assisted IoT Smart Farming Applications Using Nature Inspired Algorithm," Wirel. Pers. Commun., vol. 121, Dec. 2021, doi: 10.1007/s11277-021-08866-6.

[2] A. Vangala, A. K. Das, V. Chamola, V. Korotaev, and J. J. P. C. Rodrigues, "Security in IoT-enabled Smart Agriculture : Architecture , Security Solutions and Challenges Security in IoT-enabled smart agriculture : architecture , security solutions and challenges," Cluster Comput., no. April, 2022, doi: 10.1007/s10586-022-03566-7.

[3] M. K. Saini, "Internet of Things (IoT) Applications and Security Challenges : A Review," vol. 7, no. 12, pp. 1–7, 2019.

[4] A. Rettore, D. A. Zanella, L. Carlos, and P. Albini, "Security challenges to smart agriculture : Current state , key issues , and future directions," Array, vol. 8, no. October, p. 100048, 2020, doi: 10.1016/j.array.2020.100048.

[5] R. Fotohi and F. Shams Aliee, "Securing communication between things using blockchain technology based on authentication and SHA-256 to improving scalability in large-scale IoT," Comput. Networks, vol. 197, no. December 2020, p. 108331, 2021, doi: 10.1016/j.comnet.2021.108331.

[6] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of Trust: A decentralized blockchain-based authentication system for IoT," Comput. Secur., vol. 78, no. 2018, pp. 126–142, 2018, doi: 10.1016/j.cose.2018.06.004.

[7] X. Yang et al., "A Survey on Smart Agriculture : Development Modes , Technologies , and Security and Privacy Challenges," IEEE/CAA J. Autom., vol. 8, no. 2, pp. 1–30, 2020.

[8] H. Ben Attia, L. Kahloul, and S. Benharzallah, "FRABAC: A new hybrid access control model for the heterogeneous multi-domain systems," Int. J. Manag. Decis. Mak., vol. 17, no. 3, pp. 245–278, 2018, doi: 10.1504/IJMDM.2018.093493.

[9] S. Pal, M. Hitchens, V. Varadharajan, and T. Rabehaja, "Policy-based access control for constrained healthcare resources in the context of the Internet of Things," J. Netw. Comput. Appl., vol. 139, no. April, pp. 57–74, 2019, doi: 10.1016/j.jnca.2019.04.013.

[10] N. Mancosu, R. L. Snyder, G. Kyriakakis, and D. Spano, "Water scarcity and future challenges for food production," Water (Switzerland), vol. 7, no. 3, pp. 975–992, 2015, doi: 10.3390/w7030975.

[11] V. N. Malavade and P. K. Akulwar, "Role of IoT in Agriculture," Natl. Conf. "Changing Technol. Rural Dev., vol. 1, no. 13, pp. 422–425, 2019.

[12] N. H. Motlagh, M. Mohammadrezaei, J. Hunt, and B. Zakeri, "Internet of things (IoT) and the energy sector," Energies, vol. 13, no. 2, pp. 1–27, 2020, doi: 10.3390/en13020494.

[13] F. J. Ferrández-Pastor, J. M. García-Chamizo, M. Nieto-Hidalgo, and J. Mora-Martínez, "Precision agriculture design method using a distributed computing architecture on internet of things context," Sensors (Switzerland), vol. 18, no. 6, 2018, doi: 10.3390/s18061731.

[14] A. D. Jurcut, P. Ranaweera, and L. Xu, Introduction to IoT Security , no. December. 2020. doi: 10.1002/9781119527978.ch2.

[15] M. Wazid, A. K. Das, S. Shetty, P. Gope, and J. J. P. C. Rodrigues, "Security in 5G-Enabled Internet of Things Communication: Issues, Challenges and Future Research Roadmap," IEEE Access, vol. 9, 2020, doi: 10.1109/ACCESS.2020.3047895.

[16] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal, "Security and Privacy in Smart Farming: Challenges and Opportunities," IEEE Access, vol. 8, pp. 34564–34584, 2020, doi: 10.1109/ACCESS.2020.2975142.

[17] T. Zengeya, P. Sambo, N. Mabika, and C. Science, "Trust In The Adoption Of Internet Of Things For Smart Agriculture In Developing Countries," vol. 12, no. 3, pp. 11–22, 2021.

[18] G. Ikrissi and T. Mazri, "IoT=-Based Smart Environments : State Of The Art , Security Threats And Solutions," vol. XLVI, no. October, pp. 27–29, 2021.

