




Research Article

IoT Devices, User Authentication, and Data Management in a Secure, Validated Manner through the Blockchain System

Talha Ahsan ¹, Farrukh Zeeshan Khan ¹, Zeshan Iqbal ¹, Muneer Ahmed,²
Roobaea Alroobaea ³, Abdullah M. Baqasah ⁴, Ihsan Ali ⁵,
and Muhammad Ahsan Raza ⁶

¹University of Engineering and Technology, Taxila 47080, Pakistan

²School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Sector H-12, 44000 Islamabad, Pakistan

³Department of Computer Science, College of Computers and Information Technology, Taif University, P. O. Box 11099, Taif 21944, Saudi Arabia

⁴Department of Information Technology, College of Computers and Information Technology, Taif University, P. O. Box 11099, Taif 21944, Saudi Arabia

⁵Department of Computer System and Technology, Faculty of Computer Science and Information Technology, Universiti Malaya, Kuala Lumpur, Malaysia

⁶Department of Information Technology, Bahauddin Zakariya University, Multan 60000, Pakistan

Correspondence should be addressed to Ihsan Ali; ihsanalichd@siswa.um.edu.my

Received 5 October 2021; Revised 28 December 2021; Accepted 7 January 2022; Published 8 February 2022

Academic Editor: Samarendra Nath Sur

Copyright © 2022 Talha Ahsan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Advancement in technology has led to innovation in equipment, and the number of devices is increasing every day. Industries are introducing new devices every day and predicting 50 billion connected devices by 2022. These devices are deployed through the Internet, called the Internet of Things (IoT). Applications of IoT devices are weather prediction, monitoring surgery in hospitals, identification of animals using biochips, providing tracking connectivity in automobiles, smart home appliances, etc. IoT devices have limitations related to security at both the software and hardware ends. Secure user interfaces can overcome software-level limitations like front-end-user interfaces are accessed easily through public and private networks. The front-end interfaces are connected to the localized storage to contain data produced by the IoT devices. Localized storage deployed in a closed environment connected to IoT devices is more efficient than online servers from a security perspective. Blockchain has emerged as a technology or technique with capabilities to achieve secure administrative authentication and accessibility to IoT devices and their computationally produced data in a decentralized way with high reliability, interrogation, and resilience. In this paper, we propose device, end-user, and transactional authentication techniques using blockchain-embedded algorithms. The localized server interacts with the user interface to authenticate IoT devices, end-users, and their access to IoT devices. The localized server provides efficiency by reducing the load on the IoT devices by carrying out end-user heavy computational data, including end-user, IoT device authentication, and communicational transactions. Authentication data are placed on the public ledger in block form, distributed over the system nodes through blockchain algorithms.

1. Introduction

With the rapid growth of smart gadgets and high-speed networks that are used for communication for these smart devices, the Internet of Things (IoT) has gained human attention and popularity in the past few years. These embedded devices or IoT

devices connect through public or private networks, are accessed remotely, and perform the desired functionality. Public and private networks use networking protocols for sharing information and communicating among the IoT devices.

IoT devices aid humans by performing various functions such as detecting weather conditions, supporting hospital

equipment for operations, identifying animals using bio-chips, and providing tracking and connectivity in automobiles. IoT servers gather data from these devices in real time and process the data to enhance the efficiency of the system.

Internet of Things (IoT) is being deployed at a large scale around the world, with Corps Information System Control officers (Cisco) predicting 40 billion devices at the end of the year 2021 [1]. Internet of things (IoT) are resource-consuming appliances and are not capable of fixing and protecting themselves against malicious attacks like Man in the middle attack, masquerading, DOS attacks, etc., and can be easily hacked by hackers. Due to this deficiency, everybody can easily access IoT devices and perform computations accordingly. Therefore, it is the present day need that enhances the security of the IoT devices; for this, it is essential to adopt proper methods for the user as well as device authentication and computational transaction to verify that IoT devices are secure in every respect. There is also the demand for the system to ensure the interaction between end-users and IoT devices. End-users are mapped on IoT devices through networking protocols [1]. Any proper user and IoT device authentication scheme must recognize the reality that these IoT devices are service-constrained appliances and unable to execute heavy transactions and processing. The user and device authentication techniques must be authentic, capable of being scaled, and reliable against multiple threats and attacks.

Numerous authentication techniques [2] are designed and deployed to provide the security to IoT devices, but these are all based on centralized architecture and depend on a centralized authority like database or servers of the system. Centralized authority verifies the end-users, system IoT devices, and communication record between end-users and IoT devices by using different protocols. Mutual authentication, certificate-based authentication, and token-based authentication are all centralized authentication techniques. These techniques have many flaws such as high transactional computational costs, centralized trusted third parties, single point of failure, lack of privacy, and the likelihood of hacking. Because these techniques rely on the trusted third party in this way, double dependency problems occur. Figure 1 describes the double dependency problem.

To reduce the flaws of the centralized (trusted third party) authentication of IoT devices, a decentralized end-user, IoT devices, and transaction authentication scheme are proposed using algorithms that provide blockchain technology. The proposed system provides the facility of end-user and IoT device authentication. The proposed system also facilitates end-users with the secure communication mapping to the IoT devices while ensuring security without any requirement of a centralized identity.

The fundamental purpose of this research is to furnish the security of an end-user, IoT devices, and the interaction between them in a decentralized manner. Particularly, we present a whole system that consists of a design and architecture involving IoT devices, end-users, and blockchain

algorithms that apply authentication rules and deploy the blockchain algorithmic logic into the public area network. Furthermore, the main objectives and contributions of this research can be abridged below:

Our main objective in this paper is to provide hardware-level security to IoT devices. For the achievement of this goal, we need to use blockchain technology. Blockchain technology consists of decentralized techniques rather than all other techniques.

- (1) We present a reliable, scalable, and authentic decentralized end-user and IoT device authentication technique that utilizes a graphical user interface with connectivity to blockchain algorithms. These algorithms consist of the logic that authenticates end-user access to IoT devices and they also authenticate the devices that are accessible by the end-user. Through these, issues of a centralized third party and the double dependency problem can be removed.
- (2) We describe the detailed analysis of the whole system that constitutes the system entities, sequence flow diagram, blockchain algorithm (smart contracts), and interactions between the graphical user interface and participants in the algorithms.
- (3) We present an analysis on the security of our proposed authentication technique and discuss how the proposed technique achieves security goals (of confidential, integrity, and availability), and can overcome eavesdropping, replay, masquerading, denial of service (DoS), and Man in the Middle attacks.
- (4) We achieve prevention of the denial-of-service attack through blocking an intruder in the system. If an intruder wants to access the system multiple times with the wrong hash key value, then the system identifies the intruder identity and blocks it.

