Received May 31, 2021, accepted July 10, 2021, date of publication July 20, 2021, date of current version July 28, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098795

Privacy-Preserving Mechanism in Smart Home Using Blockchain

AMJAD QASHLAN[1], PRIYADARSI NANDA[1], (Senior Member, IEEE), XIANGJIAN HE[1], (Senior Member, IEEE), AND MANORANJAN MOHANTY[2]

[1]Faculty of Engineering and IT (FEIT), University of Technology Sydney (UTS), Sydney, NSW 2007, Australia
[2]Faculty of Science, University of Technology Sydney (UTS), Sydney, NSW 2007, Australia

Corresponding author: Amjad Qashlan (amjad.qashlan@student.uts.edu.au)

ABSTRACT The IoT, or Internet of Things has been a major talking point amongst technology enthusiasts in recent years. The internet of thing (IoT) has been emerged and evolved rapidly, making the world's fabric around us smarter and more responsive. The smart home uses one such transformation of IoT, which seems to be the wave of the future. However, with the increasing wide adoption of IoT, data security, and privacy concerns about how our data is collected and shared with others, has also risen. To solve these challenges, an approach to data privacy and security in a smart home using blockchain technology is proposed in this paper. We propose authentication scheme that combines attribute-based access control with smart contracts and edge computing to create a secure framework for IoT devices in smart home systems. The edge server adds scalability to the system by offloading heavy processing activities and using a differential privacy method to aggregate data to the cloud securely and privately. We present several aspects of testing and implementing smart contracts, the differential private stochastic gradient descent algorithm, and system architecture and design. We demonstrate the efficacy of our proposed system by fully examining its security and privacy goals in terms of confidentiality, integrity, and availability. Our framework achieves desired security and privacy goals and is resilient against modification, DoS attacks, data mining and linkage attacks. Finally, we undertake a performance evaluation to demonstrate the proposed scheme's feasibility and efficiency.

INDEX TERMS Blockchain, smart home, access control, smart contract, differential privacy, cyber threats.

I. INTRODUCTION

Technologies have made it possible for residence building with integrated Internet of Things (IoT) network offering increased comfort, security and quality of life. As such, an IoT infrastructure underpins a smart home network, which connects various smart devices (such as smartphones, smart metres, wearable gadgets, and so on). People's ability to live independently can be enhanced and enabled by smart home technology. They include a variety of useful technologies, such as those for monitoring and assessing health, which appeals to both users and device manufacturers. The value of the worldwide smart home market is expected to hit $53 billion by 2022, which is not surprising. This prediction is based on a nearly 21 percent annual growth rate predicted for the market from 2018 to 2022 [1]. Although the advantages of smart homes to homeowners and stakeholders are well documented, there are a number of concerns to be aware of, including cyber-attacks and threats to user data security and privacy [2].

Traditional techniques dealing with such threats rely on centralised structures that are vulnerable to cyberattacks [3]. As a result, the access control function is critical for preventing unauthorised users from accessing resources by explicit or implicit requirements and only allowing authorised parties access to resources. Traditionally, access controls have been handled by centralised systems that are relatively easy to operate [4]. This means that all access restrictions, such as assigning access privileges, managing access (e.g. updates, revocations) and access verification, are handled by a central server. However, there is a risk that the server may fail as a result of 'natural' (functional) or external factors (cyber-attack), compromise the access control mechanism. Furthermore, IoT systems' massive scale and distributed

The associate editor coordinating the review of this manuscript and approving it for publication was Md Zakirul Alam Bhuiyan.

nature create challenges for centralised strategies to regulate requests for access to the desired resource [4].

Some of the limitations of centralised networks can be overcome by distributed access control networks. Rather than employing a single server, these networks use several nodes to handle access control activities. To provide trustworthy and dependable access controls that can withstand malicious attacks, the nodes 'agree' on the rights to be assigned, the policies to provide access, and the verification results. As a result, there is an interest in using emerging blockchain technology for distributed and reliable access control.

The emerge of distributed and tamper-resistant ledger-based blockchain techniques to protect data has opened up new possibilities for smart home data privacy, security, and integrity challenges. Blockchain is made up of a digital ledger that records and shares transaction information in the network. Each user has access to secure, cryptographic public and private keys in order to interact with the system. One user can initiate the transaction with his keys, and the other users in the network can accept it with their own keys. Once the nodes agree that the originating user possesses the data they claim, the transaction will be accepted, else, it is rejected [5].

Blockchain technology has proven to be effective in a variety of smart home applications, including control over access to the home, exchanging data, and so on. The use of blockchain in smart home networks is also justified because it works independently of current heterogeneous protocols commonly used in smart homes (such as Z-Wave, Zigbee, Bluetooth, and Thread) [6]. Nonetheless, using blockchain directly in a smart home is always a challenge due to the high level of resources consumed during mining and consensus procedures as well as the limitations of node resources in smart home devices.

In turn, Edge computing provides an alternative and complementary technique for managing proof-of-work (PoW) challenges while also supporting blockchain applications in the smart home. Edge computing extends the spread of cloud-based resources and services by performing computation at the network's extremes (edges). It has a multi-access system that allows users to access cloud-like services for better computing, apps, and storage. As a result, resource-constrained smart home appliances may be able to expand their computational capabilities by outsourcing mining and storage to specified edge servers. Hence, the combination of blockchain with edge computing creates a decentralised system for outsourcing computation and storage security for scalable and safety proof operations [7].

While blockchain is regarded as the future of data storage due to its decentralized structure, several issues are still to be resolved before it is implemented in daily life scenarios. A significant parameter in blockchain applications that needs further development is data preservation and transaction privacy. Blockchain user identification across the decentralized network is supported by the public key. As a result, all identities do not remain private or anonymous. An adversary in the role of a third-party may analyse the transactions on the network and potentially infer the identities of other users. In addition, blockchain's decentralized structure allows unprotected blockchain scenarios to be observed. Moreover, additional privacy features are needed to better protect personal data on the blockchain nodes. With financial blockchain systems for instance, the transaction details are broadcasted across the decentralized network whenever a transaction takes place [8]. This broadcasting occurs to safeguard each blockchain node with up-to-date information. Furthermore, the ledger recording the transaction remains uniform across the network. An adversary may use this information to monitor an individual and go back through the transaction details to discover transaction information. Moreover, with regards to blockchain-based IoT devices, an adversary may compromise the information exchange between devices for illegal purposes.

Furthermore, there are also privacy risks associated applying blockchain in other sectors such as financial, real estate, and asset management [9]. That is, blockchain's distributed nature means that the individual's identity or personal information may be leaked during transactions. To date, literature in the field on how to preserve the individual's privacy in blockchain has mostly focused on anonymization strategies and their derivatives [10]. However, Studies show [11] that, anonymization cannot ensure total privacy because of the potential to combine anonymized data with similar datasets to discover personal information.

To overcome the above mentioned issues and provide privacy protections, it may be useful to integrate differential privacy based on machine learning with the use of the latest blockchain technology. Differential privacy is efficient at preserving privacy in statistical databases and real-time settings [12]. Differential privacy is an approach to preserve the confidentiality of data without risking its leakage by adding noise to data without influencing the correct output of the data analysis result.

The use of differential privacy can create a level of indistinguishability in statistical blockchain data, leaving the analyst unable to predict with any certainty the accessibility of individual blockchain nodes. Differential privacy is a good fit to be used in blockchain technology in order to preserve the individual's identity during a broadcast. While ensuring that the information remains useful for completing transactions, differential privacy can still perturb the person's identity to the network and an adversary will not be able to determine the sender's or receiver's actual identity. Thus, differential privacy can help to keep private sensitive/personal information in a dataset. Therefore, differential privacy in blockchain applications may prove to be beneficial to protect data privacy [12].

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, smart-edge home architecture. In our framework, every single home has multi edge servers as local blockchain miners and the smart contracts are utilized to apply the policies and rules in an automated manner and regulate the smart home IoT devices based on the Attribute Based Access Control (ABAC) scheme. The edge servers aggregate data from the IoT smart home devices to the cloud server for further storage and analysis after applying a differential privacy mechanism and providing a privacy preservation system.

In this paper, we extend our earlier work published in [13] and expand the functional capabilities of our architecture by adding differential privacy as a scheme to preserve privacy of users. Hence, this paper presents a novel architecture involving authentication scheme based on Ethereum smart contract [13], integrated edge computing and differential privacy enhancement model. The main contributions in our paper are based on the following:

- We design a privacy-preserving and secure decentralized Stochastic Gradient Descent (SGD) algorithm using blockchain.
- We apply machine learning on the differential privacy mechanism to send data from private smart home to the cloud.
- We present detailed analysis on our proposed scheme and show how the proposed model can defend against traffic analyses and data mining based attacks such as linkage attacks.
- Complete design of the Ethereum smart contract including implementation and testing scenarios are presented to validate our proposed scheme.
- Performance evaluation of our proposed scheme is presented by comparing them with existing models with respect to various performance metrics.
- Security analysis of our proposed scheme using threat model to overcome Denial of Service (DoS) attack scheme is presented by determining the efficiency of our proposed model.

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

II. RESEARCH BACKGROUND

This section provides the background information needed to understand the proposed framework. It discusses the key concepts of smart home, access control, blockchain technology, Ethereum with smart contracts, ERC 20 token, Edge computing and differential privacy that set the stage for the rest of this paper.

A. SMART HOME

Despite countless publications attempting to define the criteria for a standard definition of smart home, there is still no consensus on what represents a smart home. The term ''smart home'' is commonly used to describe a place of residence with technology capabilities to enable task automation, monitoring of people and activities, and health-maintenance mechanisms. All elements in a smart home can communicate with one another across a network and can be controlled both locally (from within the home) and remotely (through the Internet). Given its wide range of applications, this type of system has a lot of promise to improve security, provide a more energy-efficient alternative, and promote user comfort [9]. In this paper, we use a holistic definition of a smart home; one that uses Internet-connected devices to serve a variety of functions. Smart TVs, smart temperature controls, smart hubs, and other connected devices are examples of smart home gadgets. Typically, companies that provide IoT devices for smart homes need access to their interface in order to control the devices. As a result, smart homes with several devices from various manufacturers may have several disconnected interfaces, necessitating well-defined device management. Because most of the device's resources are used to perform other functions, IoT devices lack the resources to carry out security actions [14]. Hence, a security mechanism integrates the necessary processes to address current IoT concerns without utilizing significant resources.

B. ACCESS CONTROL SCHEME

Access control systems are usually based on access control lists (ACLs), which provide users access permissions. When there is an increase in the number of users seeking resources, ACLs become more difficult to govern. As a solution to this limitation of ACL systems, designers have created Role Based Access Control (RBAC) systems, [15] which add an intermediate layer to the process of distributing role permissions rather than giving them directly to users and then assigning them their roles. This strategy can considerably reduce the time and effort required to monitor access control rules. This is even when the number of subject roles and resources are increased, or when the system contains many administrative fields. Attribute Based Access Control (ABAC) systems attempt to address the issues associated with increase in the number of roles by allowing users to apply the subject's attributes directly, as well as resource and environmental properties. This can be done to describe the access policies and, as a result, reduce the number of rules or rule updates. On the other hand, ABAC still requires to access a consistent description of the field attribute and the definition of attributes across many fields. [16].