[19] T. Le and M. W. Mutka, "Access control with delegation for smart home applications," IoTDI 2019 - Proc. 2019 Internet Things Des. Implement., pp. 142–147, 2019, doi: 10.1145/3302505.3310076.

[20] E. Bertin, D. Hussein, C. Sengul, and V. Frey, "Access Control in the Internet of Things: A Survey of Existing Approaches and Open Research Question," Proc. ACM Conf. Comput. Commun. Secur., pp. 1056–1073, 2018, doi: 10.1145/3243734.3243817.

[21] G. Fedrecheski, L. C. C. De Biase, P. C. Calcina-Ccori, R. De Deus Lopes, and M. K. Zuffo, "SmartABAC: Enabling Constrained IoT Devices to Make Complex Policy-Based Access Control Decisions," IEEE Internet Things J., vol. 9, no. 7, pp. 5040–5050, 2022, doi: 10.1109/JIOT.2021.3110142.

[22] L. Song, X. Ju, Z. Zhu, and M. Li, "An access control model for the Internet of Things based on zero-knowledge token and blockchain," Eurasip J. Wirel. Commun. Netw., vol. 2021, no. 1, 2021, doi: 10.1186/s13638-021-01986-4.

[23] M. Alshahrani and I. Traore, "Secure mutual authentication and automated access control for IoT smart home using cumulative Keyed-hash chain," J. Inf. Secur. Appl., vol. 45, pp. 156–175, 2019, doi: 10.1016/j.jisa.2019.02.003.

[24] K. Lei et al., "Blockchain-Based Cache Poisoning Security Protection and Privacy-Aware Access Control in NDN Vehicular Edge Computing Networks," J. Grid Comput., vol. 18, no. 4, pp. 593–613, 2020, doi: 10.1007/s10723-020-09531-1.

[25] J. Brogan, I. Baskaran, and N. Ramachandran, "Authenticating Health Activity Data Using Distributed Ledger Technologies," Comput. Struct. Biotechnol. J., vol. 16, pp. 257–266, 2018, doi: 10.1016/j.csbj.2018.06.004.

[26] C. Lin, D. He, X. Huang, K. K. R. Choo, and A. V. Vasilakos, "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0," J. Netw. Comput. Appl., vol. 116, no. February, pp. 42–52, 2018, doi: 10.1016/j.jnca.2018.05.005.

[27] S. Hong, "P2P networking based internet of things (IoT) sensor node authentication by Blockchain," Peer-to-Peer Netw. Appl., vol. 13, no. 2, pp. 579–589, 2020, doi: 10.1007/s12083-019-00739-x.

[28] J. P. Cruz, Y. Kaji, and N. Yanai, "RBAC-SC: Role-based access control using smart contract," IEEE Access, vol. 6, pp. 12240–12251, 2018, doi: 10.1109/ACCESS.2018.2812844.

[29] M. U. Rahman, "Scalable Role-based Access Control Using The EOS Blockchain," 2020.

[30] J. Huang, D. M. Nicol, R. Bobba, and J. H. Huh, "A framework integrating attribute-based policies into role-based access control," Proc. ACM Symp. Access Control Model. Technol. SACMAT, pp. 187–196, 2012, doi: 10.1145/2295136.2295170.

[31] A. Ismail, Q. Wu, M. Toohey, Y. C. Lee, Z. Dong, and A. Y. Zomaya, "TRABAC: A Tokenized Role-Attribute Based Access Control using Smart Contract for Supply Chain Applications," Proc. - 2021 IEEE Int. Conf. Blockchain, Blockchain 2021, no. December 2021, pp. 584–588, 2021, doi: 10.1109/Blockchain53845.2021.00088.

[32] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A Novel Attribute-Based Access Control Scheme Using Blockchain for IoT," IEEE Access, vol. 7, pp. 38431–38441, 2019, doi: 10.1109/ACCESS.2019.2905846.

[33] S. Bhatt, T. K. Pham, M. Gupta, J. Benson, J. Park, and R. Sandhu, "Attribute-Based Access Control for AWS Internet of Things and Secure Industries of the Future," IEEE Access, vol. 9, pp. 107200–107223, 2021, doi: 10.1109/ACCESS.2021.3101218.

[34] P. Wang, Y. Yue, W. Sun, and J. Liu, "An Attribute-Based Distributed Access Control for Blockchain-enabled IoT," 2019 Int. Conf. Wirel. Mob. Comput. Netw. Commun., pp. 1–6, 2019.

[35] J. Wang, H. Wang, H. Zhang, and N. Cao, "Trust and Attribute-Based Dynamic Access Control Model for Internet of Things," Proc. - 2017 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. CyberC 2017, vol. 2018-Janua, pp. 342–345, 2017, doi: 10.1109/CyberC.2017.47.