The research paper is organized as follows. We discuss in Section 2 the IoT Security challenges, and in Section 3 we will present the proposed solutions of these challenges. Both these sections are part of the literature review. An overall description of the system architecture is presented in Section 4, including the interaction between system entities. Section 5 contains the experimental work which consists of a detailed description of the proposed methodology with algorithms. Section 6 presents the evaluation and results that are computed from the proposed work. In Section 7, we discuss the security analysis. The conclusion and future work is given in Section 8.

2. Security Issues in IoT

With the gradual increase in the number of IoT devices and the passage of time and equipment ranging from small embedded processing chips to large high-end servers, it needs to address many security issues at different

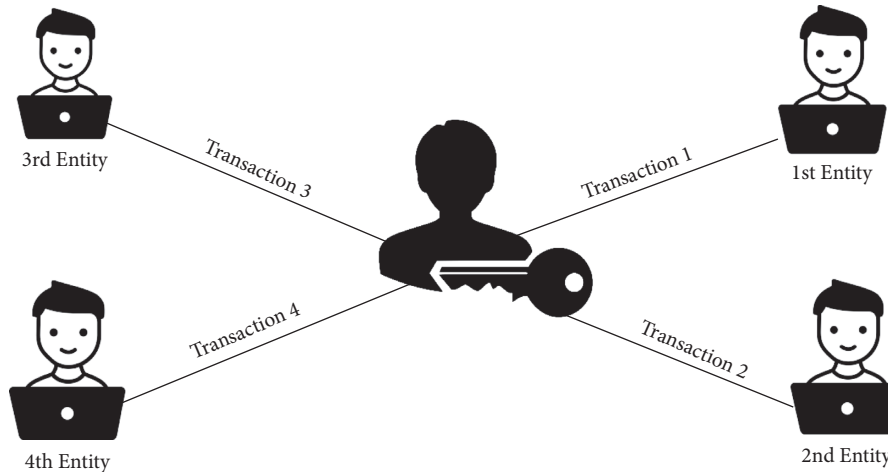


FIGURE 1: Owners of the system can send the same transaction containing digital currency to the multiple participants of the system.

architectural levels of embedded IoT devices. We categorize the security threats/issues concerning the IoT device deployment architecture as described below:

- (1) Low-level security issues
- (2) Intermediate-level security issues
- (3) High-level security issues

2.1. Low-Level Security Issues. The level of security issues is concerned with Physical (layer one) and DLL (layer two) layers. Low-level security issues are also concerned with the hardware of the security issues.

In jamming attacks, radio frequency signals are emitted without following specific protocols in the wireless IoT devices [3, 4]. These radio frequency signals impact on the operation of the network and they also impact the sending or receiving of data through insecure nontrusted nodes, resulting in an unpredictable behavior of the system.

The Sybil attacks in a wireless network for accessing the IoT devices due to the presence of Sybil nodes in the network which produces fake identification and acts as part of the system to utilize the services of the IoT. At the first layer, a Sybil node utilizes fake addresses of the device's port like MAC values for masquerading [5, 6].

An insecure physical interface means managing poor physical security and not recognizing it from the considerations. The poor physical security tools for debugging/testing software and access through physical interfaces may breach the security through compromise nodes in the network. Though these, nodes can access the service of the IoT devices and perform some maliciousness [7, 8].

2.2. Intermediate-Level Security Issues. The intermediate-level security issues are concerned with network and transport layer communication, routing, and session management.

There is a need for IoT deployment architecture to be identified with every IoT device uniquely in the network. In neighbor discovery, data are transmitted in different steps including router discovery and address resolution [9].

Without proper verification, the utilization of packets got through neighbor discovery may have breached the security. Neighbor discovery packets also cause the occurrence of denial-of-service (DoS).

The Internet Protocol version 6 is routing protocol which generates unsecure networks. Compromised nodes in the network breach the security of the whole network and allow intruders to perform malicious attacks onto the network [10].

IoT devices and end-users are both required to add more security through hash key values or other security techniques. The fourth layer of the OSI model determines transmission pathways. Because of security flaws in the network's routing layer, data is sent in the wrong direction [11–13]. Datagram Transport Level Security and overhead, due to the available resources, need to be minimized [14].

Communicational sessions have been established between two entities of the system at the time of communication. Sessions can hijack on the fourth layer of the networking protocol with the help of fake links. Through this, some maliciousness occurs in the network in terms of denial of services [15, 16]. A session establishes between two nodes, an attacking node, and a victim node. The communicating nodes may even require retransmission of messages by changing the sequence number of the messages.

2.3. High-Level Security Issues. High-level security issues occur on the application layer. Application layer security issues are described below.

The interfaces that are used for accessing IoT services, web, mobile, and cloud can be affected by different attacks which may also affect data privacy [17].

The middleware of Internet of things is developed for the interaction between system entities, so these heterogeneous entities must be secure while service provisioning. Different environments and interfaces using middleware need to provide secure communication [18, 19].

High-level security issues occur when IoT devices are connected to the Internet. IoT devices' front-end interfaces are connected to the Internet; if they compromise, then large loss of data or information occurs. Therefore there is a need

to test XML SQL XSS carefully in which front-ends are designed for access of IoT devices. Constrained application protocols are used for communication between IoT devices and interfaces. CoAP provides end to end communication.

3. Solutions to Security Issues

In the literature, significant efforts have been made to address the security issues discussed in the previous section. Client authentication is the process that substantiates user recognition through a set of accreditations which recognize the data stored in an authentication server or database [3, 20]. In terms of mutual authentication, both the system entity user and the server take part to identify and recognize each other, the server authenticates the user because the user is stored in the server, whereas a user authenticates its presence in the server [21, 22]. Mutual authentication is classified into two types, one is username-/password-based authentication and the second one is certificate-based authentication [22]. Username-/password-based authentication and certificate-based authentication minimize many threats and risks of hacking in various computational processes like shopping websites, ensuring computational transactions with clients and servers for authorized purposes.