Goyal *et al.* [17] demonstrate the applicability of Attribute-Based Encryption to share of audit-log information and broadcast encryption. In their scenario, the data is stored on the server in an encrypted form while different users are still allowed to decrypt different pieces of data according to their security policy. This effectively eliminates the need to rely on the storage server for preventing unauthorized data access. Moreover, Hu *et al.* [18] publish a guide to Attribute Access Control with a definition of ABAC and

149 descriptions on the functional components of ABAC. Also, the guide provides planning, design, implementation, and operational considerations for employing ABAC within a large enterprise with the goals of improving information sharing while maintaining control of that information. Furthermore, attribute based access control has been used in blockchain architecture. Zhu *et al.* [19] present a new digital asset management platform, called DAM-Chain, with Transaction-based Access Control (TBAC) which integrates the distribution ABAC model and the blockchain technology. They take transaction as a bridge to integrate ABAC and blockchain into a new platform for resource distribution and sharing. They claimed that their proposed platform supports flexible and diverse permission management as well as, verifiable and transparent access authorization process in blockchain based architecture.

Also, Rouhani *et al.* [20], propose a distributed Attribute-Based Access Control (ABAC) system based on blockchain to provide trusted auditing of access attempts. Besides auditability, the system presents a level of transparency that both access requestors and resource owners can benefit from it. They present a system architecture with an implementation based on Hyperledger Fabric, achieving high efficiency and low computational overhead. They validated their solution through a decentralized access control management application in digital libraries.

This paper examines attribute based access control in particular, because, it is deemed to be an appropriate decentralised model for IoT setup and provides scalability, flexible and strong dynamics. Our access control scheme is different from [21], in which the authors used three types of access control procedures; device-to-device (D2D) access control, device-to-user (D2U) access control, and device-to-fog server (D2FS) access control to authenticate users in (internet of Everything)IoE. Our access control is based on different policies which combines a set of subjects (users), a set of Objects (IoT devices) and a set of Actions to state that this user can perform the action in the IoT device. The policy is invoked whenever there is an access request from any user or device in the network using the smart contract. Moreover, we integrate the token mechanism to further finalize the permission to access the IoT devices. The smart contract checks the policies and then tracks the token amount to 'who owns' and 'how much' of that particular token to access certain IoT device.

C. BLOCKCHAIN TECHNOLOGY

Blockchain is defined as a decentralised, distributed, and immutable ledger that maintains a record of assets and transactions on a peer-to-peer (P2P) network [8]. Thousands of network-based mining nodes register and validate each transaction digitally in the blockchain. All the transactions are stored and organised in 'blocks' using timestamps. Several blocks are then linked together to form a 'blockchain'. To ensure the authentication and integrity of the data, the blockchain uses elliptic curve cryptography (ECC) and SHA-2 hashing technique for robust cryptographic proof. Bitcoin is a well-known example of blockchain infrastructure. In general, the blockchain architecture that supports Bitcoin is the same architecture that powers most cryptocurrencies. In turn, the Ethereum blockchain's growth and use of smart contracts result in infinite number of cryptocurrencies [22].

D. ETHEREUM WITH SMART CONTRACT

Ethereum includes smart contract features as part of its decentralised platform. The Ethereum smart contract, invented by Vitalik Buterin in 2013, supports event-directed, turing complete scripting functionalities for verifying and processing complex transactions to demonstrate the contract's validity [23].) In terms of the smart contract, it works similarly to an event-directed script in that it executes the script automatically once the pre-defined criteria are met. All relevant functions and processes must be in place before the smart contract can be executed [24]. Externally Owned Accounts (EOA) and Contract Accounts are the two types of accounts in Ethereum. Each account type has its own unique address, which is a 20-byte hexadecimal string. The EOA, which includes an ether balance, is controlled by the owner's private key, which also transmits transactions (for example, sending a message to prompt the initiation of a smart contract). An EOA does not have a code assigned to it. On the other hand, a contract account with an ether balance also has a related code that is activated by another smart contract or a transaction.

E. ERC-20 TOKEN

ERC-20 stands for "Ethereum Request For Comments," and the number 20 serves as a unique identifier to differentiate from the other standards. It is a protocol that defines a set of standards and rules for token issues on the Ethereum network and is used to create blueprints for smart contracts based on Ethereum. As a technical standard, ERC-20 has become one of the most important and widely used tokens for all smart contracts on the Ethereum blockchain [25]. ERC-20 defines a set of six functionalities within the Ethereum system for the benefit of other tokens.

1) totalSupply (): to figure out how many tokens were created and exist in the system.
2) balanceOf (address owner): to returns the number of tokens in an account for a given address.
3) allowance (address tokenowner, address spender): The user's balance is one of the most critical data needed to complete a transaction. To carry out a transaction, the user must have a certain number of tokens. If the user does not have the required number of tokens, the allowance () function is used to cancel the transaction.
4) approve (address spender, unit tokens): The contract owner allows collecting the required amount of tokens from the contract's address once the user has the required amount of tokens for a transaction and the balance has been checked. By comparing the

transaction to the total token supply, this function ensures that there are no additional or missing tokens.

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): The contract owner can transmit tokens using the transfer() function. It allows the contract owner to send token amounts to different addresses. Also it allows a certain number of tokens to be transferred from the overall supply to a user account.

F. EDGE COMPUTING

The ability of cloud computing to provide limitless processing, data storage, and systems administration resources has led to the development of many cloud-based apps and the rapid expansion of Internet-based corporations such as Amazon in recent years. The trend recently has been to move cloud functions to network edges [26]. This is dependent on delay-sensitive applications (for example, virtual reality) with strict delay requirements. Edge computing has put more strain on cloud resources and services in order to provide mobility, location detection, and lower latency. As a result of these benefits, network edge technology is critical to realise the future IoT [27].

The edge computing structure has three levels: end device (front-end), edge server (near-end), and core cloud (far-end). The three-level hierarchy depicts the elements' computing capacity as well as their edge computing characteristics. Sensors and actuators on the front-end provide extra and improved user responsiveness. The resource requirements have to be dispatched to the server, however, given their restricted capacity, near-end edge servers handle most network traffic and a variety of resource needs (such as real-time data processing and computation offloading). As a result of deploying edge servers, end users benefit from improved computation performance at the cost of increased latency. Far-end cloud servers provide greater processing power (e.g., big data analytic) and additional data storage space. The objective of this system architecture is to enable the edge network to support computation-intensive and time-critical applications. Furthermore, certain edge server apps offer data synchronisation via cloud communications.

G. DIFFERENTIAL PRIVACY

One of the efficient privacy preservation strategies is differential privacy which is used to maintain the confidentiality of data without risking data loss or data leakage. In 2006, C. Dwork first introduced additional noise to the data as a way to preserve privacy [28]. In terms of statistical databases, attempts are made to protect privacy based on differential privacy techniques adding noise to the data prior to query assessment. Researchers subsequently started to use differential privacy-related concepts in other domains to preserve privacy in the user's personal data. To achieve this outcome, differential privacy perturbs sensitive data through the addition of a specified (calculated) noise value. As such, differential privacy can guarantee that the presence or absence of a participant in a dataset will not affect the output results of database query. Researchers have also applied the concept of differential privacy in other applications such as health data monitoring in real-time and IoT data etc. Additionally, to protect data from IoT nodes in the context of blockchain-based IoT systems, they use data perturbation mechanisms using differential privacy [29].

When including differential privacy in data, it is important to consider two key parameters: sensitivity measurements and suitable noise additions. The added noise may conceal the critical value leaving the adversary unable to make an approximation of a particular individual's presence or absence. The sensitivity value typically varies according to the specifics of the scenario; for instance, applications that need high level privacy utilise large sensitivity values and those needing low level privacy utilise small sensitivity values. Other solutions have also been proposed by researchers including the choice of dynamic sensitivity values where sensitivity values vary automatically based on analyst and data provider requirements [30]. However, a high level of noise needs a high sensitivity level. The use of a high sensitivity value reduces data usefulness. Hence, a suitable trade-off between the need for privacy and the need for truthfulness must be maintained via adjustment to the sensitivity value. Moreover, the noise addition method is essentially a protective event involving minimum noise value calculations needed to protect data privacy. The noise output is related to the sensitivity value. The base function in this method needs the input to be of a certain parameter to calculate the amount of noise.

There are three noise addition methods that researchers use when calculating the noise value: Laplace mechanism, Exponential mechanism, and Gaussian mechanism. As with sensitivity, best choice noise addition method depends on the nature of application. If it is a numerical output for instance, the Laplace and Gaussian method will typically be used, whereas, the Exponential method is applied for non-numerical output [31]. In this paper, we consider the definition of differential privacy as follows:

Definition: ϵ-differential privacy [28]: A randomized mechanism f: D → R satisfies (ϵ, δ)-differential privacy if for any adjacent datasets D and D' and for any subset of outputs S $\subset$ R where R is the output space of f

$$\frac{Pr(f(D) = S)}{Pr(f(D') = S)} \leq e^{\epsilon} + \delta \tag{1}$$

H. PRIVACY ISSUES IN BLOCKCHAIN-BASED INTERNET OF THINGS

Blockchain technology relies on authentication and encryption services to preserve data security (i.e., secure transactions). Cryptography and the use of a public key encryption are linked to such blockchain services. This means that users must have access to both the public and private keys in order to manage their transactions. Public key cryptography works on the basis of two key types: public keys, also known as distributed network keys, and private keys, also known as personal individual keys. Public key infrastructure is the most frequent technique providing key management functions for cryptography in the blockchain. (PKI) techniques based on blockchain are decentralised, which eliminates the need for a centralised access point or a trusted third-party [12]. Furthermore, these methods do not require trustworthiness to be established via nodes or system users in order to make the public system more visible. Instant Karma PKI, Blockstack, and Certcoin are only a few of the blockchain approaches that have been mentioned in the literature to enable PKI encryption and transaction security on blockchain nodes. Blockchain privacy and security, on the other hand, are only now beginning to be fully addressed. As explained by S. Nakamoto in [32] any exposure of the private key owner's identity can lead to the disclosure of additional transactions by that owner using linking techniques. Furthermore, when exposed to certain types of attacks, the anonymity of blockchain users may be compromised [33].

Moreover, as a means of privacy in Ethereum, Ethereum uses cryptographic hash functions and transactions are secured using cryptographic mechanisms-based privacy. However, since Ethereum is a public ledger, all users may access the decentralized ledger. The transaction data is available online, and the inclusion of these cryptographic frameworks does not guarantee full privacy. Deanonymization attack is the most well-known privacy attack on Ethereum, in which data from a distributed ledger is deanonymized by linking and tracing features with other databases [34].

Hence, the methods for preserving privacy in blockchain applications is an important research issue. Some researchers have sought to improve blockchain privacy through the use of different strategies such as the use of two-level anonymity. Additionally, Christidis and Devetsikiotis in [35] focused on resolving confidentiality issues based on public blockchain transaction to enhance blockchain trustworthiness. Another potential solution is the use of a differential privacy preserving strategy that utilises data perturbation methods for the protection of private data in the blockchain. Differential privacy provides the functionality of adding noise to the stored distributed ledger records to address this problem. Differential privacy's randomness noise can be used in a variety of ways such as adding the noise for non-trusted users or users without a clear task in the network. It may be possible to only allow query evaluation in the public ledger to analyze any record or previous transaction and add noise to this query evaluation to protect privacy. Also, Ethereum's smart contract gives developers the ability to add differential privacy to their truncation's [34]. The flexibility of choosing suitable way to add noise based on privacy and utility requirements make the use of differential privacy optimal to overcome the privacy issues in blockchain based architecture.

