Security and Privacy Implementation in Smart Home: Attributes Based Access Control and Smart Contracts

Amjad Qashlan
University of Technology Sydney (UTS)
Faculty of Engineering and IT (FEIT)
Australia, NSW, Sydney
Email: Amjad.Qashlan@student.uts.edu.au

Priyadarsi Nanda
University of Technology Sydney (UTS)
Faculty of Engineering and IT (FEIT)
Australia, NSW, Sydney
Email: Priyadarsi.Nanda@uts.edu.au

Xiangian He
University of Technology Sydney (UTS)
Faculty of Engineering and IT (FEIT)
Australia, NSW, Sydney
Email:Xiangjian.He@uts.edu.au

***Abstract*—There has been wide range of applications involving smart home systems for user comfort and accessibility to essential commodities. Users enjoy featured home services supported by the IoT smart devices. These IoT devices are resource-constrained, incapable of securing themselves and can be easily hacked. Edge computing can provide localized computations and storage which can augment such capacity limitations for IoT devices. Furthermore, blockchain has emerged as technology with capabilities to provide secure access and authentication for IoT devices in decentralized manner. In this paper, we propose an authentication scheme which integrate attribute based access control using smart contracts with ERC-20 Token (Ethereum Request For Comments) and edge computing to construct a secure framework for IoT devices in Smart home system. The edge server provide scalability to the system by offloading heavier computation tasks to edge servers. We present system architecture and design and discuss various aspects related to testing and implementation of the smart contracts. We show that our proposed scheme is secure by thoroughly analysing its security goals with respect to confidentiality, integrity and availability. Finally, we conduct a performance evaluation to demonstrate the feasibility and efficiency of the proposed scheme.**

***Keywords*—** Blockchain, Smart Home, Access Control, Smart Contract, Cyber Threats

I. Introduction

Residence building with an integrated Internet of Things (IoT) network that affords homeowners outcomes such as increased comfort, security and quality of life. As such, a smart home network is underpinned by the IoT infrastructure, which connects heterogeneous smart devices (e.g. smartphones, smart meters, wearable devices and the like. Smart home systems can both enable and enhance people's ability to live independently. They include a suite of invaluable technologies including those to monitor and assess health, thus making them attractive to users and device designers. Not surprisingly, it is predicted that by 2022 the value of the global smart home market will exceed $53 billion. This prediction is based on almost 21 percent annual rate of growth forecast for the market from 2018 to 2022. Although the benefits of smart homes to homeowners and stakeholders are well documented, several risks must also be considered including cyber-attacks and threats to the data security and privacy of users [1].

Traditional approaches to the resolution of such risks rely on centralised frameworks which are susceptible to cyberattacks. Hence, access control function is important for preventing access to unauthorised users via explicit or implied specifications and only permitting access to the resources for authorised party. Access controls have traditionally been supported by a centralised system which are relatively simple to manage [2]. This means a central server is used to process all access controls; namely, assigning access rights, managing access (e.g., updates, revocations) and access verifications. However, there are risks around the server being the point of failure due to 'natural' (functional) or external forces (cyberattack) and potentially compromising the access control system. Furthermore, the large scale and distributed nature of IoT systems means there are difficulties related to controlling requests by centralised schemes to access the desired resource [2].

Distributed access control networks can counter some of the above limitations presented by centralised networks. These networks perform the processes related to access control using multiple nodes rather than a single server. The nodes 'agree' on the rights to be assigned, the policies to provide access and the verification results to provide solid and reliable access controls that can resist malicious attacks. As a result, there is growing interest in utilising emergent blockchain technology for distributed and reliable access control.

The emergence of distributed and tamper-resistant ledger based blockchain mechanisms to protect data is providing new options for resolving data privacy, data security, and data integrity issues in smart homes. Blockchain technology demonstrates strong performance across a range of smart home applications including control over access to the home, data sharing, and so forth. The implementation of blockchain in smart home networks is also justified on the basis that it exists independently of current heterogeneous protocols often applied in smart homes (e.g., Z-Wave, Zigbee, Bluetooth and Thread). Nonetheless, due to the high level of resources consumed during the mining and consensus procedures and the limitations of the node resources in smart home devices, it is challenging to use blockchain directly in a smart home. In turn, edge computing offers an alternative and complimentary method for managing proof-of-work (PoW) puzzles and supports the blockchain applications in a smart home. Edge computing takes place at the network extremes (edges) by extending the distribution of cloud-based resources and services. It supports a multi-access system for users to access cloud-like services for enhanced computing, applications, and storage. Resource-constrained smart home appliances can consequently increase their computing capabilities by divesting the mining and storage jobs to edge servers. The incorporation of blockchain and edge computing sets up a decentralised system for computation outsourcing and storage security related to scalable and safety proof operations [3].

To address the concerns discussed above and motivated by the advantages of integrated blockchain technology and edge computing,

we present a novel lightweight Ethereum blockchain based multi-tier edge-smart home architecture. In our framework, every single home has multi edge servers as local blockchain miners and the smart contracts are used to enforce the rules and policies in automated manner to regulate the smart home IoT devices based on the Attribute Based Access Control (ABAC).

Specially, we present an architecture involving authentication rules and logic based on Ethereum smart contract integrated edge computing.