[36] Y. Zhang, B. Li, B. Liu, J. Wu, Y. Wang, and X. Yang, "An attribute-based collaborative access control scheme using blockchain for IoT

devices," Electron., vol. 9, no. 2, 2020, doi: 10.3390/electronics9020285.

[37] S. Liu, J. Yu, Y. Xiao, Z. Wan, S. Wang, and B. Yan, "BC-SABE: Blockchain-Aided Searchable Attribute-Based Encryption for Cloud-IoT," IEEE Internet Things J., vol. 7, no. 9, pp. 7851–7867, 2020, doi: 10.1109/JIOT.2020.2993231.

[38] S. Bhatt and R. Sandhu, "ABAC-CC: Attribute-based access control and communication control for internet of things," Proc. ACM Symp. Access Control Model. Technol. SACMAT, pp. 203–212, 2020, doi: 10.1145/3381991.3395618.

[39] L. Cruz-Piris, D. Rivera, I. Marsa-Maestre, E. De La Hoz, and J. R. Velasco, "Access control mechanism for IoT environments based on modelling communication procedures as resources," Sensors (Switzerland), vol. 18, no. 3, 2018, doi: 10.3390/s18030917.

[40] T. Rabehaja, S. Pal, and M. Hitchens, "Design and implementation of a secure and flexible access-right delegation for resource constrained environments," Futur. Gener. Comput. Syst., vol. 99, pp. 593–608, 2019, doi: 10.1016/j.future.2019.04.035.

[41] Y. Zhang, R. H. Deng, G. Han, and D. Zheng, "Secure smart health with privacy-aware aggregate authentication and access control in Internet of Things," J. Netw. Comput. Appl., vol. 123, no. June, pp. 89–100, 2018, doi: 10.1016/j.jnca.2018.09.005.

[42] M. U. Aftab et al., "A Hybrid Access Control Model with Dynamic COI for Secure Localization of Satellite and IoT-Based Vehicles," IEEE Access, vol. 8, pp. 24196–24208, 2020, doi: 10.1109/ACCESS.2020.2969715.

[43] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems," IEEE Access, vol. 7, pp. 66792–66806, 2019, doi: 10.1109/ACCESS.2019.2917555.

[44] D. Prashar, N. Jha, S. Jha, Y. Lee, and G. P. Joshi, "Blockchain-based traceability and visibility for agricultural products: A decentralizedway of ensuring food safety in India," Sustain., vol. 12, no. 8, 2020, doi: 10.3390/SU12083497.

[45] A. F. Hussein, N. ArunKumar, G. Ramirez-Gonzalez, E. Abdulhay, J. M. R. S. Tavares, and V. H. C. de Albuquerque, "A medical records managing and securing blockchain based system supported by a Genetic Algorithm and Discrete Wavelet Transform," Cogn. Syst. Res., vol. 52, pp. 1–11, 2018, doi: 10.1016/j.cogsys.2018.05.004.

[46] M. N. M. Bhutta et al., "A Survey on Blockchain Technology: Evolution, Architecture and Security," IEEE Access, vol. 9, no. April, pp. 61048–61073, 2021, doi: 10.1109/ACCESS.2021.3072849.

[47] G. Zhao et al., "Blockchain technology in agri-food value chain management: A synthesis of applications, challenges and future research directions," Comput. Ind., vol. 109, pp. 83–99, 2019, doi: 10.1016/j.compind.2019.04.002.

[48] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," Proc. - 2017 IEEE 6th Int. Congr. Big Data, BigData Congr. 2017, no. June, pp. 557–564, 2017, doi: 10.1109/BigDataCongress.2017.85.

[49] S. H. Alsamhi and B. Lee, "Blockchain-Empowered Multi-Robot Collaboration to Fight COVID-19 and Future Pandemics," IEEE Access, vol. 9, pp. 44173–44197, 2021, doi: 10.1109/ACCESS.2020.3032450.

[50] N. A. M. Razali, W. N. W. Muhamad, K. K. Ishak, N. J. A. M. Saad, M. Wook, and S. Ramli, "Secure Blockchain-Based Data-Sharing Model and Adoption among Intelligence Communities," IAENG Int. J. Comput. Sci., vol. 48, no. 1, 2021.

[51] M. M. A. Khan, H. M. A. Sarwar, and M. Awais, "Gas consumption analysis of Ethereum blockchain transactions," Concurr. Comput. Pract. Exp., vol. 34, no. 4, p. e6679, Feb. 2022, doi: https://doi.org/10.1002/cpe.6679.