III. EXISTING WORKS IN BLOCKCHAIN

Data security and privacy with IoT devices in a smart home is one of the major challenges as connected IoT devices are vulnerable to various attacks and they lack basic security features. To address these issues, numerous centralized solutions have been proposed [36]. Amadeo *et al.* [37], proposed an information-centric network-based system for smart home services with a three-layered architecture including remote cloud, fog layer with smart home servers and end devices. The platform enables real-time systems to be deployed, including smart monitoring and control applications. Another framework proposed in [38] integrate existing IoT architecture components. They looked at IoT smart home challenges and solutions in order to bridge the gap between current state-of-the-art smart home applications and the possibility of integrating them into an IoT-enabled world. Also, Sun *et al.* [39], promote the vision of Smart and Connected Communities (SCC). They integrate IoT with cyber physical cloud computing and big data for smart tourism to enhance a community's preservation, liveability, revitalisation, attainability, and security. However, all these works are based on central architecture where the communication and processing overhead, access control and a single point of failure are major challenges. Therefore, various researchers [12], [32]–[35] have turned-out the attention towards distributed Frameworks and proposed popular blockchain based solutions for various IoT use cases. Furthermore, because in the design of blockchain-based IoT systems, privacy is not pre-enforced and private data can be leaked using certain attacking approaches. Researchers proposed various privacy preservation strategies such as differential privacy for different applications of blockchain based on cloud computing and machine learning [49]–[61].

A. BLOCKCHAIN AUTHENTICATION, ACCESS CONTROL AND EDGE COMPUTING IN SMART HOME APPLICATION

Authors in Lee *et al.* [3], looked at the concerns surrounding 'gateways,' or connections between IoT devices, claiming that such centralised arrangements present several security risks such as integrity, certification, and availability. The authors responded by proposing a blockchain-based smart home gateway network that can protect against potential gateway attacks. The blockchain technology network, which is made up of three layers: device, gateway, and cloud, is utilised at the gateway layer to facilitate decentralisation by storing and exchanging data blocks. This maintains data integrity both inside and outside the smart home and availability through authentication and communication between network users. On the other hand, their architecture has some limitations in terms of the computing complexity imposed by blockchain operations at the gateways.

Moreover, in [40] authors integrates both blockchain and group signature to anonymously authenticate group members, as well as message authentication code to efficiently authenticate home gateway without leaking information in smart home scenario. In HomeChain, all request records from group members (or revocation requests from the group manager) will be chained into the blockchain. Due to the immutability of blockchain and traceability of group signature, these records are not easy to be tampered or deleted and hence may provide reliable auditing. however, there was no access control policy but, they adopted a revocation list to revoke authorities of malicious users.

The benefits of using Ganache, Remix, and web3.js architecture for smart home based IoT blockchain (SHIB) to overcome the difficulties of data privacy, trust access control, and the ability to extend the system were advocated by the authors in [41]. They presented an IoT gateway for connecting a smart home's cluster of IoT devices to a blockchain network. Their work is complicated by the fact that each user and IoT device must be assigned to one and only one subject-object pair due to the fact that, gateway may not have enough computer power to handle large transactions.

In [42], authors presented a private blockchain-based access control (PBAC) approach to solve data security and privacy issues while using smart devices in smart home systems. Within the IoT system, the proposed PBAC provides ‘‘an unforgeable and auditable foundation’’ that can prevent unauthorised data access, protect data security from threats, and enable accurate, robust, and instant access to information. They only recommended one internet server as an administrator. However, the entire system fails if the administrator is inactive.

Authors in [36] proposed utilising a blockchain-based approach based on Proof-of-Authority to develop a consensus mechanism for better managing home appliances in a decentralised framework. When compared to a standard Proof-of-Work based system, the authors demonstrated additional features to improve the effectiveness of a blockchain method using Proof-of-Authority as the consensus mechanism to address security concerns.

The implementation of IoT and blockchain-based Multi-Sensory Frameworks in the context of in-home quality of life (QOL) for recently diagnosed cancer patients was studied in [43]. Multiple medical and ambient intelligent IoT sensors can capture QOL data from the smart home environment and securely share it with a specified community of interest using the authors' suggested blockchain and off-chain based framework. The in-home secure monitoring system captures QOL data, such as transactional records and multimedia-based big data (e.g. physiological and mental state data), which the authors may manage using blockchain-based data analytics.

In [14], the author suggested a lightweight blockchain-based architecture for IoT that considerably decreased the overheads of traditional blockchain while retaining the majority of its security and privacy benefits. The design allows high-resource devices to create an overlay network in order to use a publicly available distributed blockchain that ensures end-to-end security and privacy. Furthermore, it employs a distributed trust to provide excellent security and privacy for IoT applications, it minimizing the time necessary to execute block validation. However, no information on the establishment of this scalable blockchain or the security certificates was provided.

In [44] the author implemented IoT-based architecture in tandem with BC (Hyperledger Fabric) to assess the validity of the communicating devices whether normal or malicious. They tested their scheme in a smart home-based scenario. However, the transaction size in Fabric are larger than other blockchain platform because they also carry the certificate information for approval. Therefore, the latency gets worse with increase in block size in their scenario.

In [45] authors proposed an Attribute-Based Access Control (ABAC) framework for IoT systems by using the emerging Ethereum smart contract technology. The framework consists of four different smart contracts to manage ABAC policies, attributes of subjects and objects and perform access control. However, the main drawback of their framework is that, the average time for access control is high due to complex interactions between the access control contract and other smart contracts for retrieving attributes and policies.

In [46] authors propose a smart contract-based framework, which consists of multiple access control contracts (ACCs), one judge contract (JC) and one register contract (RC), to achieve distributed and trustworthy access control for IoT systems. However, one ACC is deployed for only one subject-object pair. Therefore, the gas cost will increase linearly as the number of subject-object pairs of the system increases which indicate a higher cost to implement the framework. In our work, we address all these issue by implementing Ethereum smart contract to decrease the transaction size and the latency. We also proposed two smart contracts to avoid the complexity and consume less gas and better cost compared with other frameworks.

B. INTEGRATING DIFFERENTIAL PRIVACY INTO BLOCKCHAIN

Blockchain data training using machine learning algorithms are currently being used to generate useful solutions by providing better insights to the available data across most fields including, bioinformatics and wireless communication [47]. In addition, a machine learning based approach combines many practical applications including blockchain and healthcare. This creates new possibilities in data analytics. Machine learning includes the use of an available dataset to train a computer. Traditionally, the dataset has a centralized information but, when used in blockchain, the training occurs in a decentralized distributed information source with multiple computing nodes involved in the learning process [48]. Because, data are distributed across all computing nodes, learning can be supported by a privacy preserving strategy. To resolve this issue, Chen *et al.* [49]

recommended a decentralized approach to machine learning based on differential privacy that protects user privacy by utilising stochastic gradient descent (SGD). Referred to as ''LearningChain'', the authors claim that the strategy facilitates both private learning and a reduction in error rates. The strategy relies on the process of perturbing normalized local gradient information prior to it being mined into the blockchain. As such, data are protected and made tamper-proof, and only targeted and protected records are mined in the blockchain. The authors also utilise a public blockchain and conduct consensus via the use of a proof-of-work (PoW) consensus tool.

To ensure that the system remains protected against byzantine attacks, an l-nearest aggregation algorithm is applied providing protection to private data prior to and during the collection by rendering it impossible to differentiate from its neighbours. An Ethereum network is used to develop the final model and is analysed via the use of MNIST [50], and Wisconsin breast cancer datasets [51]. Kim *et al.* [52] also discuss differential privacy integration in machine learning scenarios using blockchain. Their work improves usability and transaction latency as well as provide privacy protections by conducting experiments to add noise repeatedly using differential privacy. Repeated-additive noise is utilised along with local gradient to protect the privacy of blockchain users. A private blockchain was used by the authors to mine the blocks with a PoW consensus tool. According to the authors, the trust users have in distributed machine learning can be strengthened with the introduction of an efficient perturbation tool using differential privacy. Furthermore, the authors stated that it increases user participation by overcoming attacks across the blockchain network. With this in mind, it may be concluded that a differential privacy protection strategy is an efficient way to protects the privacy of users in scenarios related to machine learning using a decentralized blockchain.

Advances in smart grids are also developed and deployed leading to new challenges in research and technology. One challenge for instance is how to manage and perform smart grid operations (e.g. communication, energy trading, renewable energy management, and so forth) effectively [53]. The research field is currently investigating how to address these challenges while also supporting the smart grid transformations to manage the challenges. A potential solution to improve the management of smart grid operations is to integrate it with blockchain technology. Various scenarios are currently under consideration such as; the deployment of blockchain at specific layers of the smart grid (e.g. consumption layer and generation layer) to make the technology more secure for users. Researchers in [54] provide a case study of the implementation of a blockchain-based micro-grid in Kazakhstan, focusing on its potential to improve the nation's energy trading possibilities via blockchain. Indeed, there is a wide discussion in the literature regarding smart grid and blockchain integration. However, it is also clear that the literature often neglects to focus on privacy preservation issues in such scenarios. As a public distributed ledger, the integration of privacy protection in these types of models is paramount.

Most operations conducted via smart grid scenarios are regarded as real-time data analytic. Hence, the integration of differential privacy noise-additive tool is a possible solution to these challenges. The authors in [55] conducted several scenarios related to private energy trading in blockchain-based smart grids. A private energy trading model was developed by the authors by applying basic differential privacy implementation and by comparing their model with current methods of differential privacy. The model relies on a blockchain-based token bank to perform and store transactions. In addition, the model provides differential privacy by inhibiting linkages and circumventing data mining with minimal consumption of computational power. Moreover, the integration of differential privacy into a de-regulated blockchain-based smart grid is presented in [56]. The authors enhanced the proof-of-authority (PoA) mechanism through its integration with PageRank to generate reputation ratings. Laplace noise was added to enhance user privacy protections and thus promote user participation. According to the authors, user trust is enhanced in their strategy by overcoming issues of similarity, and double-spending attacks. These examples demonstrate that user privacy should be a key issue of focus when integrating blockchain with smart grid. Hence, additional research is also needed to generate evidence that blockchain-based smart grids can be trusted.

C. INTEGRATION OF DIFFERENTIAL PRIVACY, MACHINE LEARNING AND BLOCKCHAIN

Cloud computing is increasingly utilised by all industries. In turn, researchers continue to enhance this practice with the development of more advanced cloud computing models. An example of such a model is edge/fog computing and its capacity to provide fast access to critical tasks [57].