- We propose ERC-20 token generation and attribute based access control mechanism that utilizes Ethereum smart contracts integrated with edge computing(servers) for authenticating user access to IoT smart home devices. The access token are issued by the smart contracts with no intermediary or trusted third party.
- We present the details of the overall system including the architectural design, workflow scenario, and interactions among entities with the smart contracts including the attribute access control scheme designed to provide protection from illegal data access in smart home system.
- We provide the full design of the Ethereum smart contract including the implementation and the testing scenarios.
- We discuss the performance evaluation of the proposed scheme and compare with existing models with respect to various performance metrics.
- We present security analysis of our proposed authentication scheme, and discuss how the scheme achieves security goals (confidential, integrity, and availability), and is able to overcome modification and denial of service (DoS) attacks.

The remaining sections of this article are organized as follows. Section II presents relevant background information about core technology. Section III review the existing works in Blockchain and smart home. The proposed solution is implemented and described in Section IV. We investigate the main result of security and performance analysis in Section V. Finally, in Section VI, we conclude the paper and provide direction for future work.

II. Research Background

A. Smart home

A standard definition of 'smart home' remains unavailable, although numerous articles have sought to establish the criteria for such a definition. It is generally accepted that 'smart home' refers to a place of residence that has within it the technological capabilities to support task automation, monitoring of people and activities, and mechanisms to maintain good health. All units in a smart home can communicate with each other via the network and can be operated both internally (i.e., from inside the home) or remotely via the Internet. This type of system has great potential given its diverse range of applications including to enhance security, provide a more energy-efficient alternative, and for increasing user convenience [4]. In this paper, we adopt a more holistic definition of a smart home; a home that utilises Internet-connected devices to support diverse functionalities. The type of devices in a smart home may include smart-TVs, smart-temperature controls, smart-hubs and further integrated devices. Companies providing IoT devices for smart homes typically require access to their interface to control the devices. As such, smart homes with multiple devices from different companies can have a number of disconnected interfaces, which require a well-defined device management. IoT devices do not have the resources to execute security actions because most of the device's resources are utilised to perform different functions [5]. Hence, what is needed is a security mechanism that incorporates the required processes to address the present IoT issues without having to utilise significant resources.

B. Access control scheme

Access control systems are typically built upon access control lists (ACLs) that give access rights to users. ACLs are increasingly more complicated to govern when there is an increase in the number of users seeking resources. As a remedy to this ACL system limitation, designers have developed Role Based Access Control (RBAC) systems which add an intermediate layer into the process of assigning role rights rather than giving them directly to users and then giving users their roles. This method can significantly diminish the effort needed to oversee the access control rules. This is even when there has been a surge in the subject roles and the number of resources or when the system comprises multiple administrative fields. Attribute Based Access Control (ABAC) systems attempt to resolve the problems around surges in the number of roles by providing the option to apply the subject's properties directly along with the resources and environmental properties. This can be done to specify the access policies and thus potentially reduce the number of rules, or rule changes. However, ABAC still needs to access a consistent definition of the field attribute or the definition of attributes across different fields [6]. This paper examines attribute base access control particularly because it is deemed to be an appropriate decentralised model for IoT setup and provides scalability, flexible and strong dynamics.

C. Blockchain Technology

Blockchain is characterised as shared decentralised, and distributed and unchangeable ledger that keeps a registry of the assets and transactions on a peer-to-peer (P2P) network [7]. Each transaction in blockchain is digitally registered and validated by thousands of network-based mining nodes. Time stamps are used to store and organise all transactions in 'blocks' (groups). Several blocks then form a chain referred to as a 'blockchain'. The blockchain relies on elliptic curve cryptography (ECC) along with a SHA-2 hashing scheme for robust cryptographic proof to support the authentication and integrity of the data. A well-known example of the use of the blockchain infrastructure is Bitcoin. Broadly speaking, the blockchain infrastructure used by Bitcoin underpins the technology used for most cryptocurrencies. In turn, emergence of the Ethereum blockchain and its use of smart contracts has delivered an almost endless amount of crypto currencies [8].

D. Ethereum with Smart Contract

As a distributed platform, Ethereum includes the functions of smart contracts. Developed by Vitalik Buterin in 2013, the Ethereum smart contract supports event-directed, Turing complete scripting functionalities for verifying and processing complex transactions to show the viability of the contract transaction. In terms of the smart contract, it functions in a way similar to an event-directed script and performs the script automatically upon satisfaction of the pre-defined conditions. Prior to the execution of the smart contract, all associated functions and processes must be in place [9]. Two account types exist in Ethereum: Externally Owned Accounts (EOA) and Contract Accounts. Each account type has a unique 20-byte hexadecimal string identification address. The private key of the owner controls the EOA, which includes an ether balance, and transmits transactions (i.e., send a message to prompt the initiation of a smart contract). An EOA does not have an associated code. Alternatively, a contract account, which also includes an ether balance has an associated code that is triggered by another smart contract or by a transaction.

E. ERC-20 Token

As a technical standard, ERC-20 emerged as one of the most essential and significant tokens used for the entire smart contracts living in the Ethereum Blockchain [10]. ERC-20 stand for "Ethereum Request For Comments" and the number 20 acts as a unique identifier for differentiation from the other standards. It is a standard protocol used for creating blueprints of smart contracts based on Ethereum by

defining set of standards and rules for token issue on the Ethereum network. Within the system of Ethereum, for the advantage of other tokens, ERC-20 defines a set of six functions.