Open Authorization (OAuth) is the most eminent and comprehensively used identification and authentication technique for IoT device security. OAuth uses an open standard communicational protocol that provides tokens to end-users and IoT devices. Tokens are stored on the server or database. The system's resources are used by end users. End-users are authenticated in the system using tokens [23, 24]. The open standard protocol consists of four actors: The data and resource owner who generates validated resources and provides the access of the server to the users. The Open Authentication server (OAS) that provides tokens for secure communication with authentic clients/users or any other entity. The resource server or database which provides authenticated resources/data. The user who wants to access services from the server. The open authorization process is described in 6 steps. In the first step, the client generates requests to the resource owner for a successful access to the authenticated resources. In the second step, the username and password are set for the client by the authorization server. In the third step, the authentication of clients with the username and password, the client generates a request to the authorization server for providing access tokens. In the fourth step, the secure server identifies the password as well as the username of the client and assigns a protocol-generated access token to the authenticated clients. The token which is provided to the user consists of the public and private key values. In the fifth step, after getting the access tokens, the client generates access transaction for protected resources and sends it to authorization server or wants to execute any transactions by using the protocol generated token. In the sixth step, the authorization server authenticates the token, if the verification is successful, then protected resources or IoT devices are provided to the clients to perform computations.

A famous third entity technique, Kerberos authentication system, narrated in Ref. [25] utilizes temporal tickets for

user authentication. These temporal tickets have many issues to authenticate clients and servers because these are exits for a specific period.

Delegation server is also a well-known technique for the authentication of the user and device [26]. Delegation server incurs high computational costs to authenticate devices through delegation servers every time a user needs to access new values for authentication purposes.

Another authentication technique: Mahalle et al. [27] introduced a process in which a set of user authentication protocols is used for user authentication. Group authentication protocol generates keys that are shared among multiple nodes on the network. Due to key sharing among multiple nodes in the network, many risks occur if one of the nodes breaches the security by sharing the key and creates security holes.

For distributed IoT systems, the technique involves [28] proposing that certificates are used in terms of an authentication protocol. In certification-based authentication protocols, cryptographic techniques are used for identification like hashing key values stored in the server node. Although cryptography certification-based authentication protocol provides much better security than existing techniques, many limitations arise due to the centralized architecture. With a centralized authentication architecture, the system cannot be secure in terms of redundancy, reliability, single point of failure attack, and scalability.

The drawback of user authentication is that they authenticate users with only the username/password. Therefore, usernames and passwords can be easily breached.

The open authentication technique for user authentication is a centralized technique in which all computations are performed across a central identity.

Group authentication technique shares the authentication technique across multiple nodes, but the group authentication technique does not use hash key values, so every entity in the system can perform some maliciousness.

All the abovementioned techniques have many deficiencies, and they only authenticate users rather than authenticating IoT devices and computational transactions for the communication between the user and IoT devices. They also used centralized techniques for authentication. In centralized technique, all computational data are stored on to the server. Entities in the system rely too heavily on the server or central authority to complete desired access transactions. If the centralized authority is nontrusted, then it can breach the security of the whole system. There is a research gap that exists with respect to complete user, device, and computational transaction authentications in decentralized manner.

4. Blockchain-Based Authentication of IoT Devices, End-User, and Transaction between Them

This part of the paper describes the various aspects of the architecture and detailed design of our proposed research blockchain-based IoT devices and end-user registration and

validation system in which blockchain algorithmic logic will be used to identify consumers and available IoT devices in a secure and validated process.

4.1. System Architecture. There are five major entities in the proposed system architecture with access to web-based algorithms through the Internet: Admin, IoT devices, end-users who facilitate direct connection with the system, and MySQL server containing end-users, IoT devices, and communicational data. During registration we assign unique hash addresses (public and private keys) to IoT devices and end-users. Both IoT devices and the end-users are registered on the web. Admin and databases are also part of the system. Detailed architecture of the system is presented in Figure 2.

This summarizes the whole system entities as follows:

4.1.1. Admin. Admin is the most valuable entity of the system and is responsible for user access control, list of users, IoT device services, and permission to end-users for accessing IoT devices. Only the owner of a particular organization or architectural system has the services of management and access control. The primary client within the framework is the proprietor or the maker of the blockchain algorithm. The owner of the blockchain algorithm can add IoT devices per user request as a part of the system. Admin also gives permission through the blockchain algorithm for end-users to access IoT devices. Admin has the ability to block new transactions in the system and add the block into the chain. The very first entity in the system is the admin, so its block does not have a previous hash value.

4.1.2. END_USERS. Within the framework of the system, utilizers are clients who ask for consent from the blockchain algorithm to access particular IoT devices. Once end-users are allowed to get authorization after authentication via the authentication algorithm, they contact the designated server node capable of governing the desired IoT device for authentication and access.

4.1.3. Blockchain Algorithm. The blockchain algorithm allows the authenticated utilizer access to the authenticated smart device. Registration of end-users, smart devices, access control, authentication, and functionalities are deployed to become a centralized architecture through the blockchain algorithm.

4.1.4. Database. In our solution, the database is utilized in overseeing access to IoT devices. Database stores IoT devices, information, end-user data, and computational transactions in the form of public ledgers. The database is distributed among all end-users in the form of a distributed ledger, but they cannot change, delete, or update any records; they can only create their own transaction record.

4.1.5. IoT Devices. The smart appliances in the system are expected to be a resource-consuming device with restricted storage, processing capacity, and memory.

4.2. Interaction between Entities. The interaction between the system entities happens in two major steps, namely, online and off-line interactions. Figure 3 shows a sequence between end-user and IoT devices for successful authentication of the user as well as user access to the IoT devices. A secure session is established for a secure connection between end-users and smart devices. In the online interaction, the admin initially generates the algorithm or smart contract and registers the user into the system and maps it to a MySQL server through select functions. Unique private and public addresses are assigned to the users through algorithms. Admin adds the devices as well into the system and also assigns unique addresses through algorithms and stores them in the MySQL server. That is why authenticated users can access authenticated devices that are part of the system.

When the user successfully authenticates (gains its unique public and private keys) and needs to get a specific IoT device, the user initially transfers the authentication request to the blockchain algorithm by using the registration request. The algorithm will recognize the SQL server of authenticated smart device for that end-user. If the device is unauthentic or the client is not eligible to facilitate the services of that device, then the system rejects the request of the user. Otherwise, if both the IoT device and user are valid and part of the system entities, the blockchain algorithm will allocate access permission to the end-user to access the authorized device and store the access information in the form of an encryption into the SQL server and broadcast it to all users, publicly.