Researchers have also developed models that use machine learning algorithms to extract co-related features on data in the cloud. Moreover, efforts are being made to improve data storage, network access and control reliability, and large-scale server functionality by integrating blockchain with edge computing [7]. This has prompted researchers to investigate edge and cloud computing based on blockchain to improve efficiencies and reduce time-delay [58]. Notwithstanding these efforts, some researchers point to privacy leakage as an issue in cloud systems based on blockchain [59]. To address these flaws, researchers now look to employ strategies around the integration of privacy protection with blockchain-based cloud as a possible solution. In [60], authors undertake the integration to distribute autonomous privacy budget when mining in blockchain. The integration resulted in increased work-load while executing queries, with the authors claiming that the method both provides answers to queries more effectively as well as protect user privacy. Researchers also utilise private/permissioned blockchain models with byzantine fault tolerant (BFT) consensus mechanisms to guarantee the cooperation and control of authorized nodes. Moreover, the researchers also assert that, their mechanism manages all re-identification attacks effectively as a result of data perturbation. Zhao *et al.* [61], also present their exploration of federated learning-based edge computing. Hence, considering the points discussed above, it is evident that edge and cloud computing based on blockchain is not fully secured and private. As a result, further research is required to improve privacy outcomes in decentralized cloud scenarios.

However, many of these works lack real implementation and are established only in theory. Others still have limitations in regards to communication and computation cost. In a smart home scenario, there is a lack of privacy enhancement mechanisms and in particular, when such systems are connected to the cloud. Conversely, our work focus on developing and implementing an architecture which integrates the access control scheme using two smart contracts deployed in multi-edge servers to achieve a secure distributed blockchain for smart home IoT devices. The use of many edge servers provides a complementary 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. To enhance the privacy in our model, we introduce the concept of differential privacy using Stochastic Gradient Descent (SGD) algorithm. To the best of our knowledge, this is the first work that aims to implement a privacy preserving strategy by integrating differential privacy mechanism with a machine learning algorithm in blockchain smart home scenario.

IV. PROPOSED ATTRIBUTE BASED ACCESS CONTROL SCHEME FOR SMART HOME

The following section explains the key architecture and design details of our proposed blockchain based architecture, in which Ethereum smart contracts are used to register, and manage Home user, IoT smart home devices and edge servers.

A. SYSTEM ARCHITECTURE

Fig.1 shows the proposed system architecture which consists of four 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. All the participants have a unique Ethereum Address with public and private keys. The edge servers and the cloud node connect directly with the smart contract through an Ethereum client, while end users connect through a wallet/front end application. The following summarizes the key role of different architectural elements:

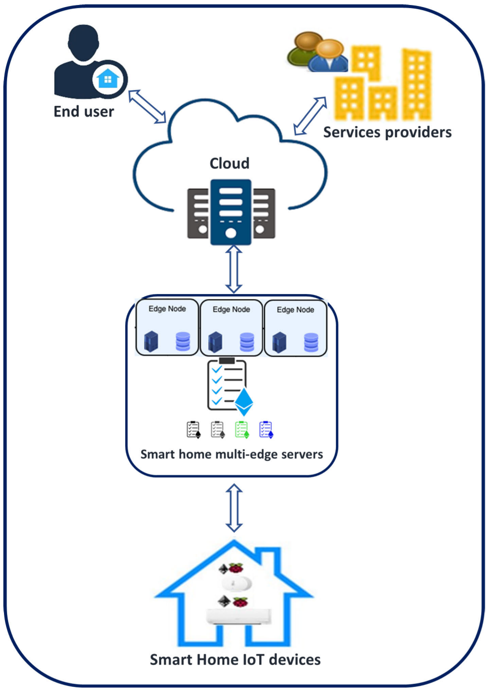


FIGURE 1. Proposed system architecture.

1) End user: Request access permission through the smart contract to access a certain smart home device. The home user device could be PCs, tablets, and smartphones that can request a service from the servers such as checking the home temperature. Also, there are service accessors involving service providers such as health care, police or other parties who need to access the smart home data to provide services to the end users.
2) IoT devices: The IoT devices primarily include sensors and actuators that can observe home data (e.g., temperature) or perform some operations (changing the air conditioner status).
3) Smart home multi-edge servers (Admin Edge): An Edge node is a device or a cluster of devices that communicate directly with the IoT devices and the cloud. It provides a range of services such as collecting home data from the sensors and sending commands to actuators to perform a task. Also, it can request or store data in the cloud. Edge nodes process all incoming and outgoing transactions and use a shared key for local communications with IoT devices and local storage. It maintains the smart contracts that manage registering the end users and IoT devices, authenticates end users to access the IoT devices. The mining work is only done by the edge servers which have more resources than the IoT devices. Moreover, the edge servers propagate the data to the cloud for further storage or analysis using differential privacy enhancement mechanism.
4) Cloud: Infrastructure which provides long-term data analytic 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.

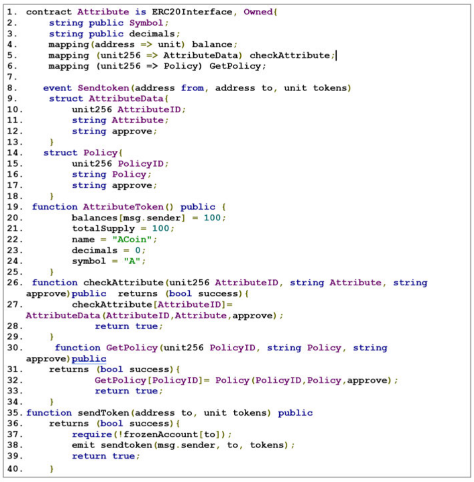
B. ATTRIBUTE BASED ACCESS CONTROL AND SMART CONTRACTS

The proposed framework comprises of two Ethereum smart contracts; the Register contract and the Access contract, to avoid the complexity of a single smart contract. The first contract stores and manages the subject and object attributes, as well as policies (e.g., updating, adding, removing). The Access contract controls IoT device access by producing ERC-20 tokens and finalising authorisation to access IoT devices. The smart contract's description is as follows:

1) Register contract: The policy is used to register and maintain the attributes of individuals and IoT devices on the blockchain. This contract can only be executed by the administrator. Users, devices, and policies can all be added by administrators as shown in Fig.2.
   Each user and IoT device has its own unique identifier (Ethereum account address) and a set of attributes related to it. This contract includes functions for adding, deleting, and changing subject and object attributes. In addition, based on the user type, this contract describes the policy associated with each user and IoT device as described in Fig.3.
   A policy is a statement that states which user can do an action on an IoT device by combining a set of subjects (users), a set of Objects (IoT devices), and a set of Actions. Table 1 is an example of a policy.
2) Access Contract: This contract governs access requests from users (subject) to IoT devices (object). As shown in Fig.4, the user executes this contract to request a token in order to communicate with an object. This contract includes functions for validating subject attributes and checking policy; the Access Contract (AC) assesses whether the subject has rights to do an action on the object based on the policy received, and then sends a token to the subject. The main functions of the contract are Check attribute(), Get policy() and TransferToken(). This contract is also in charge of generating ERC-20 tokens. Fig.5 illustrates how to use some Access contract functionalities. To prevent a valid user from flooding the network with access control requests, each user has a specific number of valid tokens at a time dependent on user type.

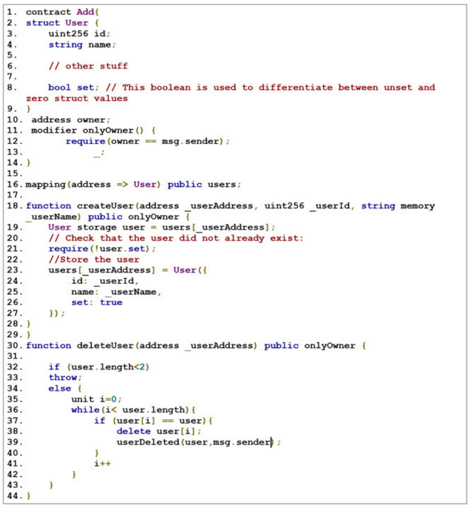
TABLE 1. Example of user attributes, IoT attributes and permissions.

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



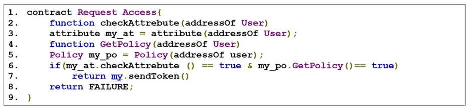
```
contract Attribute is ERC20Interface, Owned{
    string public Symbol;
    string public decimals;
    mapping(address => unit) balance;
    mapping (unit256 => AttributeData) checkAttribute;
    mapping (unit256 => Policy) GetPolicy;

   event Sendtoken(address from, address to, unit tokens)
    struct AttributeData{
        unit256 AttributeID;
        string Attribute;
        string approve;
    }
   struct Policy{
        unit256 PolicyID;
        string Policy;
        string approve;
    }
 function AttributeToken() public {
        balances[msg.sender] = 100;
        totalSupply = 100;
        name = "ACoin";
        decimals = 0;
        symbol = "A";
    }
 function checkAttribute(unit256 AttributeID, string Attribute, string
approve)public  returns (bool success){
        checkAttribute[AttributeID]=
AttributeData(AttributeID,Attribute,approve);
            return true;
    }
     function GetPolicy(unit256 PolicyID, string Policy, string
approve)public
    returns (bool success){
            GetPolicy[PolicyID]= Policy(PolicyID,Policy,approve);
            return true;
    }
function sendToken(address to, unit tokens) public
    returns (bool success){
        require(!frozenAccount[to]);
        emit sendtoken(msg.sender, to, tokens);
        return true;
    }
```

FIGURE 4. Main access contract function.



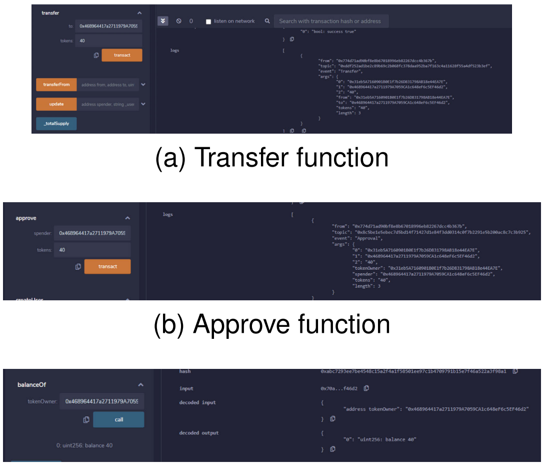
```
contract Add{
struct User {
    uint256 id;
    string name;

    // other stuff

    bool set; // This boolean is used to differentiate between unset and
zero struct values
}
 address owner;
 modifier onlyOwner() {
        require(owner == msg.sender);
            _;
}

mapping(address => User) public users;

function createUser(address _userAddress, uint256 _userId, string memory
_userName) public onlyOwner {
    User storage user = users[_userAddress];
    // Check that the user did not already exist:
    require(!user.set);
    //Store the user
    users[_userAddress] = User({
        id: _userId,
        name: _userName,
        set: true
    });
}
}
function deleteUser(address _userAddress) public onlyOwner {

    if (user.length<2)
    throw;
    else {
        unit i=0;
        while(i< user.length){
            if (user[i] == user){
                delete user[i];
                userDeleted(user,msg.sender);
            }
            i++
        }
    }
}
```

FIGURE 2. Add and delete user functions.



```
contract Request Access{
    function checkAttrebute(addressOf User)
    attribute my_at = attribute(addressOf User);
    function GetPolicy(addressOf User)
    Policy my_po = Policy(addressOf user);
    if(my_at.checkAttrebute () == true & my_po.GetPolicy()== true)
        return my.sendToken()
    return FAILURE;
}
```

FIGURE 3. Request access function.