1) totalSupply(): To determine the total count of tokens that created and exist in the system.

2) balanceOf(address owner): to returns the count of tokens that a specific address has in their account.

3) allowance (address tokenowner, address spender): Balance of the User is one of the most critical data required to conduct a transaction. The user should have a minimum count of tokens to carry out a particular transaction. The allowance() function is used to cancel the transaction if the user does not have the minimum required number of tokens.

4) approve(address spender, unit tokens): Once the user has the required number of tokens to carry out a transaction, and the balance is checked, the contract owner approves to collect the required count of tokens from the contract's address. This function also verifies that there are no extra or missing tokens by checking the transaction against the total token supply.

5) transfer(address to, unit tokens): This transfer() function enables the contract owner to send tokens. It enables the contract owner to transfer amounts of the token to other addresses. Also enables a definite number of token transfer between the total supply and a user account.

6) transferFrom(address from, address to, uint256 tokenId): In addition to the transfer function, the transferFrom() function allows payment transfer automation to specific accounts. Technically a token is a smart contract that tracks 'who owns' and 'how much' of that particular token.

F. Edge computing

The capacity of cloud computing over recent years to make available unlimited computing, data storage and systems management resources has led to the development of many cloud-based applications and the fast-paced expansion of Internet-based companies such as Amazon. Recently, the trend has been to transition from cloud functions to network edges [11]. This relies on delay-sensitive applications (e.g., virtual reality) that have rigorous delay conditions. Edge computing has increased the pressure on cloud resources and services to support mobility support, location recognition, and reduced latency. Such affordances position network edge technology is integral realising the future IoT [12].

Three levels in the edge computing structure: end device (front-end), edge server (near-end), and core cloud (far-end). The three-level hierarchy indicates the computing capacity of the elements and their characteristics of edge computing. Front-end such as sensors and actuators deliver additional and enhanced user responsiveness. The resource requirements have to be dispatched to the server, however, given their restricted capacity. Near-end edge servers support the bulk of the flow of traffic across the network in addition to various resource requirements (e.g., real-time data processing and computation offloading. As a result, end users are provided with enhanced computation performance, with some latency increase, using edge servers. Far-end cloud servers offer additional computing power (e.g., big data analytics) and extra data storage space with a transmission latency.

The objective of this system architecture is to support the edge network to perform compute-intensive and delay-sensitive applications. Additionally, some edge server applications provide data synchronisation via communications with the cloud.

III. Previous works

Security and privacy of IoT devices in smart home is the biggest concerns because connected IoT devices are vulnerable to various attacks and they lack basic security feature .To address these issues, numerous centralized solutions have been proposed [13]. However, the communication and processing overhead on centralized solutions, access control and single point of failure are major challenges. Therefore various researchers [5, 13-17] have turned-out the attention towards distributed Framework and proposed popular blockchain based solutions for various IoT use cases.

Reference [14] has examined the issues around 'gateways' or connections among IoT devices, suggesting that such centralised structures "present multiple security vulnerabilities such as integrity, certification, and availability" (p. 2). In response, the authors proposed a blockchain-based smart home gateway network that can ward off possible gateway attacks. Consisting of three layers: device, gateway, and the cloud, the blockchain technology network is used at the gateway layer to support decentralisation where blocks of data are stored and exchanged. This helps to maintain the integrity of the data both inside and outside of the smart home and ensures availability via authentication and communication among network members. However, their architecture has some limitation in teams of additional computational complexity by blockchain operation in the gateways.

Reference [15] argued the benefits of using Ganache, Remix, and web3.js as architecture for smart home based IoT-Blockchain (SHIB) to resolve the challenges around data privacy, trust access control, and the capacity to extend the system. They proposed IoT gateway to connect a cluster of IoT devices to blockchain network in smart home. Though, their work is little complicated because every user and IoT devices have a policy associated with one and only one subject-object pair, and the gateway may not have sufficient computing power to deal with the large transactions.

Reference [16] proposed a private Blockchain-based access control (PBAC) scheme to address issues around data security and privacy protections when using smart devices in smart home systems. The proposed PBAC provides "an unforgeable and auditable foundation" within the IoT system that can block illegal data access; preserve data security against attacks; and provide accurate, robust, and timely access to records. However, they proposed only one online server as administrator If the administrator is inactive, the whole system fails. Reference [13] proposed the use of a Blockchain-based approach using Proof-of-Authority to establish a consensus mechanism to better manage home appliances within a decentralized framework. The authors demonstrated the enhanced effectiveness of a Blockchain approach using Proof-of-Authority as the consensus mechanism to address security concerns compared to the use of a traditional Proof-of-Work based system. Reference [17] examined the application of IoT and Blockchain-Based Multi-Sensory Frameworks in the specific context of in-home quality of life (QOL) for recently diagnosed cancer patients. The Blockchain and off-chain based framework proposed by the authors permits multiple medical and ambient intelligent IoT sensors to collect QOL data from the smart home envi/ronment and to share it securely with a specific community of interest. The in-home secure monitoring system collects QOL data including transactional records and multimedia-based big data (e.g. physiological and mental state data) and can be managed by the Blockchain-based data analytics developed by the authors.