When a user successfully registers into the system through a smart contract, then it stores the user's unique identification (ID), public address (PA), private address (PA), and previous public hash values. A block is generated, and it contains the user public key, transactional data, and previous hash values. Users' public and private addresses are stored in the form of hash key values. SHA-256 hashing key algorithm is used to generate hash key values. 64-bit key values generated by SHA-256 are in encrypted form.

Firstly, users register into the system and get public and private hash key values which are stored in the SQL server, then the user logs into the system by using the public key. The smart contract identifies whether it is a valid user or not. If the user is unauthorized, its request is rejected with an error; otherwise; the user successfully enters into the system with public and private hash key values.

5. Experimental Setup

In the experimental setup, we highlight the key usage viewpoints associated with the algorithm/smart contract of security. The existing blockchain platforms like Ethereum, Ripple, and R3 facilitate the development of apps on Blockchain networks, but these are all paid projects. In the case of Ethereum, we should buy eth currency and spend eth

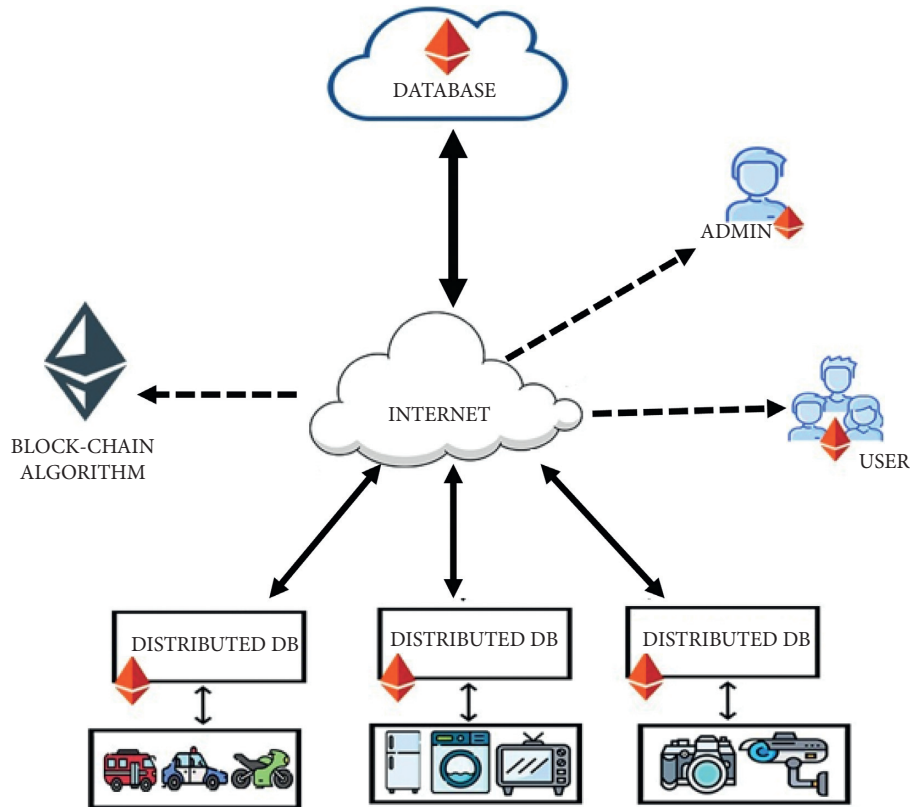


FIGURE 2: Blockchain-based proposed system architecture.

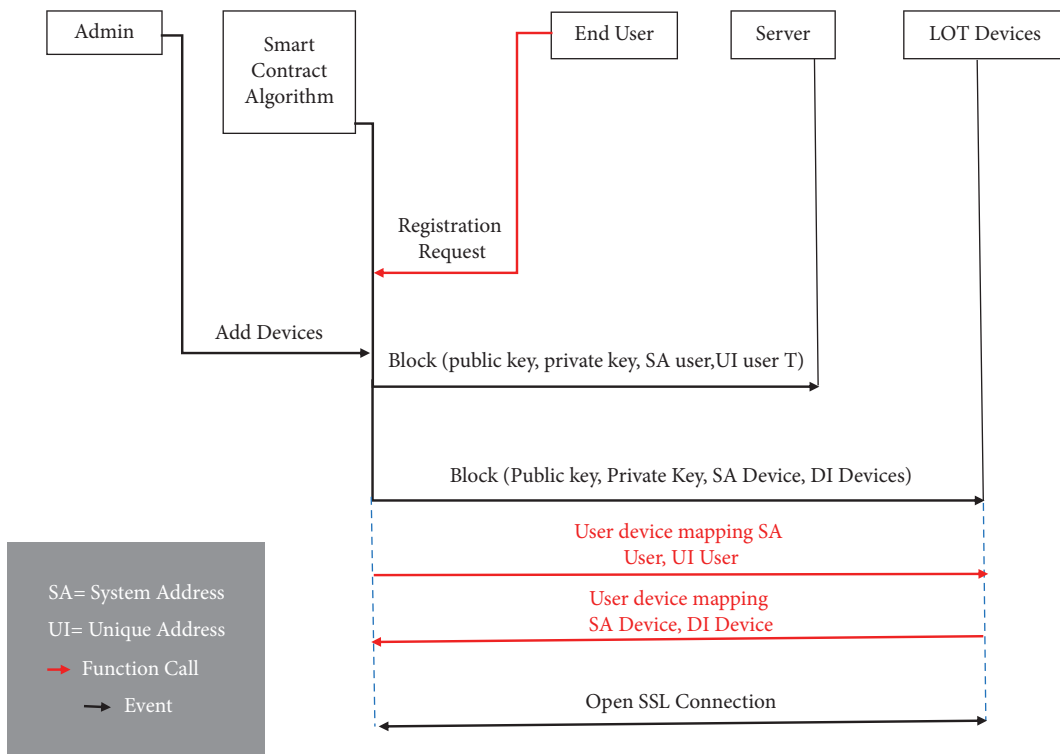


FIGURE 3: Sequence diagram of successful authentication and communication between end-user and IoT devices.

to perform each computation. Therefore we implement the decentralized logic of blockchain and develop a decentralized application in which end-user and IoT devices registration and authentication is performed in a decentralized manner through a distributed ledger. The implemented smart contract includes three main components: (1) Admin authority and end-user registration and authentication, (2) IoT devices registration, (3) user transactional authentication to IoT devices.