to: 0x468964417a2711979A705
tokens: 40

```
"0": "bool: success true"
}
logs    [
          {
            "from": "0x774d71ad90bf8e8b67018996eb82267dcc4b367b",
            "topic": "0xddf252ad1be2c89b69c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef",
            "event": "Transfer",
            "args": {
                "0": "0x31eb5A7160901B0E1f7b26D831798AB18e44EA7E",
                "1": "0x468964417a2711979A7059CA1c648eF6c5EF46d2",
                "2": "40",
                "from": "0x31eb5A7160901B0E1f7b26D831798AB18e44EA7E",
                "to": "0x468964417a2711979A7059CA1c648eF6c5EF46d2",
                "tokens": "40",
                "length": 3
            }
          }
        ]
```

(a) Transfer function

spender: 0x468964417a2711979A705
tokens: 40

```
logs    [
          {
            "from": "0x774d71ad90bf8e8b67018996eb82267dcc4b367b",
            "topic": "0x8c5be1e5ebec7d5bd14f71427d1e84f3dd0314c0f7b2291e5b200ac8c7c3b925",
            "event": "Approval",
            "args": {
                "0": "0x31eb5A7160901B0E1f7b26D831798AB18e44EA7E",
                "1": "0x468964417a2711979A7059CA1c648eF6c5EF46d2",
                "2": "40",
                "tokenOwner": "0x31eb5A7160901B0E1f7b26D831798AB18e44EA7E",
                "spender": "0x468964417a2711979A7059CA1c648eF6c5EF46d2",
                "tokens": "40",
                "length": 3
            }
          }
```

(b) Approve function

tokenOwner: 0x468964417a2711979A705

0: uint256: balance 40

```
hash             0xabc7293ee7be4548c15a2f4a1f58501ee97c1b4709791b15e7f46a522a3f98a1
input            0x70a...f46d2
decoded input    {
                     "address tokenOwner": "0x468964417a2711979A7059CA1c648eF6c5EF46d2"
                 }
decoded output   {
                     "0": "uint256: balance 40"
                 }
```

(c) Token balance

FIGURE 5. Example of access contract functions execution.

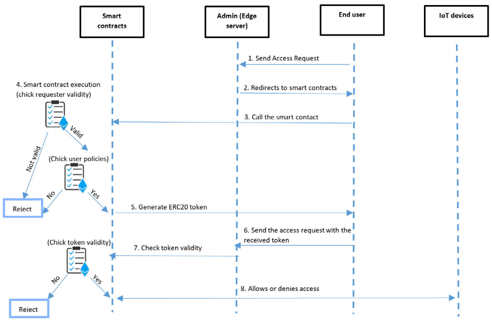
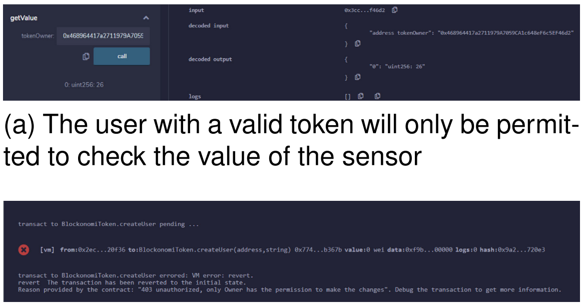


FIGURE 6. Typical transactions in proposed scheme.



```
0: uint256: 26

input            0x3cc...f46d2
decoded input    {
                     "address tokenOwner": "0x468964417a2711979A7059CA1c648eF6c5EF46d2"
                 }
decoded output   {
                     "0": "uint256: 26"
                 }
logs             []
```

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

```
transact to BlockonomiToken.createUser pending ...

[vm] from:0x2ec...20f36 to:BlockonomiToken.createUser(address,string) 0x774...b367b value:0 wei data:0xf9b...00000 logs:0 hash:0x9a2...720e3

transact to BlockonomiToken.createUser errored: VM error: revert.
revert  The transaction has been reverted to the initial state.
Reason provided by the contract: "403 unauthorized, only Owner has the permission to make the changes". Debug the transaction to get more information.
```

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

FIGURE 7. User request for room temperature data.

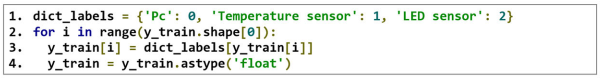
C. SYSTEM DESIGN

The proposed system provides authentication for users using an attribute based access contract and token distribution. Fig.6 illustrates typical attribute-based access contract transactions with this authentication mechanism. Users can remotely access or control home devices using the fresh generation token that only the requester is able to receive the response from the legitimate home admin. There are four phases in our system through which the transactions are carried out; Initialization, Request Control, State Delivery and Chain Transaction.

1) Initialization: For the sake of demonstration, we'll assume that family members make up a group of users from whom a group administrator is picked. To add more users and IoT devices, an admin uses the Register Contracts command. For signing transactions, users assign their Ethereum Address (EA) and private keys. In turn, each home admin is in charge of the group public key, which is used to verify transactions. To avoid a single point of failure, the admin is run on many edge nodes through the miners associated with these nodes.
2) Request Access: A token is generated for a specific period and with exact access time when a user wishes to publish an access or control request with the home admin. To avoid replay attacks and profiling, this is the recommended strategy. The user constructs the transaction from his or her requirements after getting the token by activating the TransferToken () from Access Contract. For example, If a user requests the room temperature, the transaction is computed when the user is redirected to the smart contract and asks a token. In that contract, three primary functions are invoked: Check attribute(), Get policy(), and TransferToken ().The user then sends the admin the received valid token along with the request for access. If the user has a valid token, they will be permitted access. The output of valid and invalid user requests for accessing data on room temperature is shown through the screen shots in Fig.7.
3) State Delivery: The smart contract is monitored by the home admin for new requests. If the transaction passes verification after a user asks new access or service, the home admin validates the token validity and allows or denies access to the IoT device.
4) Chain Transaction: Admin nodes (miners) are in charge of obtaining transactions from the smart contract, and they compete to be the first to solve the PoW for chaining the data block to the blockchain. Once the PoW is solved, the miner broadcasts the solution to the blockchain network in order to establish consensus. The mining reward is given to the first miner who successfully mines a block that reaches consensus.

D. DIFFERENTIAL PRIVACY ENHANCEMENT MECHANISM

In this paper we implement privacy-preserving classification using edge computing and blockchain scenario. As a privacy-preserving Machine learning throughout, it fulfills learning accuracy. The proposed mechanism trains the Machine learning model accurately to suit all IoT smart home data. The model also classifies a given packet to an IoT device in the smart home scenario as shown in Fig.8.



```
dict_labels = {'Pc': 0, 'Temperature sensor': 1, 'LED sensor': 2}
for i in range(y_train.shape[0]):
  y_train[i] = dict_labels[y_train[i]]
  y_train = y_train.astype('float')
```

FIGURE 8. Classification model.

The aim is to provide a privacy-preserving data aggregation method, in the context of Smart Homes that agree to provide their data to a cloud server, so that the cloud can learn privately from data produced from IoT devices inside the home and then deliver these data to external entity in order to provide better services for home users.

As Fig.9 shows, we consider that a number of edge nodes have private data from the IoT devices in the smart home and collaborate with each other to return the results to the cloud. These edge nodes assist the smart home in sharing their data with the cloud by learning the model and train the data before sending final result to the cloud. The edge nodes first calculate the gradients based on the current model while attempting to limit the privacy leakage. They employ a differential privacy scheme to perturb their data.

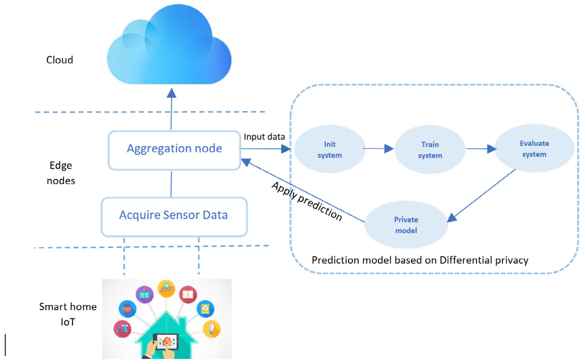


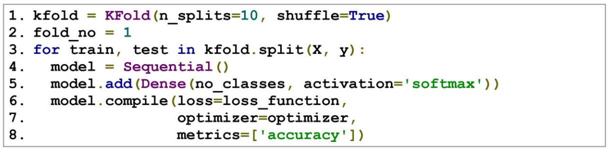
FIGURE 9. Edge node functions for data privacy scheme.

The cloud collects the gradients broadcasted by the edge nodes and perform their desired scheme to analyse the data. In the proposed model, we consider two different methods to train the model using a machine learning algorithm on the prepared data. First, without considering privacy, we train a neural network with one-layer on the data and analyze accuracy of the proposed scheme. We call this approach as a ‘‘plain algorithm’’. Second, we train the same one-layer neural network on our data based on the scenario explained before. We use Stochastic Gradient Descent (SGD), as one of the most popular optimization algorithms [49]. Stochastic gradient descent (SGD) algorithms have received significant attention recently because they are simple and satisfy the same asymptotic guarantees as more computationally intensive learning methods [62]. We call the second algorithm ‘‘Private algorithm’’.

1) PLAIN ALGORITHM

As mentioned before, this algorithm is without considering any privacy and the data is completely handed over to the cloud server as described through the algorithm in Fig.10. The algorithm specification is as follows:

- Model: K-fold-one-layer neural network
- Loss function: categorical cross entropy
- Optimizer: adam (adaptive moment estimation)
- Number of epochs (training rounds): 10

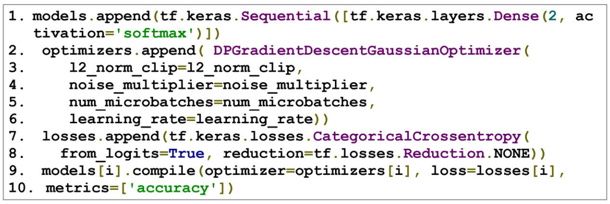


```
kfold = KFold(n_splits=10, shuffle=True)
fold_no = 1
for train, test in kfold.split(X, y):
  model = Sequential()
  model.add(Dense(no_classes, activation='softmax'))
  model.compile(loss=loss_function,
                optimizer=optimizer,
                metrics=['accuracy'])
```

FIGURE 10. K-fold-one-layer neural network-plain algorithm.

2) PRIVATE ALGORITHM

The basic idea of this approach is presented in Fig.11. The scheme called differential private stochastic gradient descent (DP-SGD), modifies the gradients used in stochastic gradient descent (SGD), which lies at the core of almost all deep learning algorithms. Models trained with DP-SGD provide provable differential privacy guarantees for their input data. We made the following two modifications to the SGD algorithm in order to accommodate privacy aspects with the data:



```
models.append(tf.keras.Sequential([tf.keras.layers.Dense(2, ac
 tivation='softmax')]))
optimizers.append( DPGradientDescentGaussianOptimizer(
    l2_norm_clip=l2_norm_clip,
    noise_multiplier=noise_multiplier,
    num_microbatches=num_microbatches,
    learning_rate=learning_rate))
losses.append(tf.keras.losses.CategoricalCrossentropy(
   from_logits=True, reduction=tf.losses.Reduction.NONE))
models[i].compile(optimizer=optimizers[i], loss=losses[i],
 metrics=['accuracy'])
```

FIGURE 11. K-fold-one-layer neural network-private algorithm.