Reference [5] proposed a lightweight Blockchain-based architecture for IoT significantly reduces the overheads of classic Blockchain while preserving the bulk of its security and privacy benefits. The architecture supports the creation of an overlay network by high resource devices to employ a publicly accessible distributed Blockchain that guarantees end-to-end security and privacy. Moreover, it reduces the time required to process block validation using distributed trust to provide effective security and privacy for IoT applications. However, the creation of this scalable blockchain and its related security certificates were not provided.

However, some of these works lack of real implementation and are based only in theory or simulation. Other still have limitations regards to communication and computation cost. Conversely, our work focus in developing and implementing an architecture which integrated the access control scheme within two smart contracts deployed in multi edge servers to achieve a secure distributed blockchain to serve smart

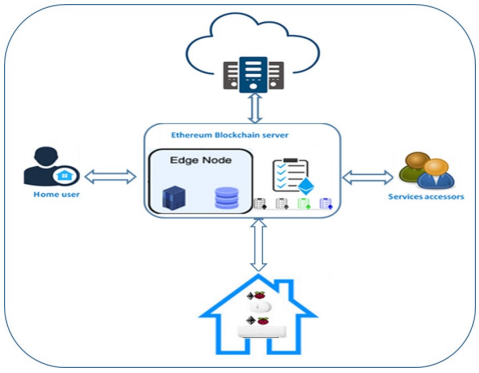


Fig. 1. Proposed system architecture

home IoT devices. The use of multi-edge servers as an admin provides a complimentary way to overcome the computation cost and single point of failure. We also investigate one of the popular blockchain technology, Ethereum smart contract and ERC-20 token generation for implementing a real smart home scenario.

IV. Proposed Attribute Based Access Control Scheme for Smart Home

This section describes the key architecture and design details of our proposed Blockchain-based system, in which Ethereum smart contracts are used to register, and manage Home user, IoT smart home devices and edge server.

A. System Architecture

The proposed system architecture is shown in Fig.1 The architecture is composed of four main participants with access to Ethereum smart contracts through the Internet: end users (home users, services accessors), IoT smart home devices, edge servers, and the cloud servers hosting IoT data, (transactions between the edge servers and the cloud will not be covered in this paper). IoT smart home devices do have a unique Ethereum addresses (with public and private keys). All other participants have unique Ethereum Address (EA) and interface directly with the smart contract through an Ethereum client in the case of edge and cloud nodes, or through a front-end application/wallet in the case of the end users.

The following summarizes the key role of the different system participants:

1) End user: Request access permission through the smart contract to access a certain smart home devices.

a: Home user: User device is a device (e.g., PCs, laptops, smart phones) through which users can enjoy the services (e.g., checking the current temperature of his/her own house) provided by the servers.

b: Service accessors: any service providers such as health care, police or other parties who need to access the smart home data to provide any type of services.

2) IoT devices: The IoT devices in the system mainly include sensors, which can perceive environmental data (e.g., temperature) and send these data to the edge servers or storage devices for further use, and actuators, which can perform some operations (e.g., turning on the air conditioner) once receiving a command from users.

3) Smart home multi-edge servers (Admin Edge): An Edge node is a device or a cluster of devices that can interact with the IoT devices and storage devices to provide a variety of services. Interactions between the servers and other peers (e.g., IoT devices, storage devices) include collecting environmental data from the sensors, sending commands to the actuators to perform some operation, querying data from or storing data to the storage devices. Edge nodes process all incoming and outgoing transactions and uses a shared key for local communications with IoT devices and local storage. It maintains the smart contracts that mange registering the end users and IoT devices, authenticate end users to access the IoT devices. The mining work is only done by the edge servers that have more resources than the IoT devices. It is never done by the resource constrained IoT devices.

4) Cloud: Provide long-term data analytics and storage. The resources in the cloud can also be configured as nodes on blockchain to ensure privacy and integrity of data in the system.

B. Attribute based access control and Smart contracts

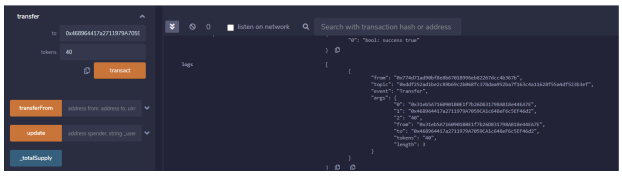
To avoid complexity of one smart contract, the proposed framework consists of two Ethereum smart contracts, namely the Register contract and Access contract. The first contract responsible for storing and managing (e.g., updating, adding, deleting) the subject and object attributes, and policies. The Access contracts responsible for control the access of the IoT devices by generating ERC-20 token and finalise the permission to access the IoT devices. Each smart contract is introduced as follows:

1) Register contract: The policy is deployed on the blockchain to register and manage the attribute of users, IoT devices. Only the administrator has the permission to execute this contract. Each users and IoT devices has a unique identifiers (Ethereum account address) and multiple attributes associated with its ID. This contract has functions of managing subject and object attributes such as adding, deleting and updating. Also, this contract specify the policy associated with each user and IoT devices based on user type. A policy is a statement that combine a set of Subjects (users), a set of Objects (IoT devices) and a set of Action to state that this user can perform the action in the IoT devices. Example of policy is shown in Table 1.