The algorithm was executed and tested using the MySQL server [15], which offers interesting highlights that encourage testing. Our main focus is on the execution of on-chain and off-chain parts of the proposed work which includes authentication components.

5.1. Admin Authorities and End-User Registration and Authentication

5.1.1. Admin Authorities. Smart contract I describes the admin authorities in which other end-users are restricted to add devices into the system. The very first entry of the system contains the address of the public hash value, but it does not contain the address of the previous hash value. Apart from admin, other end-users have previous hash values that can be used to add the next record into the system, but they cannot add IoT devices into the system.

SMART CONTRACT I. Admin authority of only admin can add devices into the system.

```
(i) session_start();
(ii) //connect to the database
(iii) $db = mysqli_connect('localhost', 'root', '',
    'registration');
(iv) if (!isset($_SESSION['username'])) {
(v) $_SESSION ['msg'] = "You must log in first";
(vi) header ('location: login.php');}
(vii) if (isset ($_GET ['logout']))
(viii) {
(ix) session_destroy();
(x) unset ($_SESSION['username']);
(xi) header ("location: login.php");
(xii) }
(xiii) $_SESSION ['username'];
(xiv) $favcolor = $_SESSION ['username'];
(xv) switch ($favcolor)
(xvi) {
(xvii) case "admin":
(xviii) echo "Add device";
(xix) break;
(xx) default:
    echo "Only Admin Can Enter Device"; header
    ("location:msg.php");
}
```

5.2. User Registration and Authentication. Smart Contracts II and III present the function through which users register into the system. A list of client hash addresses being displaced in the system might have different users as compared to the initial added users being the owner of the algorithm. A block is created to recognize data related to user authentication in the system. As stated earlier, a block is generated with the verification or authentication of the end-user to get the services of an IoT device. A list of blocks is generated to keep the record of all end-user-generated blocks. These blocks save the address of the user's hash key values (public key, private key) to the server. These hash key values are assigned to the users during the registration of the user.

The block of the very first user does not contain the previous public hash value of the user because the very first user is the owner of the algorithm. The rest of the users who are part of the system contain the previous user public hash value. In this regard, blocks are related and make a chain. Smart Contract IV presents the function to generate the user's previous public hash value.

```
Public key
function generateRandomString ($length = 64) {
    $characters = '0123456789abcdefghijklmnopqrstuvwxyz';
    $charactersLength = strlen ($characters);
    $randomString = "";
    for ($i = 0; $i < $length; $i++) {
        $randomString .= $characters [rand (0, $character-
            sLength - 1)];
    }
    return $randomString;
}

Private key
function generateRandomString1 ($length = 64)
{
    $characters = '0123456789abcdefghijklmnopqrstuvwxyz';
    $charactersLength = strlen ($characters);
    $randomString = "";
    for ($i = 0; $i < $length; $i++)
    {
        $randomString .= $characters [rand (0, $character-
            sLength - 1)];
    }
    return $randomString;
}

Getting previous hash value
$query = "SELECT * FROM users ORDER BY id
DESC LIMIT 1";
$result = mysqli_query ($db, $query);
while ($row = mysqli_fetch_array($result))
```

```
{
$publickey = $row ['publickey'];
}
```

5.3. IoT Devices Registration and Authentication. The communication, networking, and connectivity protocols used on Internet-enabled devices mainly depend on the specific Internet of Things applications deployed. The communication protocols that are used to provide the services of the IoT devices include CoAP, MQTT, and DTLS, among others. Wireless protocols include IPv6, LPWAN, Z-Wave, Bluetooth Low Energy, Zigbee, RFID, and NFC. Wi-Fi, Cellular, satellite, and Ethernet can also be used for communication. IoT devices are deployed on the server site and can be accessed through the abovementioned communication protocols.

```
IoT devices registration into the system
function generateRandomString ($length = 64)
{
$characters = '0123456789abcdefghijklmnopqrstuvwxyz
wxyzABCDEFGHIJKLMNopqrstuvwxyz';
$charactersLength = strlen ($characters);
$randomString = "";
for ($i = 0; $i < $length; $i++)
{
$randomString .= $characters [rand(0, $character-
sLength - 1)];
}
return $randomString;
}
```

Each option has its tradeoffs like power consumption, bandwidth, and range, all of which must be considered when IoT device services are provided through protocols for particular IoT applications.

Smart Contract V presents the device registration function in the system. Two types of hash addresses are presented during the registration of the IoT device. A block is created to save information related to IoT device authentication in the system. The block contains the server address and hash key values that are assigned to the IoT device for authentication. These blocks save the address of the device hash key values (public key, private key) to the server. IoT devices are added into the system through the AddDevice function call. AddDevice function calls can be accessed only by the admin.

5.4. User Transactional Authentication to IoT Devices. The user's access to a list of devices is mapped through UserDeviceMappingFunction. When an end-user wants to access the services of IoT devices, then UserDeviceMapping function authenticates if it is a valid user. Otherwise, the UserDoesnotValid function is executed, which indicates that the end-user is not valid. Only those IoT devices that can be added into the system by admin to the server can be accessed

by the end-user. In addition, only those users that are part of the system and have valid hash key values can access IoT devices.

User authentication and IoT device authentication also provide two-way authentication, also known as 2-Factor authentication. After the authentic transaction of the end-user map to the IoT device, a block is generated in the distributed ledger. A distributed ledger contains a block of data. Each block contains the public address of the user and the IoT device.

```
Nonauthenticated request rejection
session_start();
//connect to the database
$db = mysqli_connect ('localhost', 'root', "",
'registration');
if (!isset ($_SESSION ['username']))
{
$_SESSION ['msg'] = "You must log in first";
header ('location: login.php');
}
if (isset ($_GET ['logout']))
{
session_destroy();
unset ($_SESSION ['username']);
header ("location: login.php");
}
Nonauthenticated request rejection
session_start();//connect to the database
$db = mysqli_connect ('localhost', 'root', "",
'registration');
if (!isset ($_SESSION ['username']))
{
$_SESSION ['msg'] = "You must log in first";
header ('location: login.php');
}
if (isset ($_GET ['logout']))
{
session_destroy();
unset ($_SESSION ['username']);
header ("location: login.php");
}
```

6. Evaluation and Results

In this section, we essentially pay attention to testing functionality among system entities along the web algorithms deployed in the web server environment. We assign unique Hash Addresses (HA) for multiple system entities. Three main functionalities of the system participants were tested, including end-user authentication, IoT device authentication, and transactional authentication operations.