- First, the sensitivity of each gradient needs to be bounded. In other words, we need to limit how much each individual training point sampled in a mini batch can influence gradient computations and the resulting updates applied to model parameters. This is done by clipping each gradient computed on each training point.
- Random noise is sampled and added to the clipped gradients to make it statistically impossible to know whether or not a particular data point was included in the training dataset by comparing the updates which SGD applies when it operates with or without this particular data point in the training dataset.
- We select following parameters and specifications in the design of our algorithm:
  - Model: k-fold-one-layer neural network
  - Loss function: categorical cross entropy
  - Optimizer: DP-SGD (differentially private stochastic gradient descent)
  - Number of epochs (training rounds): 10
  - l2-norm-clip: 1.5
  - noise multiplier: 2

3) DATASET

The experiment has been conducted to detect and classify a type of device in a private blockchain of the Smart home. One such way is to observe how machine learning techniques on captured packets (Stored in files like pcap files) are applied in order to distinguish between different devices in the network. The dataset was produced by generating a pcap file using Wireshark to capture the network packets in our private network. Our synthetic dataset consists of n = 11,000 samples. Using Tshark, we then filter the captured packets and extract the headers of each packet. Then, processing and creating the dataset is done using the Python script. We selected our dataset based on network traffic generated by our private Ethereum network, thus providing accurate representations of the devices we use in the experiment.

4) IMPLEMENTATION

Our system is developed on a secure network. The approach is based on a private Ethereum network that consists of one laptop (Dell XPS) that serves as an edge server, with two miners connected to two single-board computers (Raspberry Pi 3 Model B). The temperature sensors attached to the LEDs and one home user laptop are then utilised to simulate the aforesaid scenario. The edge server has four independent CPU cores and 16 GB of RAM. One processor core is dedicated to the mining environment, while the remaining processor cores are dedicated to the edge computing service. The miner has a 3.5 GHz CPU, 8 GB RAM, and 1 TB storage capacity. Two Raspberry Pis with a 1.2 GHz CPU, 1 GB RAM, and 32 GB storage are included in our IoT devices, together with accessory modules such as a temperature sensor and an LED sensor. As a home user, we configure one laptop with a 2.2 GHz CPU, 16 GB RAM, and 256 GB storage.

The smart contract is developed on the edge server using G0-ethereum as blockchain framework and Solidity as programming language. The contracts are written and compiled using the Remix integrated development (IDE) (Remix 2020). The model additionally employs Web3.js (Ethereum JavaScript API) for contract deployment and compilation, as well as contract status monitoring. The HTTP connection is used to interact with the corresponding geth client via JavaScript. A basic html web page supports the interface between the home users and the devices. The Raspberry Pis run the Raspbian operating system with Go-Ethereum to work in light mode without block mining functions. The first laptop in the testbed supports two edge service providers and a block miner that solves a PoW puzzle. The Raspberry Pis and the second laptop function as blockchain clients, creating and submitting resource requests transactions to the edge server. According to the preceding configurations, the edge server functions as a ''complete'' blockchain node, storing all transactions, executing predefined smart contracts, and mining new blocks. IoT devices, on the other hand, function as ''light'' blockchain nodes that store transaction data.

The private blockchain is set up through number of stages that include choosing a compatible version of Ethereum, launching geth with Windows power shell, and requiring each node to meet numerous conditions before joining. This includes: (1) creating the first block using the genesis file (Test.json), (2) connecting to the same blockchain using the network ID and (3) creating the private blockchain using the geth command. For each node, the miner creates an account with a private and public key and indexes it according to its address, from which it can communicate with other nodes and smart contracts. The geth on each node is then started with a command that includes various flags for various functionalities. All nodes have the ''no discovery'' flag set to prevent them from being exploited by external attackers. As a result, they are unable to connect to other peers unless they have specified addresses. The node ID is then retrieved via a specific command, allowing syncing to take place. To build a private blockchain with completely synchronised nodes, the last step is repeated with the two Raspberry Pi and the home user laptop.

The smart contracts assign varying permissions to different devices based on the user type, with the edge server having full access to all functionalities while other users and IoT devices are only allowed to use a subset of them. If a user or several vulnerable devices are compromised, this setting limits the impact of the attacker's malicious activity.

V. EVALUATION AND ANALYSIS

This section provides a discussion on the security, privacy and performance analysis of the Attribute based smart contract edge computing scheme. We also present the performance of our integrated scheme using differential privacy enhancement model.

A. THREAT MODEL

Our goal is to collect Smart Home data from the edge nodes and analyze efficiency of our proposed scheme using different threat models. Since all data stored in blocks will be available to all blockchain users, we assume that, the adversary in our model may have full access to the data. We focus on side channel attacks where, adversaries use machine learning algorithms to infer information of smart home IoT devices by monitoring the incoming/outgoing network traffic from/to smart home. We would like to emphasize that, traffic patterns extracted from the IoT data may provide the adversaries to correlate their side information on some residents, thus giving adversaries prior information aid in lunching inference attack on the system. As a result, the adversaries can create a profile about smart home residents and launch subsequent sophisticated attacks such as the linkage attack.

In addition, we also consider other threats associated with the malicious user where adversaries steal identity information such as the geographical data about the edge node and allowing the adversary to steal specific tasks the edge node execute. Also, another assumption is the adversary can legally communicate with the edge node and as a result, leaking geographical information. Attackers can easily measure the communication time and estimate the physical distance from measuring/ comparing latency.

We assume that the cloud server deployed is secured as it is one element of the described architecture in section Fig.1. The classification model is trained on different edge nodes with a tailored machine learning algorithm to classify a given packet to one of the IoT devices in the smart home.

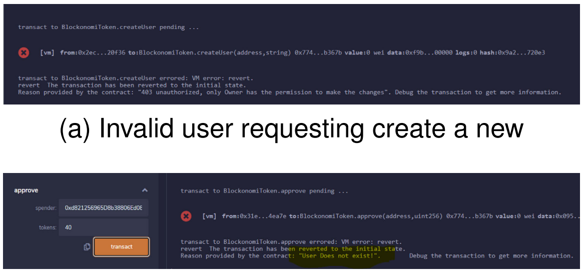
B. SECURITY ANALYSIS

Confidentiality aims to ensure unauthorised users are prevented from gaining access to IoT devices and their data and making sure that private data is delivered only to the intended users. One approach to achieve confidentiality is message encryption using SSL session after authenticating the user successfully [22]. As a powerful feature of blockchain, our framework assigns a unique 20-byte Ethereum Addresses (EA) directly to authorised node (including IoT devices) with almost no collision. EA has asymmetric public key pairs that can be used to establish secure SSL session for communication between any authenticated nodes such as authenticated user or IoT device. During the private network formation, 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 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 transaction received is 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 a valid token and only the intended user can use that Token. Fig.12 shows the revert error when anyone other than the admin tries to create a user or issue a 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 prevents 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 the target in order to disrupt its availability. The use of attribute-based access control smart contracts in our architecture reduces 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 device to issue a 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 policy is enforced automatically by the smart contracts. If adversaries compromise and control the IoT devices for malicious activities, such as making continuous resource requests, or initiating denial of service attacks, the smart contracts will execute automatically based on the preprogrammed policies of the total token supply, the access time and duration. For example, in our scenario we specify the total token supply by 100 form each user, if users or devices request an access, the request contract will issue one valid token at a time and if the requests are exceeded the number of their token supply, the transaction will be rejected.



```
transact to BlockonomiToken.createUser pending ...

[vm] from:0x2ec...20f36 to:BlockonomiToken.createUser(address,string) 0x774...b367b value:0 wei data:0xf9b...00000 logs:0 hash:0x9a2...720e3

transact to BlockonomiToken.createUser errored: VM error: revert.
revert  The transaction has been reverted to the initial state.
Reason provided by the contract: "403 unauthorized, only Owner has the permission to make the changes". Debug the transaction to get more information.
```

(a) Invalid user requesting create a new

```
spender: 0xd821256965D8b38806Ed08
tokens: 40
```

```
transact to BlockonomiToken.approve pending ...

[vm] from:0x31e...4ea7e to:BlockonomiToken.approve(address,uint256) 0x774...b367b value:0 wei data:0x095..

transact to BlockonomiToken.approve errored: VM error: revert.
revert  The transaction has been reverted to the initial state.
Reason provided by the contract: "User Does not exist!".      Debug the transaction to get more information.
```

(b) Invalid user requesting for token

FIGURE 12. Revert transaction.

2) Modification attack: In this attack, the attacker may try to alter or delete stored data of a particular user or device. To launch this attack, the attacker has to compromise the local storage security. Different cases of modification attacks have been discussed in blockchain based information sharing frameworks. Authors in [63], [64] claimed that the implementation of smart contract protocol prevent the adversary from breaking the security of their proposed scheme. Similarly, in our scheme only the admin has the right to store, delete or update the data based on the policy in the smart contracts. All the information about users, devices and policy are shared between the edge nodes and the cloud, assuming adversary wants to change or modify the ID of a user or any device. The change will be detected by the edge nodes 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 [9] 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 controls 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 the contract, the transaction is dropped.

The second defence is that all devices in the home are required to have a unique address and follow the 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.

C. PRIVACY ANALYSIS

In our proposed model, we assume that all participants have a verified identity that is managed and issued by the access control scheme in the smart contract in a private blockchain. Therefore, the identity privacy in our framework is out of the scope of our work. We only consider privacy leakage from data when a learning process runs.

We present security analysis on the proposed differential privacy-based blockchain system, which are associated with the pre-defined threat environment given in the threat model section. Based on the threat assumption, adversaries have full access to all data stored in blocks. In our model, for the first type of threat, without adding noise, adversaries can easily obtain real identities and behaviour of users through mining information or launching a linkage attack. Fortunately, our model uses a differential privacy protection method (Gaussian distribution mechanism) to add noise into the real data, such that a distortion is made to protect the target set. We observe that, using the gaussian mechanism can successfully screen and classify the IoT devices while insure and guarantee the privacy of all data.

In data mining-based attacks, from the adversary's perspective, adding the noise can escalate the complexity of the feature extraction and information retrieval. Moreover, added noise is also essential to defend users' and IoT devices' identities to prevent the second type of threat, as matching data are hardly done between blockchain data and other supportive databases for processed data. Thus, our model can efficiently improve the privacy-preserving capability.

D. PERFORMANCE ANALYSIS

To evaluate the performance of the proposed model, we conduct experiments in a 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 gives access to the user as described previously in the 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 a 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 [40], [44], [45] and [46].

1) Block size: The block size in Ethereum's is based on the contracts being run and associated 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 in the form of blocks. 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 the 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 executed. We evaluate the interaction delay of register contract and access contract which are important to ensure system effectiveness.

Fig.13 shows the time for one transaction to be completed is less than 30ms in the Register contract and 50ms for the Access contract. Such a delay should satisfy the latency requirement of the real-time applications.

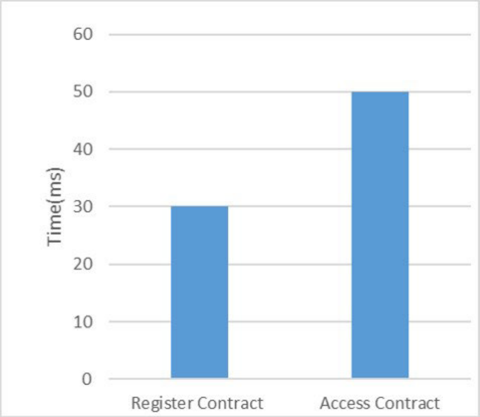


FIGURE 13. Time to complete one transaction.