2) Access Contract: This contract controls the access request from the users (subject) to the IoT devices (object) in the system. This contract is executed by the user to request a token to be able to

TABLE I
EXAMPLE OF USER ATTRIBUTES, IOT ATTRIBUTES AND PERMISSIONS

User attributes	IoT Device attributes	Action
UserAddress	IoTAddress	Execute
UserType	IoTName	Read
UserName	IoTFun	write



(a) Transfer function

(b) Approve function

(c) Token balance

Fig. 2. Example of Access contract functions

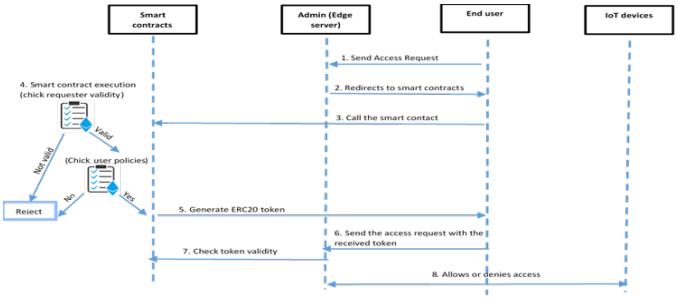


Fig. 3. Typical transactions in proposed scheme

(a) The user with a valid token will only be permitted to check the value of the sensor

(b) User without enough token or an unregistered user requests for checking the temperature.

Fig. 4. User request for room temperature data

communicate with an object. This contract contains function to check the validation of subject attributes and also check policy, based on the received policy the AC determines if the subject has rights to perform action on the object then send a token to the subject. The main functions of the contract are Check attribute (), Get policy() and TransferToken(). This contract is also responsible of ERC-20 token generation. Fig.2 shows example of some Access contract functions. Each user has certain amount of valid tokens at a time based on user type to prevent a valid user from flooding the network with access control requests.

C. System design

The proposed system provides authentication for users using an attribute access contract and token distribution. Fig.3 illustrates typical attribute based access contract transactions with this authentication mechanism. Users can remotely access or control home devices using the freshly generation token that only the requester is able to receive the response from the legitimate home admin. We will now describe four phases in our system: Initialization, Request Control, State Delivery and Chain Transaction.

1) Initialization: For illustration purpose, we assume that family members constitute a group of users, from which a group admin is chosen. An admin invokes the Register Contracts to add other users and IoT devices. Users allocate their Ethereum Address (EA) and individual private keys for signing transactions. Correspondingly, each home admin holds the group public key for the transaction verification. The admin is run in different miners in multiple edge nodes to avoid single point of failure.

2) Request Access: When a user wishes to publish an access or control request with the home admin, a token is generated for a certain duration and exact access time. This is the suggested approach to avoid replay attacks and profiling. After obtaining the token by invoking the TransferToken () from Access Contract, the user constructs the transaction from his/her requirement. For example, the user requests to get the room temperature, the transaction computed after the user redirected to the smart contract and request token, three main functions will invoke in that contract Check attribute(), Get policy() TransferToken(). The user will send the received valid token with the request access to the admin, if the user has a valid token then the access will be granted. Fig.4 demonstrates different output of valid and invalid user request for room temperature.

3) State Delivery: The home Admin monitors the smart contract for new requests. Once a user requests a new access or services. If the transaction passes the verification then the home admin checks the token validity then grant or deny access to the IoT device.

4) Chain Transaction: Admin nodes (miners) are responsible for retrieving transactions in the smart contract and compete with each other to be the first to successfully solve the PoW for chaining the block to the blockchain. Once successfully solving the PoW, the miner broadcasts its solution to the blockchain network to reach consensus. The first miner to successfully mine a block that reaches the consensus earns the mining reward.

D. Implementation

The detailed hardware and software configurations are as follows. Our system is developed on a private network. The model runs in Private Ethereum Network which consists of one laptop device (Dell XPS) to simulate the edge server running two miners connected to two single-board computer (Raspberry pi 3 Model B) which are simulating temperature sensor and LED and one home user laptop. The edge server is equipped with 4 independent CPU cores and 16 GB RAM. The mining environment is set up using one core and the rest of the processor cores are reserved for the edge computing service. The miner can boost up to 3.5 GHz CPU, 8 GB RAM, and 1 TB storage. As the IoT devices, two Raspberry Pi has 1.2 GHz CPU, 1 GB RAM, and 32 GB storage with accessory modules including temperature sensor and LED sensor. The laptop as a home user has 2.2 GHz CPU, 16 GB RAM, and 256 GB storage.

The edge server has installs Go-ethereum as the blockchain running framework and Solidity as the smart contract development language. Remix integrated development (IDE) is used to write and compile the contracts (Remix 2020). This uses Solidity as the language to write smart contracts. Web3.js (Ethereum JavaScript API) is also used in the model to deploy and compile the contracts and to monitor the contract state. JavaScript is used to interact with the corresponding geth client via the HTTP connection. A simple html web page is built to support the interaction between the home users and the devices. The Raspberry Pis have been installed with Raspbian operating system and Go-ethereum to work in the light mode without block mining function. The home user laptop is with windows 10 home (64bit).