[52] T. Jaikla, C. Vorakulpipat, E. Rattanalerdnusorn, and H. D. Hai, "A secure network architecture for heterogeneous IoT devices using role-based access control," 2019 27th Int. Conf. Software, Telecommun. Comput. Networks, SoftCOM 2019, 2019, doi: 10.23919/SOFTCOM.2019.8903605.

[53] N. H. N. Zulkipli and G. B. Wills, "An event-based access control for IoT," ACM Int. Conf. Proceeding Ser., pp. 0–3, 2017, doi: 10.1145/3018896.3025170.

[54] N. Noor, N. Matrazali, N. Malizan, K. Ishak, M. Wook, and N. Hasbullah, "Decentralized Access Control using Blockchain Technology for Application in Smart Farming," Int. J. Adv. Comput. Sci. Appl., vol. 13, Jan. 2022, doi: 10.14569/IJACSA.2022.0130993.

[55] E. Chen, Y. Zhu, Z. Zhou, S. Y. Lee, W. E. Wong, and W. C. C. Chu, "Policychain: A Decentralized Authorization Service With Script-Driven Policy on Blockchain for Internet of Things," IEEE Internet Things J., vol. 9, no. 7, pp. 5391–5409, 2022, doi: 10.1109/JIOT.2021.3109147.

[56] P. Wang, N. Xu, H. Zhang, W. Sun, and A. Benslimane, "Dynamic Access Control and Trust Management for Blockchain-Empowered IoT," IEEE Internet Things J., vol. 9, no. 15, pp. 12997–13009, 2022, doi: 10.1109/JIOT.2021.3125091.

[57] R. Wang, X. Wang, W. Yang, S. Yuan, and Z. Guan, "Achieving fine-grained and flexible access control on blockchain-based data sharing for the Internet of Things," China Commun., vol. 19, no. 6, pp. 22–34, 2022, doi: 10.23919/JCC.2022.06.003.

[58] S. Y. A. Zaidi et al., "An attribute-based access control for IoT using blockchain and smart contracts," Sustain., vol. 13, no. 19, pp. 1–26, 2021, doi: 10.3390/su131910556.

[59] Y. E. Oktian and S. G. Lee, "BorderChain: Blockchain-Based Access Control Framework for the Internet of Things Endpoint," IEEE Access, vol. 9, pp. 3592–3615, 2021, doi: 10.1109/ACCESS.2020.3047413.

[60] S. Pal, T. Rabehaja, M. Hitchens, V. Varadharajan, S. Member, and A. Hill, "On the Design of a Flexible Delegation Model for the Internet of Things Using Blockchain," IEEE Trans. Ind. Informatics, vol. PP, no. c, p. 1, 2019, doi: 10.1109/TII.2019.2925898.

[61] R. Nakanishi, Y. Zhang, M. Sasabe, and S. Kasahara, "IOTA-Based Access Control Framework for the Internet of Things," 2020 2nd Conf. Blockchain Res. Appl. Innov. Networks Serv. BRAINS 2020, pp. 87–91, 2020, doi: 10.1109/BRAINS49436.2020.9223293.

[62] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," IEEE Internet Things J., vol. 6, no. 2, pp. 1594–1605, 2019, doi: 10.1109/JIOT.2018.2847705.

[63] M. Cinque, C. Esposito, S. Russo, and O. Tamburis, "Blockchain-empowered decentralised trust management for the Internet of Vehicles security," Comput. Electr. Eng., vol. 86, p. 106722, 2020, doi: 10.1016/j.compeleceng.2020.106722.

[64] M. Wazid, B. Bera, A. Mitra, A. K. Das, and R. Ali, "Private blockchain-envisioned security framework for AI-enabled IoT-based drone-aided healthcare services," DroneCom 2020 - Proc. 2nd ACM MobiCom Work. Drone Assist. Wirel. Commun. 5G Beyond, pp. 37–42, 2020, doi: 10.1145/3414045.3415941.

[65] S. Bouraga, "An Evaluation of Gas Consumption Prediction on Ethereum based on Transaction History Summarization," Sep. 2020. doi: 10.1109/BRAINS49436.2020.9223288.

[66] T. Sultana, A. Almogren, M. Akbar, M. Zuair, I. Ullah, and N. Javaid, "Data sharing system integrating access control mechanism using blockchain-based smart contracts for IoT devices," Appl. Sci., vol. 10, no. 2, 2020, doi: 10.3390/app10020488.