However, the latency gets worse with register contract as the block size is increased. The latency increases 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, the completion of new block validation and the transmission is faster since the edge server has more computing and bandwidth resource. On the other hand, when comparing with [44], IoT-BC is based on Fabric architecture which in general has a 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%.

The CPU and memory usage are also explored as illustrated in Fig.14. We realize that a very low percentage of CPU resource is taken by the regular transactions while the memory usage is slightly greater since the

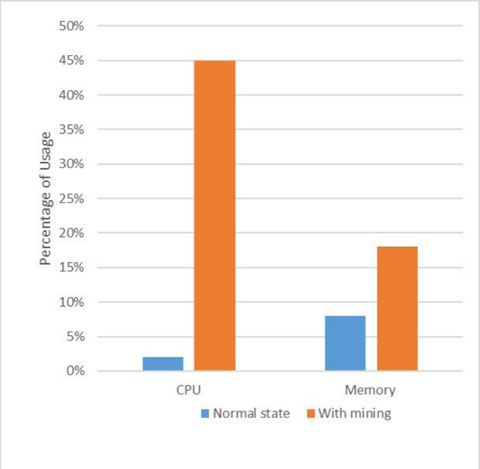


FIGURE 14. Resource usage for single transaction.

TABLE 2. **Calculated gas cost.**

	Proposed Scheme	Scheme in [46]	Scheme in [45]
AddUser	85,662	-	152,863
AddPolicy	360,273	128,777	363,964
DeployACC	1,377,071	1,706,290	1,301,972

blockchain client uses 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 a possible impact on the blockchain overhead. Since the miner is located at the edge server, mining, verifying and storing new blocks will increase the computing resources use. 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.

2) Gas cost: The deployment of smart contracts on the blockchain and execution of these contracts ABIs (Application Binary Interface) require a fee to be paid to the miner which mines the block. A unit called gas is utilized by Ethereum to measure the amount needed to complete a task, e.g. implementing a smart contract or executing an ABI. In general, more gas is consumed with a more complex task. Gas has a price that differs with time. Thus, the fee needed to be paid for completing a task is the result of the amount of used gas and the gas price. Table 2 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 gas amount required for deploying the access contract is 1,377,071, which is more than the existing schemes compared here. We can observe from the table that the proposed ABAC framework in [45] consumes less gas than our scheme. This increased 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.

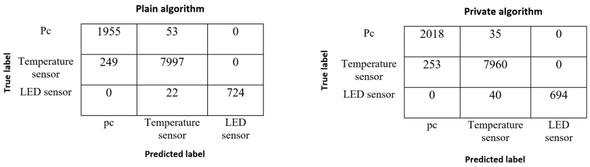
   However, in [46] one ACC is deployed for only one subject-object pair. The gas cost increases linearly as the number of subject-object pairs of the system increases. While in our proposed system there is no need to deploy a new Access contract when the subject and object increase. This results in less 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 the scheme [45].

3) Time cost: The approximate time cost for executing the Access Contract is 40 seconds in our proposal which is, more than 36 seconds of average time for ABAC as presented in [45]. This is due to the time for invoking token in our proposal scheme and the extra time needed to check token validity and call other smart contracts. However, the fresh onetime token generated during each Access request is used for securing the session and this ensures data confidentiality which is worth the difference of few seconds. Note that the execution time of the ABI fluctuates depending on several aspects such as the system's computing power, network architecture, timing of mining, etc. so the execution time may vary within different Ethereum network.

   Furthermore, the time of deploying our access smart contract is around 185.83 seconds compared to the framework deployed in [40]. This is due to smart contract invocations (i.e., getRequest, getRL, uploadResponse, and getResult). Moreover, our framework uses Differential privacy to further improve the privacy and decrease information leak. Differential Privacy is the most suitable technique for big data as it doesn't allow degradation of system's speed compared to other techniques [65].

E. DIFFERENTIAL PRIVACY ENHANCEMENT MODEL EXPERIMENT RESULT

We utilize the confusion matrix as a way of comparing the performance of both machine learning algorithms presented in the previous section as shown in Fig.15, the possible outcomes of a classification, which in our case is either '0', for the PC, '1' for the Temperature sensor or '2' for the LED sensor, against the actual values of the class feature already present in the evaluation (testing) dataset.


Plain algorithm
True label
Predicted label
Private algorithm
True label
Predicted label

	pc	Temperature sensor	LED sensor
Pc	1955	53	0
Temperature sensor	249	7997	0
LED sensor	0	22	724

	pc	Temperature sensor	LED sensor
Pc	2018	35	0
Temperature sensor	253	7960	0
LED sensor	0	40	694

FIGURE 15. **The confusion matrix of device classification.**

There are four parameters presented in the confusion matrix, True Positive (TP), where the classifier has correctly measured the number of packets that are correctly classified to a device type, True Negative (TN), similar to TP but the value of the class feature is negative, False Positive (FP), where the classifier measures the number of packets that are incorrectly classified as a device type and False Negative (FN), which measures the number of packets that are incorrectly not classified as a device type. One metric is created by combining the TP, TN, FP, FN values, namely Accuracy which we can use to evaluate the Classifiers. Accuracy represents the probability that a record is correctly identified as one of the device types. The Accuracy (Overall Success Rate) is calculated using the following equation:

$$OSR = (TN + TP)/(TP + FP + TN + FN) \tag{2}$$

For the classification stage, we use the python in google colab environment for applying a well-known machine learning algorithm. We illustrate the approach using k-fold cross-validation on the neural network model to ascertain the efficiency of our proposed scheme.

Fig.16 shows the accuracy of the model before (plain algorithm) and after (private algorithm) adding the noise.

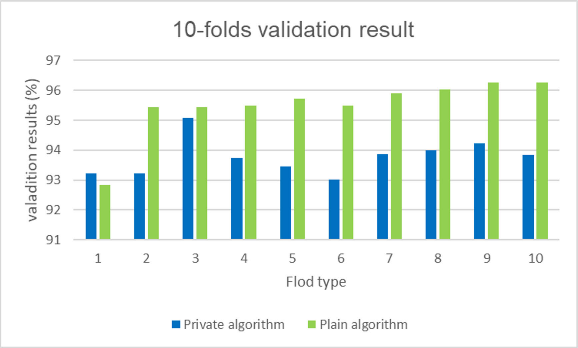


FIGURE 16. 10-fold validation results.

As shown in Table 3, the plain model, has an average accuracy close to 0.95 (95%) while in the private model the accuracy is close to 0.93 (93%).

TABLE 3. Calculated accuracy.

Classifier	Accuracy
Plain algorithm	0.95
Private algorithm	0.93

Our experiment shows that the accuracy of our private model is very close to that of the plain one when the privacy budget is 0.7 because the private method with noise disturbance is relatively small. Therefore, the accuracy of this classification method is close to that of the plain classification method. It is shown in the experiment that, the private model has the same accuracy as the plain model in classifying the device type. Thus, our results demonstrate the feasibility of differential privacy guarantees without significant loss in terms of accuracy. Thus, edge nodes aggregate noisy data to the cloud while preserving smart home privacy and provide accurate data for further analysis.

However, there is a trade-off between accuracy and privacy that directly links to add noise to the scheme. To increase the level of privacy, we increase the amount of noise. But, on the other hand this may result in loss of data accuracy. Therefore, efficient measurements are required to achieve the best result. However, it is outside of the scope of this work and we leave it for future work. In our future work, we will conduct further analysis to measure the differential privacy guarantee to reach improved privacy protection without losing accuracy.

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 scheme to authenticate smart home users and IoT devices. We also integrate differential privacy scheme in our proposed model to preserve data privacy. By combining the blockchain technology with attribute-based access control, differential privacy and edge computing, our proposed model solves the problem of the traditional access control method which is based on the centralized design and meet the access control requirements in IoT. In this paper, we develop Ethereum blockchain, multiple smart contracts and our implementation demonstrates a better performance of our proposed scheme. Compared with the 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 and privacy goals and is resilient against modification, DoS attacks, data mining and linkage attacks. Our work is an ongoing research, and we are currently working on testing our proposed model with differential privacy in a wider scale with different classifier algorithms as a proof of concept. Also, we aim to conduct further research to achieve a better privacy guarantee to highly protected smart home data with better accuracy.

REFERENCES

[1] I. C. Vidal, F. Rousseau, and J. C. Machado, "Achieving differential privacy in smart home scenarios," in *Proc. 34th Anais Principais do Simpósio Brasileiro de Banco de Dados*. Fortaleza, Brazil: SBC, 2019, pp. 211–216.

[2] M. Moniruzzaman, S. Khezr, A. Yassine, and R. Benlamri, "Blockchain for smart homes: Review of current trends and research challenges," *Comput. Electr. Eng.*, vol. 83, May 2020, Art. no. 106585.

[3] Y. Lee, S. Rathore, J. H. Park, and J. H. Park, "A blockchain-based smart home gateway architecture for preventing data forgery," *Hum.-Centric Comput. Inf. Sci.*, vol. 10, no. 1, pp. 1–14, Dec. 2020.

[4] Y. Nakamura, Y. Zhang, M. Sasabe, and S. Kasahara, "Capability-based access control for the Internet of Things: An Ethereum blockchain-based scheme," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[5] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, Jun. 2017, pp. 557–564.

[6] J. Mao, Q. Lin, and J. Bian, "Application of learning algorithms in smart home IoT system security," *Math. Found. Comput.*, vol. 1, no. 1, p. 63, 2018.

[7] 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 Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, Feb. 2019.

[8] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *J. Netw. Comput. Appl.*, vol. 126, pp. 45–58, Jan. 2019.

[9] W. Ejaz and A. Anpalagan, *Internet of Things for Smart Cities: Technologies, Big Data and Security*. Cham, Switzerland: Springer, 2019.

[10] L. Axon, "Privacy-awareness in blockchain-based PKI," CDT Tech. Paper, Oxford, U.K., Tech. Rep. 21, 2015.

[11] Y.-A. de Montjoye, L. Radaelli, V. K. Singh, and A. Pentland, "Unique in the shopping mall: On the reidentifiability of credit card metadata," *Science*, vol. 347, no. 6221, pp. 536–539, 2015.

[12] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy in blockchain technology: A futuristic approach," *J. Parallel Distrib. Comput.*, vol. 145, pp. 50–74, Nov. 2019.

[13] A. Qashlan, P. Nanda, and X. He, "Security and privacy implementation in smart home: Attributes based access control and smart contracts," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 951–958.

[14] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 618–623.

[15] A. Alnemari, S. Arodi, V. R. Sosa, S. Pandey, C. Romanowski, R. Raj, and S. Mishra, "Protecting infrastructure data via enhanced access control, blockchain and differential privacy," in *Proc. Int. Conf. Crit. Infrastruct. Protection*. Cham, Switzerland: Springer, 2018, pp. 113–125.

[16] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the Internet of Things," *Math. Comput. Model.*, vol. 58, nos. 5–6, pp. 1189–1205, 2013.

[17] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, 2006, pp. 89–98.