In the testbed, the first laptop works supporting two edge service provider and a block miner solving PoW puzzle. The Raspberry Pis and the second laptop act as blockchain clients generating and sending transactions of resource requests to the edge server. Given the above installations, the edge server works as a “full” blockchain node which stores all the transactions, executes the predefined smart contracts and mines new blocks. The IoT devices work as “light” blockchain nodes which only store the transactions data.

The private blockchain is configured following a series of steps, including the selection of a compatible version of Ethereum, the use of Windows power shell to initiate geth, and the requirement for each node to satisfy multiple requirements before being able to join the blockchain. This includes; (1) initialisation of the genesis file (Test.json) to create the first block, (2) use of network ID to connect to the same blockchain, (3) initialisation of the private blockchain using a geth command. An account with a private and public key is created by the miner for each node and indexed according to its address, which it can interact with other nodes and smart contracts. The geth on each node is then started using a command which includes different flags for different functions. All nodes are set “no discovery” flag, so they cannot connect to other peers without explicit

addresses and that secures the nodes from being hooked by external attackers. A specific command is then used to retrieve the node ID to allow syncing to occur. This last step is repeated to add the two Raspberry Pi as nodes and the home user laptop to create a private blockchain with fully synchronized nodes.

The smart contracts specify various permissions to different devices based on user type, where the edge server owns higher authority to access all the functions but other users and the IoT devices are only limited to some functions. Such a setting reduces the impact even if one user or some weak devices are hacked to perform malicious activities.

V. Evaluation and analysis

This section provides a complete discussion on the security and performance of the Attribute-smart contract based edge scheme. In this section, we briefly define possible threats and attacks and then discuss handling techniques to ensure satisfying the security goals of the 'CIA triad'; namely, confidentiality, integrity and availability. Authentication and access control are provided in our architecture to address these goals.

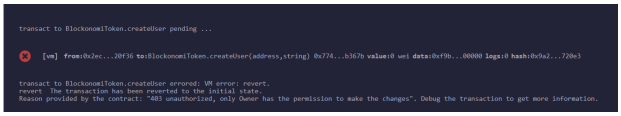
A. Security analysis

Confidentiality aims to ensure unauthorised users are prevented from gaining access to IoT devices and its data and making sure that private data is delivered only to the intended users. The common approach for achieving confidentiality is by performing message encryption and decryption using secure SSL session upon successful user authentication [8]. Our Blockchain-based design relieves the use of the Public Key Infrastructure (PKI) for key distribution. A unique 20-byte Ethereum Addresses (EA) is assigned instantly to any node (including IOT devices) with almost no collision–which is a powerful feature of blockchain. EA comes with asymmetric public key pairs, which can be subsequently used for establishing secure SSL session for communication between the authenticated user and the IoT device. During the formation of the private network the miner distributes private and public keys associated with EA for each node. The temperature sensor or the LED, as the sender node, utilises the private key to provide a digital signature allowing the requested transaction to be broadcast across the entire network.

In terms of availability, our architecture leverages the inherent properties of the block chain technology which offer reliability and robustness. Because of the decentralised structure of the blockchain and the ledger replication in multiple locations, there is no possibility for a single point of failure and that all data is circulated via multiple nodes. A copy of the transaction history is stored in each admin node, enabling it to be verified and linked back to the initial transaction. Moreover, to increase smart home availability, IoT devices are protected from malicious requests by limiting the accepted transactions to those users who have a valid token. So every transactions received are authorised by the admins before forwarding it to the IoT devices.

Furthermore, the use of valid Token increases the level of security in our architecture. That can be observed as only the admins can issue valid token and only the intended user can use that Token. Fig.5 shows the revert error when anyone other than the admin try to create a user or issue token. Also, token's owner cannot transfer the token to any other users, so if the public key of a user is compromised, the smart contract construction prevent token transfer. The admin will allow only transaction that has a valid Token associated with a valid user to be accepted in the network.

1) Denial of service (DOS) Attack: In this attack, the attacker sends a large number of transactions to target in order to disrupt its availability. The use of attribute based access control smart contracts in our architecture reduce the effect of this attack since only authorized transactions would be accepted. The admin has to examine the address and policy for each user and devices to issue valid Token to send a transaction. If the admin receives several unsuccessful access requests from an unauthorized entity, it can block that transaction and reject it. Furthermore, the smart contracts enforce the policy automatically. If some IoT devices were compromised and controlled by hackers for malicious activities, such as making continuous resource requests, or initiating denial of service attacks, the smart contracts will execute automatically based on the preprogramed policies of the token total supply, the access time and duration.



(a) Invalid user requesting create a new user

(b) Invalid user requesting for token

Fig. 5. Revert transaction

2) Modification attack: In this attack, attacker may try to alter or delete stored data of particular user or device. To launch this attack, the attacker has to compromise the local storage security. However, in our scenario, only the admin has the rights to store, delete or update date based on the policy in the smart contracts. If any attacker tries to modify the block, the change will be detected since every block contains its previous hash block and change in one block will result in a break in the chain.