When attempting to add an end-user through a web server, to access IoT devices, the request produces a hashing address that contains the public and private addresses, and successful addition of IoT devices. Table 1 shows the blocks generated through the user registration algorithm for the authenticated end-users. When an end-user is authenticated and wants to perform a communicational transaction, then end-user record is added into the block. In the same way, the next user's additional requests would facilitate in a similar generated block.

In end-user test scenario, a ledger is generated for end-users. Table 1 consists of ledger for the end-user. A ledger is distributed over the network. A distributed ledger contains blocks that are connected to each other with the previous hash key values. Each block in the distributed ledger contains end-user hash key values and the previous block public hash key values. All information or data in each block of the ledger is in encrypted form, so every end-user can access every block but cannot change, update, delete, or alter data. Through end-user distributed ledger, a secure and validated end-user can access the system IoT devices.

In IoT devices test scenario, IoT devices are also registered in the system. End-users can access only the devices that are registered into the system. IoT device registration data are stored in the distributed ledger. Table 2 shows distributed ledger of registered IoT devices.

IoT devices are authenticated in the same way as an end-user is registered in the system. Only the admin can add devices into the system. When a device adds into the system, a block is generated which assigns the hash key values to the IoT device. The block with IoT device registration is stored in a public distributed ledger. Public distributed ledgers are available for all authenticated users in the system, but they cannot change or update any block. When the end-user accesses IoT devices with its public key address, it can be authenticated with the private hash key value of the IoT device.

When the end-user maps to IoT devices, a transaction with the user request is generated. This transaction creates a block on the server which contains hash values like public key and previous hash value. These hash values are generated with the user data and their ids. The block is distributed over the system through which every end-user can check it, but it cannot be changed as it contains hash values. A distributed ledger shows which user interacts with which device. Table 3 shows authenticated transactions between IoT devices and end-users.

Through algorithm execution, we obtained an output to show that the end-user was successfully mapped onto the IoT devices. The first scenario implies that the user is nonauthorized. Therefore, the services of IoT devices cannot be provided by end-users. In this explanation, if an unauthenticated end-user, which not registered in the system, wants to access IoT devices from the system, then its request is rejected because it has no hash key values.

In the second scenario, the client endeavor is to access an IoT device, but it is not available to the system of IoT devices provided by the admin that is authorized to access. In this scenario, the algorithm allows the user to access only those

devices which are mapped to the server. In the explanation of the second scenario, if an authenticated registered end-user in the system wants to access IoT devices which is not the part of the system, then two situations occurs. In the first situation, authenticated end-users ask the admin to add the required IoT devices. In the second situation, end-user request is rejected for IoT device if the device is not available.

The third scenario is for a successful authenticated transaction in which the user maps to the IoT device and its related server node. In this scenario, the algorithm successfully shows a function that contains information as the transaction mines and the execution succeeds. Authentic computational transactional information is stored in Table 4. A distributed ledger is generated for successful transactions. Blocks are created on each transaction. These blocks are linked to the previous block public hash values. Through the distributed ledger, everyone gets to know which IoT devices mapped to the end user and are currently not available. With the help of the distributed ledger, we can implement the blockchain techniques to provide security to IoT devices, end-users, and the communication between them.

7. IoT Security Analysis

IoT security issues arise at both the hardware level and the software level. Our proposed system is used to reduce different security issues at both levels. Table 5 consists of comparison between existing techniques and proposed techniques. The security achievements of integrity, availability, and confidentiality can be accomplished through authentication of each entity in the system, encryption and decryption of data, and access control schemes.

Privacy constrained in the system is attained by performing an authorized user approach to the smart device and its data. The confidentiality is also gained by performing different types of hashing techniques using SHA-256 cryptography for a successful user, IoT device, and transaction (user access to the desired IoT device) authentication. Based on hash key values, a blockchain-based authentication architecture is proposed. Hash key values are distributed over the network via a public distributed ledger. When the end-user becomes a part of the system, it assigns the public and private key addresses. These hash key values also assign to the IoT devices at the time of admin registration of the devices.

These hash key values are unique at every time. Therefore, no collision occurs in the system, and this is a powerful feature of blockchain. Unique hash key values establish a secure session for the purpose of interaction between the authenticated IoT devices and users. The secure block of the transaction is distributed over a public ledger because it consists of hash key values, so it cannot be changed or alternated by any other end-users. After gaining confidentiality in the system, many low-level security issues are overcome.

Redundancy and integrity are major security challenges that are recognized in any IoT device platform to avoid data redundancy to access the account of any other end-users. For achieving these integrity and nonredundancy the system is

TABLE 1: Distributed ledger of the registered end-users.

Name	Public Key	Private Key
Admin	KPUncAbY0KbyUtdom4TmYhsvQ	
Raza	CDLWGMGv0Ol6uXy0Tc92IRXJST	KPUncAbY0KbyUtdom4TmYhsvQ
Talha	RRK mzUNjtryNB4zFizm3YQ5DMq	CDLWGMGv0Ol6uXy0Tc92IRXJST
Ahsan	xTy2IpZu23Z2da1Msz3jvCxIzatBm	RRK mzUNjtryNB4zFizm3YQ5DMq
Haider	RwfDD9VrROMjoserZzAYFe8Z5JAz	xTy2IpZu23Z2da1Msz3jvCxIzatBm

TABLE 2: Distributed ledger of registered IoT devices.

Name	Public Key	Private Key
Smart lock	SkOD81SAGJVuKZXarAkrFkh5tDJ	
Mobile robot	nrNYrBZMyTB60C6Exzqlh2TsrXpV	SkOD81SAGJVuKZXarAkrFkh5tDJ
Smart light switch	VuehKkFbUOAP3xyItjhm5z8WIQN	nrNYrBZMyTB60C6Exzqlh2TsrXpV
Air quality meter	gxjJSFFkXNhE3POLkCnxp6Gsq2s	VuehKkFbUOAP3xyItjhm5z8WIQN
Voice controller	h6cvjjF4E9OVUyasddhZctQrZBd9t	gxjJSFFkXNhE3POLkCnxp6Gsq2s

TABLE 3: Distributed ledger for transactional authentication.