[18] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Guide to attribute based access control (abac) definition and considerations (draft)," *NIST Special Publication*, vol. 800, no. 162, pp. 1–54, 2013.

[19] Y. Zhu, Y. Qin, Z. Zhou, X. Song, G. Liu, and W. C.-C. Chu, "Digital asset management with distributed permission over blockchain and attribute-based access control," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jul. 2018, pp. 193–200.

[20] S. Rouhani, R. Belchior, R. S. Cruz, and R. Deters, "Distributed attribute-based access control system using a permissioned blockchain," 2020, *arXiv:2006.04384*. [Online]. Available: http://arxiv.org/abs/2006.04384

[21] B. Bera, A. K. Das, M. Obaidat, P. Vijayakumar, K.-F. Hsiao, and Y. Park, "AI-enabled blockchain-based access control for malicious attacks detection and mitigation in IoE," *IEEE Consum. Electron. Mag.*, early access, Nov. 25, 2020, doi: 10.1109/MCE.2020.3040541.

[22] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A user authentication scheme of IoT devices using blockchain-enabled fog nodes," in *Proc. IEEE/ACS 15th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2018, pp. 1–8.

[23] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *Proc. 19th Int. Conf. Adv. Commun. Technol. (ICACT)*, 2017, pp. 464–467.

[24] H. Guo, E. Meamari, and C.-C. Shen, "Multi-authority attribute-based access control with smart contract," in *Proc. Int. Conf. Blockchain Technol.*, Mar. 2019, pp. 6–11.

[25] V. Buterin and F. Vogelsteller. (2015). *ERC20 Token Standard*. [Online]. Available: https://theethereum.wiki/w/index.php/ERC20Token Standard

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

[27] 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, 2018.

[28] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Appl. Models Comput.* Berlin, Germany: Springer, 2008, pp. 1–19.

[29] M. U. Hassan, M. H. Rehmani, and J. Chen, "Privacy preservation in blockchain based IoT systems: Integration issues, prospects, challenges, and future research directions," *Future Gener. Comput. Syst.*, vol. 97, pp. 512–529, Aug. 2019.

[30] E. ElSalamouny and S. Gambs, "Differential privacy models for location-based services," *Trans. Data Privacy*, vol. 9, no. 1, pp. 15–48, 2016.

[31] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[32] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Manubot, Seoul, South Korea, Tech. Rep., 2019.

[33] J. Herrera-Joancomartí and C. Pérez-Solà, "Privacy in bitcoin transactions: New challenges from blockchain scalability solutions," in *Proc. Int. Conf. Modeling Decisions Artif. Intell.* Cham, Switzerland: Springer, 2016, pp. 26–44.

[34] H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on ethereum systems security: Vulnerabilities, attacks, and defenses," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–43, 2020.

[35] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[36] P. K. Singh, R. Singh, S. K. Nandi, and S. Nandi, "Managing smart home appliances with proof of authority and blockchain," in *Proc. Int. Conf. Innov. Community Services*. Cham, Switzerland: Springer, 2019, pp. 221–232.

[37] M. Amadeo, A. Molinaro, S. Y. Paratore, A. Altomare, A. Giordano, and C. Mastroianni, "A cloud of things framework for smart home services based on information centric networking," in *Proc. IEEE 14th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2017, pp. 245–250.

[38] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *J. Cleaner Prod.*, vol. 140, no. 3, pp. 1454–1464, 2017.

[39] Y. Sun, H. Song, A. J. Jara, and R. Bie, "Internet of Things and big data analytics for smart and connected communities," *IEEE Access*, vol. 4, pp. 766–773, 2016.

[40] C. Lin, D. He, N. Kumar, X. Huang, P. Vijayakumar, and K.-K.-R. Choo, "HomeChain: A blockchain-based secure mutual authentication system for smart homes," *IEEE Internet Things J.*, vol. 7, no. 2, pp. 818–829, Feb. 2020.

[41] T. L. N. Dang and M. S. Nguyen, "An approach to data privacy in smart home using blockchain technology," in *Proc. Int. Conf. Adv. Comput. Appl. (ACOMP)*, Nov. 2018, pp. 58–64.

[42] J. Xue, C. Xu, and Y. Zhang, "Private blockchain-based secure access control for smart home systems," *KSII Trans. Internet Inf. Syst.*, vol. 12, no. 12, pp. 6057–6078, 2018.

[43] 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 *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 2116–2121.

[44] J. Ali, T. Ali, S. Musa, and A. Zahrani, "Towards secure IoT communication with smart contracts in a blockchain infrastructure," 2020, *arXiv:2001.01837*. [Online]. Available: http://arxiv.org/abs/2001.01837

[45] M. Yutaka, Y. Zhang, M. Sasabe, and S. Kasahara, "Using ethereum blockchain for distributed attribute-based access control in the Internet of Things," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[46] 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, Apr. 2019.

[47] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 227–236, Jan. 2020.

[48] W. Xiong and L. Xiong, "Smart contract based data trading mode using blockchain and machine learning," *IEEE Access*, vol. 7, pp. 102331–102344, 2019.

[49] X. Chen, J. Ji, C. Luo, W. Liao, and P. Li, "When machine learning meets blockchain: A decentralized, privacy-preserving and secure design," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 1178–1187.

[50] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[51] A. Asuncion and D. Newman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Tech. Rep., 2007.

[52] H. Kim, S.-H. Kim, J. Y. Hwang, and C. Seo, "Efficient privacy-preserving machine learning for blockchain network," *IEEE Access*, vol. 7, pp. 136481–136495, 2019.

[53] M. H. Rehmani, M. Reisslein, A. Rachedi, M. Erol-Kantarci, and M. Radenkovic, "Integrating renewable energy resources into the smart grid: Recent developments in information and communication technologies," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 2814–2825, Jul. 2018.

[54] D. Orazgaliyev, Y. Lukpanov, I. A. Ukaegbu, and H. S. V. S. K. Nunna, "Towards the application of blockchain technology for smart grids in kazakhstan," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2019, pp. 273–278.

[55] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3548–3558, Jun. 2019.

[56] O. Samuel, N. Javaid, M. Awais, Z. Ahmed, M. Imran, and M. Guizani, "A blockchain model for fair data sharing in deregulated smart grids," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–7.

[57] J. Moura and D. Hutchison, "Game theory for multi-access edge computing: Survey, use cases, and future trends," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 260–288, Aug. 2018.

[58] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018.

[59] S. Pavithra, S. Ramya, and S. Prathibha, "A survey on cloud security issues and blockchain," in *Proc. 3rd Int. Conf. Comput. Commun. Technol. (ICCCT)*, Feb. 2019, pp. 136–140.

[60] M. Yang, A. Margheri, R. Hu, and V. Sassone, "Differentially private data sharing in a cloud federation with blockchain," *IEEE Cloud Comput.*, vol. 5, no. 6, pp. 69–79, Nov./Dec. 2018.

[61] Y. Zhao, J. Zhao, L. Jiang, R. Tan, and D. Niyato, "Mobile edge computing, blockchain and reputation-based crowdsourcing IoT federated learning: A secure, decentralized and privacy-preserving system," 2019, *arXiv:1906.10893*. [Online]. Available: https://arxiv.org/abs/1906.10893

[62] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2013, pp. 245–248.

[63] S. K. Dwivedi, R. Amin, S. Vollala, and R. Chaudhry, "Blockchain-based secured event-information sharing protocol in internet of vehicles for smart cities," *Comput. Electr. Eng.*, vol. 86, Sep. 2020, Art. no. 106719.

[64] S. K. Dwivedi, R. Amin, and S. Vollala, "Blockchain based secured information sharing protocol in supply chain management system with key distribution mechanism," *J. Inf. Secur. Appl.*, vol. 54, Oct. 2020, Art. no. 102554.

[65] S. H. Begum and F. Nausheen, "A comparative analysis of differential privacy vs other privacy mechanisms for big data," in *Proc. 2nd Int. Conf. Inventive Syst. Control (ICISC)*, Jan. 2018, pp. 512–516.

AMJAD QASHLAN received the master's degree in information and communication technology from the University of Wollongong, in 2012. She is currently pursuing the Ph.D. degree with the Faculty of Engineering and IT (FEIT), University of Technology Sydney (UTS). Her doctoral research investigates developing a blockchain security solution for smart home systems. She is examining the use of blockchain technology and machine learning in order to increase the IoT smart home network security and privacy.

PRIYADARSI NANDA (Senior Member, IEEE) is currently a Senior Lecturer with the University of Technology Sydney (UTS), with more than 27 years of experience specialising in research and development of cybersecurity, the IoT security, Internet traffic engineering, wireless sensor network security, and many more related areas. His most significant work has been in the area of intrusion detection and prevention systems (IDS/IPS) using image processing techniques, sybil attack detection in the IoT-based applications, and intelligent firewall design. In cybersecurity research, he has published over 80 high quality refereed research articles, including IEEE Transactions in Computers, IEEE Transactions in Parallel Processing and Distributed Systems (TPDS), *Future Generations of Computer Systems* (FGCS), and many ERA Tier A/A conference papers. He has successfully supervised eight HDR at UTS (five Ph.D. and three Masters) and currently supervising eight Ph.D. students. In 2017, his work in cyber security research has earned him and his team the prestigious Oman Research Council's National Award for Best Research.

XIANGJIAN HE (Senior Member, IEEE) is currently the Leader of the Computer Vision and Pattern Recognition Laboratory, Global Big Data Technologies Centre (GBDTC), University of Technology Sydney (UTS). He was an IEEE signal processing society student committee member. He was involved in a team who received a UTS Chancellor's Award for Research Excellence through Collaboration, for a project funded by SydneyTrains and RMCRC, in 2018. He has also been awarded Internationally Registered Technology Specialist by International Technology Institute (ITI). He led UTS-PolyU joint research project teams wining the 1st Runner-Up Prize for the 2017 VIP Cup, and the Champion for the 2019 VIP Cup, awarded by IEEE Signal Processing Society. He has been carrying out research mainly in the areas of image processing, network security, pattern recognition, computer vision, and machine learning in the previous years. He has recently been leading his research teams for deep-learning-based and/or machine-learning-based research in the areas of human behavious recognition, human counting in a crowd, tiny object detection, 3D medical image restoration, image processing based on hexagonal structure, authorship identification of a document and a document's components, such as sentences and sections, network and cyber security, car license plate recognition of high speed moving vehicles with changeable and complex background, and video tracking with motion blur. He has played various chair roles in many international conferences, such as ACM MM, MMM, ICDAR, IEEE BigDataSE, IEEE TrustCom, IEEE CIT, IEEE AVSS, IEEE ICPR, and IEEE ICARCV.

MANORANJAN MOHANTY received the Ph.D. degree in computer science from the National University of Singapore, Singapore, in 2014. He is currently a Lecturer with the Center for Forensic Science, School of Mathematical and Physical Science. He comes from computer science background. His research interests include digital forensics and cybersecurity, with current focus mainly on source camera attribution, child explicit content detection, fake food detection, privacy-aware forensics, cloud and the IoT forensics, and application of deep learning and blockchain for forensics. After that, he spent a year as an ERCIM Alain Bensoussan Research Fellow with SICS Swedish ICT, Sweden, and two years as a Research Fellow with New York University. Before joining UTS, he was a Lecturer in digital security with the University of Auckland, New Zealand.