The next class of threat is against authentication and access control. It has been claimed by [4] that, it is possible for an attacker to take control of a smart home device or introduce a fake device to a home network. Our design employs a hierarchical defence mechanism against these attacks. First, there is an admin node which control all incoming and outgoing transactions and prevents smart home devices from being directly accessed from the Internet. If the admin detects a transaction that does not follow the policies defined by contract, the transaction is dropped.

The second defence is that all devices in the home are required to have a unique address and follow same genesis transaction in the local blockchain that allows them to initiate communication with the admin and other devices. A device without a corresponding address and genesis transaction is isolated from the network. This prevents an attacker from introducing unauthorized devices to the network.

B. Performance analysis

To evaluate the performance of proposed model, we conduct experiments in private Ethereum network where the edge server represents the home admin to add home user, and the two sensors (temperature and LED). The home user requests room temperature to turn on/off the AC (change the state of LED) based on temperature. The admin checks the user validity and then give the access to the user as described previously in system design section. We simulate two types of transactions in a smart-home setting i.e. store and access. Here we investigate the store transaction (adding new user or IoT devices using the register contract) and the request access transaction to invoke some data (using access contract). We evaluate the block size, gas cost and time cost by comparing our scheme with the works in [18], [19] and [20].

1) Block size: Ethereum's block size is based on complexity of contracts being run and the number of transactions known as a Gas limit per block, and the maximum can vary slightly from block to block. Depending on how much gas each transaction spends, transactions are combined and shaped into form of blocks. We

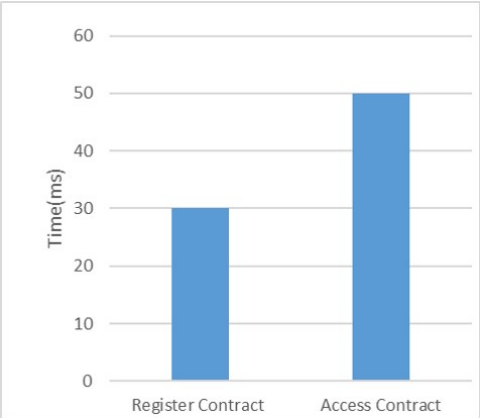


Fig. 6. Time to complete one transaction

investigate the number of transactions per block. We find that, 1MB block contains 280 store and 300 access transactions. The sizes calculated are 2.80KB for store and 4.00KB for access transactions. The average size of a block is 130KB and each block can store up to 200 user or device registrations.

Since, the size of block is the key factor that impacts the overall latency, in our experiment, we find block size varies between 118 KB to 145 kB based on the contract being execute. We evaluate the interaction delay of register contract and access contract which is significant to guarantee system efficiency. Fig. 6 shows the completion of one transaction is less than 30ms in the Register contract and 50ms for the Access contract. Such delay should satisfy the latency requirement of the real-time applications.

However, the latency gets worse with register contract as the block size is increased. The latency increase due to the increased time needed to include the transaction in the block and the increased bandwidth required to propagate a bigger block in the network. However, because the access provided by the edge server has more computing and bandwidth resource, it completes the validation and the transmission of the new blocks faster. The latency is fewer compared to the IoT-BC proposed scheme in [18]. IoT-BC is based on Fabric architecture which in general has larger transaction size because they carry the certificate information for approval. As a result, the total increase in transaction latency in IoT-BC is 22.45% while in our scheme it is around 20.23%.

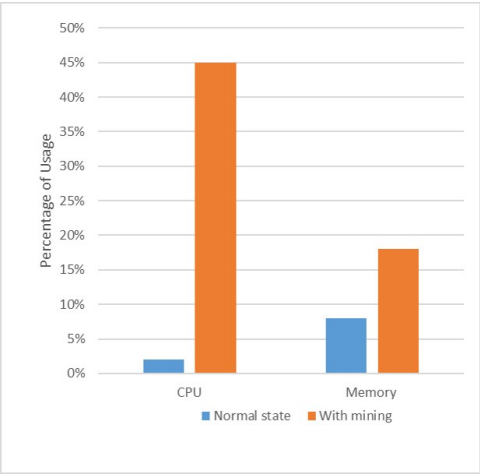


Fig. 7. Resource usage for single transaction

TABLE II
CALCULATED GAS COST

	Proposed Scheme	Scheme in [20]	Scheme in[19]
AddUser	85,662	-	152,863
AddPolicy	360,273	128,777	363,964
DeployACC	1,377,071	1,706,290	1,301,972

The CPU and memory usage are also explored as illustrated in Fig.7. We observe that the regular transactions take a very low percentage of CPU resource while the memory usage is little higher since the blockchain client occupies 8% even in normal state time.

However, we note that in a real smart home environment, the number of IoT devices connected will be increased and that will have possible impact to the blockchain overhead. Since the miner is located at the edge server, mining, verifying and storing new blocks will increase the computing resources usage. Therefore, specifying the number of IoT devices to be managed by one edge server, or launching more VM as the miners to share the load of computation are recommended idea.

2) Gas cost: The deployment of smart contracts on the blockchain and execution of these contracts ABIs require fee to be paid to the miner which mines the block. Ethereum utilizes a unit called gas to measure the amount that functions needed to perform a task, e.g., deploying a smart contract or executing an ABI. In general, more gas is consumed with more complex task. Gas has price that varies with time. Thus, the fee needed to be paid for performing a task is the product of the amount of consumed gas and the gas price. Table II lists the amount of gas paid for some functions, like adding a subject/object or policy, deploying the AC and executing the AC.