Name	Public Key	Private Key
Raza	CDLWGMGv0Ol6uXy0Tc92IRXJST	KPUncAbY0KbyUtdom4TmYhsvQ
Ahsan	xTy2IpZu23Z2da1Msz3jvCxIzatBm	RRK mzUNjtryNB4zFizm3YQ5DMq

TABLE 4: The output in the case of an authenticated communicational transaction is performed by the end-user mapped on to the IoT device.

Key Points	Authentic computational transaction description
Status	0 * 1 valid transactional and authentication succeed
From	0 * CDLWGMGv0Ol6uXy0Tc92IRXJSTOlPVIHrvQxJ0VX3iQ23v20YvHP65YR4yBD3Tia
To	0 * SkOD81SAGJVuKZXarAkrFkh5tDjjs9TW5y7XKkzJvhdunX8b5wkrejfr
Input	User data and device information for the transaction <pre>{ \$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'; \$charactersLength = strlen(\$characters); \$randomString = ""; For (\$i = 0; \$i < \$length; \$i++) { \$randomString .= \$characters [rand(0, \$charactersLength - 1)]; } while(\$row = mysqli_fetch_array(\$run)){ \$id = \$row["id"]; \$username = \$row["username"]; \$publickey = \$row["publickey"]; \$previoushash = \$row["previoushash"]; } }</pre>
Decoded input	
Decoded output	
Transactional cost	Algorithm running cost: O(n)

TABLE 5: Comparison between existing techniques and the proposed technique.

Existing techniques	Flaws in the existing techniques	Mitigation of flaws in the proposed technique
	Centralized architecture	De-centralized architecture
Mutual authentication	There is a need to be both client and server for authentication of each other. Both are relied on each other for authentication Less secure because relies on centralized authority	There is no need of both client and server for authentication each other More secure because does not rely on centralized authority
Open authentication	In open authentication, tokens are generated for end-users for authentication Tokens are not in an encrypted form, so everyone can access the token and breach security Totally based on open authentication server	Directly, hash key values are assigned to the end-users Hash key values are in an encrypted form, so other entities do not understand the hash key values and cannot breach security Not based on server

TABLE 5: Continued.

Existing techniques	Flaws in the existing techniques	Mitigation of flaws in the proposed technique
Kerberos authentication	Kerberos authentication uses temporal tickets for authentication purposes in a specific period	The proposed solution provides hash key values permanently to end-users
	Temporal tickets are not in an encrypted form	Hash key values are in an encrypted form
Group authentication	Dependent on temporal tickets and time, so kerberos authentication follows the centralized architecture	Provides decentralized architecture for authentication
	Group authentication authenticates entities with the permission of all other entities in the group. Message passing in a group is not in an encrypted form, so every entity in the group can easily perform some maliciousness	Hash key values are distributed across all entities in the system with the help of distributed ledgers. Hash key values are in an encrypted form, so it is hard to understand for any entity in the system.

TABLE 6: Security issues addressed by the proposed system.

Security Issues	Resolved by the proposed solution
Jamming adversaries	In the proposed solution, unique hash key values are assigned for end-users and IoT devices. In this respect, no redundancy occurs. One to one communication takes place between end-users and IoT devices. Therefore, the proposed solution mitigates the effect of jamming adversaries.
Sybil node attack	Unique hash key values are assigned to end-users. Therefore, only authenticated users that are part of the system can access the system IoT devices. There is no chance for sybil nodes to access IoT devices in the system.
Man-in-the-middle attack	Mitigate the effect of Man-in-the-Middle attack in the same way as the sybil node attack
Insecure physical interface	Virtual private network is created in terms of xamp, MySQL, which hold all records in the distributed ledger. Therefore, each end-user needs hash key values to access the system IoT devices. So any intruder or hacker cannot access the system IoT devices through the interface directly.
Double dependency problem	Proposed solution consists of decentralized architecture. De-centralized architecture is achieved through the blockchain technique. With the help of decentralized architecture, double dependency problem is removed from the system.

more protected against replay attacks and Man-in-the-Middle (MITM). Integrity and nonredundancy can also be achieved through cryptographically as every message exchanges within end-users and IoT devices in an encrypted form. Intermediate-level security issues are removed through cryptography because only an entity can breach the message which has a valid or authentic private key value.

Moreover, using a unique user value (UIDs) and time duration (duration in which users map to IoT devices) in the message authentication makes it secure against replay and Man-in-the-Middle (MITM) attacks. Even in the case of MITM, if the intruder wants to replace the user hash value address with his or her public hash key value, the intruder will not be able to sign in because he or she cannot match the correct user private hash key value.

Lastly, our proposed authentication (end-user, IoT device) scheme is resilient against higher-level attacks like Denial-of-Service (DoS). Denial-of-Service attacks are the most common and popular attacks to breach the security of the system in which an intruder or hacker is continuously attempting to access the services of the system. Concretely, multiple data fields of end-users, IoT devices, admin, and computational transactions are placed on the public ledger distributed over each node in a decentralized manner. Thus, if an intruder wants to access the system multiple times with the wrong key, the system will block the intruder credentials. The public distributed ledger is resistant and robust to DoS attacks as all public nodes are protected through hashing

functions that host redundant records with high consistency and integrity. Table 6 indicates security issues addressed by the proposed solution.

8. Conclusion

We proposed an architecture design and implementation of logical blockchain-based algorithms using the hash key values algorithm for end-users, IoT devices, and transaction authentication in a distributed appearance with no interruption of a third entity. We implement the hash key value algorithm using the PHP language and MySQL server for the storage of blocks in the distributed ledger. Authenticating large number of end-users, IoT devices, and transactions produces data deployment with MySQL server, which relieves the end-users and IoT devices from the authentication computational complexity of the blockchain network. We discussed the details of IoT device security issues and presented the system participants, architecture, interactions between system participants, and encrypted message exchanges among participants including the graphical user interface and MySQL server. Furthermore, we highlighted and showed how we executed the logic of the proposed system and examined the overall operations and functionality of the end-user and IoT device authentication mechanism governed by the hash key value algorithm. Different testing scenarios of transaction authentications were presented to verify the end-user and IoT devices of the system

using the PHP platform. Finally, we provided an analysis on the IoT security and presented that the proposed IoT device authentication solution is resilient to IoT device with different security-level attacks such as low-level, intermediate-level, and high-level attacks. In the existing proposed technique, only the users are authenticated in the blockchain, but in this paper, we present the user authentication, IoT device authentication, as well as transaction authentication. If the distributed ledger is not updated after each transaction, loss of data occur, which creates maliciousness between the blocks of a distributed ledger, because every block is connected with the previous block through the previous public hash key value to create a chain. In the future, decentralized architecture can be implemented to provide IoT device server side security.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful to the Taif University Researchers Supporting Project (number TURSP-2020/36), Taif University, Taif, Saudi Arabia. This research work was partially supported by the Faculty of Computer Science and Information Technology, University of Malaya under Postgraduate Research Grant (PG035-2016A).