In our proposed scheme, the amount of gas required for deploying the access contract is 1,377,071, which is more than the existing schemes compared here. We can see from the table that the proposed ABAC framework in [19] consumes less gas than our scheme. This increase value is due to the relatively complex interactions in our scheme for retrieving attributes and policies between the Access contract and Admin policy smart contract and Authority contract. However, in [20] one ACC is deployed for only one subject-object pair. The gas cost increases linearly as the number of subject-object pairs of the system increases. While, in our proposed system there is no need to deploy a new Access contract when the subject and object increase. This results in less gas consumed and hence, less cost. Moreover, when comparing the gas cost for performing functions such as Add user or add policy, our proposed scheme consumes less gas for the same functions in scheme [19].

3) Time cost: the approximate time cost of executing the Access Contract is 40 seconds in our proposal which is more than 36 seconds average time for ABAC shown in [19]. This is due to the time cost of invoking token in our proposal and the extra time needed to check token validity and call other smart contracts. However, the fresh one-time token generated during each Access request is used for securing the session and this ensures data confidentiality which worth the few seconds difference.

Note that the execution time of the ABI varies depending on various factors such as the system's computing power, network architecture, timing of mining, etc., so the execution time may differ within different Ethereum network.

VI. CONCLUSION

This paper evaluates a real-time interaction model between home users and a fully validating private blockchain node through the use of attribute based access control to authenticate smart home users and IoT devices. By combining the blockchain technology with attribute based access control and edge computing, this model solves the problem of the traditional access control method which is based

on the centralized design to meet the access control requirements in IoT. In this paper, we develop Ethereum blockchain, multiple smart contract and the implementation is described to demonstrate the feasibility of the framework. Compared with existing scheme, our proposed scheme achieves more fine-grained access control with freshly token generation and less computing cost with edge computing. Our framework also achieves desired security goals and is resilience against modification and DoS attacks. Our work is ongoing research and we are currently working on the secured transaction flow from the edge node to the storage device in the cloud. In future, we propose to apply differential privacy to enhance the privacy aspects of our model.

REFERENCES

[1] M. Moniruzzaman, S. Khezr, A. Yassine, and R. Benlamri, "Blockchain for smart homes: Review of current trends and research challenges," *Computers & Electrical Engineering*, vol. 83, p. 106585, 2020.

[2] Y. Nakamura, Y. Zhang, M. Sasabe, and S. Kasahara, "Capability-based access control for the internet of things: An ethereum blockchain-based scheme," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[3] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.

[4] W. Ejaz and A. Anpalagan, *Internet of things for smart cities: technologies, big data and security*. Springer, 2019.

[5] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for iot security and privacy: The case study of a smart home," in *2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops)*. IEEE, 2017, pp. 618–623.

[6] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1189–1205, 2013.

[7] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *Journal of Network and Computer Applications*, vol. 126, pp. 45–58, 2019.

[8] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A user authentication scheme of iot devices using blockchain-enabled fog nodes," in *2018 IEEE/ACS 15th international conference on computer systems and applications (AICCSA)*. IEEE, 2018, pp. 1–8.

[9] H. Guo, E. Meamari, and C.-C. Shen, "Multi-authority attribute-based access control with smart contract," in *Proceedings of the 2019 International Conference on Blockchain Technology*, 2019, pp. 6–11.

[10] V. Buterin and F. Vogelsteller, "Erc20 token standard," *URL: https://theethereum. wiki/w/index. php/ERC20 Token Standard*, 2015.

[11] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[12] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE access*, vol. 6, pp. 6900–6919, 2017.

[13] P. K. Singh, R. Singh, S. K. Nandi, and S. Nandi, "Managing smart home appliances with proof of authority and blockchain," in *International Conference on Innovations for Community Services*. Springer, 2019, pp. 221–232.

[14] Y. Lee, S. Rathore, J. H. Park, and J. H. Park, "A blockchain-based smart home gateway architecture for preventing data forgery," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, pp. 1–14, 2020.

[15] T. L. N. Dang and M. S. Nguyen, "An approach to data privacy in smart home using blockchain technology," in *2018 International Conference on Advanced Computing and Applications (ACOMP)*. IEEE, 2018, pp. 58–64.

[16] J. Xue, C. Xu, and Y. Zhang, "Private blockchain-based secure access control for smart home systems." *KSII Transactions on Internet & Information Systems*, vol. 12, no. 12, 2018.

[17] M. A. Rahman, M. Rashid, S. Barnes, M. S. Hossain, E. Hassanain, and M. Guizani, "An iot and blockchain-based multi-sensory in-home quality of life framework for cancer patients," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 2116–2121.

[18] J. Ali, T. Ali, S. Musa, and A. Zahrani, "Towards secure iot communication with smart contracts in a blockchain infrastructure," *arXiv preprint arXiv:2001.01837*, 2020.

[19] M. Yutaka, Y. Zhang, M. Sasabe, and S. Kasahara, "Using ethereum blockchain for distributed attribute-based access control in the internet of things," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[20] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart contract-based access control for the internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2018.