References

- [1] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A user authentication scheme of IoT devices using blockchain-enabled fog nodes 2018," in *Proceedings of the 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–8, Aqaba, Jordan, October 2018.
- [2] M. Adil, M. A. Amin Almaiah, A. Omar Alsayed, and O. Almomani, "An anonymous channel categorization scheme of edge nodes to detect jamming attacks in wireless sensor networks," *Sensors*, vol. 20, no. 8, p. 2311, 2020.
- [3] M. A. Khan and K. Salah, "IoT security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.
- [4] M. Adil, M. A. Almaiah, A. O. Alsayed, and O. Almomani, "An anonymous channel categorization scheme of edge nodes to detect jamming attacks in wireless sensor networks," *Sensors*, vol. 20, pp. 1–19, 2020.
- [5] S. Dong, X.-g. Zhang, and W.-g. Zhou, "A security localization algorithm based on DV-hop against Sybil attack in wireless sensor networks," *Journal of Electrical Engineering & Technology*, vol. 15, no. 2, pp. 919–926, 2020.
- [6] M. Jamshidi, E. Zangeneh, M. Esnaashari, A. M. Darwesh, and M. R. Meybodi, "A novel model of sybil attack in cluster-based wireless sensor networks and propose a distributed algorithm to defend it," *Wireless Personal Communications*, vol. 105, no. 1, pp. 145–173, 2019.
- [7] G. Lally and D. Sgandurra, "Towards a framework for testing the security of IoT devices consistently," in *Proceedings of the First International Workshop on ETAA 2018*, Barcelona, Spain, September 2018.
- [8] T. Bhattasali and R. Chaki, "A survey of recent intrusion detection systems for wireless sensor network," in *Proceedings of the Advances in Network Security and Applications*, pp. 268–280, Chennai, India, July 2011.
- [9] R. Riaz, K.-H. Kim, and H. F. Ahmed, "Security Analysis Survey and Framework Design for Ip Connected Lowpans," in *Proceedings of the 2009 International Symposium on Autonomous Decentralized Systems (IEEE)*, pp. 1–6, Athens, Greece, March 2009.
- [10] A. Dvir, T. Holczer, and L. Buttyan, "VeRA - Version number and rank authentication in RPL," in *Proceedings of the 2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems 2011*, pp. 2709–14, Valencia, Spain, October 2011.
- [11] J. Granjal, E. Monteiro, and J. S. Silva, "Network-layer security for the internet of things using TinyOS and BLIP," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1938–1963, 2014.
- [12] S. Raza, S. Duquennoy, T. Voigt, and U. Roedig, "Demo abstract: securing communication in 6LoWPAN with compressed IPsec 2011," in *Proceedings of the 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, Barcelona, Spain, June 2011.
- [13] W. Osamy, A. M. Khedr, A. Aziz, and A. A. El-Sawy, "Cluster-tree routing based entropy scheme for data gathering in wireless sensor networks," *IEEE Access*, vol. 6, pp. 77372–77387, 2018.
- [14] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad, "Identity authentication and capability based access control (iacac) for the internet of things," *J. Cyber Secur. Mobil.* vol. 1, pp. 309–348, 2013.
- [15] N. Park, "Mutual authentication scheme in secure internet of things technology for comfortable lifestyle," *Sensors (Switzerland)*, vol. 16, pp. 1–16, 2015.
- [16] M. H. Ibrahim, "Octopus: an edge-fog mutual authentication scheme," *International Journal on Network Security*, vol. 18, pp. 1089–1101, 2016.
- [17] S. Mishra and A. Paul, "A critical analysis of attack detection schemes in IoT and open challenges," in *Proceedings of the 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 57–62, Noida, India, October 2020.
- [18] D. Conzon, T. Bolognesi, P. Brizzi, A. Lotito, R. Tomasi, and M. A. Spirito, "The Virtus Middleware: An Xmpp Based Architecture for Secure Iot Communications 2012," in *Proceedings of the 21st International Conference on Computer Communications and Networks (ICCCN)*, Munich, Germany, July 2012.
- [19] C. H. Liu, B. Yang, and T. Liu, "Efficient Naming, Addressing and Profile Services in Internet-Of-Things Sensory Environments," *Ad Hoc Networks*, vol. 18, pp. 85–101, 2014.
- [20] S. Zulkarnain and S. Idrus, "Soft Biometrics for Keystroke Dynamics," in *Proceedings of the International Conference Image Analysis and Recognition*, Niagara Falls, ON, Canada, July 2015.
- [21] L. Luu, D. H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *Proceedings of the ACM Proceedings - ACM Conference on Computer and Communications Security*, pp. 254–69, October 2016.

- [22] Y. Zhang, S. Kasahara, Y. Shen, and X. Jiang, "Smart Contract-Based Access Control for the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1–11, 2018.
- [23] A. Alonso, F. Fernández, L. Marco, and J. Salvachúa, "IAA-CaaS: IoT Application-Scoped Access Control as a Service," *Futur Internet*, vol. 9, no. 4, p. 64, 2017.
- [24] T. Borgohain, A. Borgohain, U. Kumar, and S. Sanyal, "Authentication systems in internet of things," arxiv: 1502.00870, 2015.
- [25] C. Neuman, T. Yu, S. Hartman, and K. Raeburn, "The Kerberos Network Authentication Service (V5)," MIT, Cambridge, MA, USA, 2005.
- [26] H. Shafagh, S. Raza, T. Voigt, and K. Wehrle, "Delegation-based Authentication and Authorization for the IP-Based Internet of Things," in *Proceedings of the 2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, Singapore, June 2014.
- [27] P. N. Mahalle, N. R. Prasad, and R. Prasad, "Threshold Cryptography-based Group Authentication (TCGA) Scheme for the Internet of Things (IoT)," in *Proceedings of the 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)*, pp. 1–5, Aalborg, Denmark, May 2014.
- [28] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase Authentication Protocol for Wireless Sensor Networks in Distributed IoT Applications," in *Proceedings of the 2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2728–33, Istanbul, Turkey, April 2014.