Authentication and Key Management in Distributed IoT Using Blockchain Technology

Soumyashree S. Panda, Debasish Jena, *Senior Member, IEEE*, Bhabendu Kumar Mohanta, *Member, IEEE*, Somula Ramasubbareddy, Mahmoud Daneshmand, *Senior Life Member, IEEE*, and Amir H. Gandomi, *Senior Member, IEEE*

***Abstract*—The exponential growth in the number of connected devices as well as the data produced from these devices call for a secure and efficient access control mechanism that can ensure the privacy of both users and data. Most of the conventional key management mechanisms depend upon a trusted third party like a registration center or key generation center for the generation and management of keys. Trusting a third party has its own ramifications and results in a centralized architecture; therefore, this article addresses these issues by designing a Blockchain-based distributed IoT architecture that uses hash chains for secure key management. The proposed architecture exploits the key characteristics of the Blockchain technology, such as openness, immutability, traceability, and fault tolerance, to ensure data privacy in IoT scenarios and, thus, provides a secure environment for communication. This article also proposes a scheme for secure and efficient key generation and management for mutual authentication between communication entities. The proposed scheme uses a one-way hash chain technique to provide a set of public and private key pairs to the IoT devices that allow the key pairs to verify themselves at any time. Experimental analysis confirms the superior performance of the proposed scheme to the conventional mechanisms.**

***Index Terms*—Blockchain, decentralization, hash chain, Internet of Things, privacy, security.**

I. Introduction

THE RATE at which the number of physical devices connected to the Internet is increasing exponentially. People are gradually furnishing their homes with smart devices, such as smart remote controls, smart TVs, surveillance cameras, smart bulbs, etc., while vehicles are being equipped with different smart devices so that they can share traffic-related data [1]. In factories, robots and smart tools are being implemented to increase the productivity of their operations. The application areas of IoT are not limited to these use cases; but is largely endorsed in several other areas, including agriculture, cities, transportation system, grids, etc. Indeed, IoT has allowed the evolution of many other areas, such as smart health systems, smart transportation systems, smart agriculture, and so on. With such expansion to a wide range of fields, the number of devices connected to the Internet and to each other is expected to reach around 20 billion by 2022 [1].

By concept, an IoT application system is ubiquitous of a variety of devices (things) that are capable of interacting with each other so that a broad range of services can be provided. Each device, be it physical or virtual, of an IoT system must be accessible by the system users regardless of their location. It is critical that only authenticated and approved users can access the system; otherwise, the system will be vulnerable to numerous security attacks, such as spoofing, data tampering, Denial-of-Service (DOS) attack, impersonation attack, information theft, etc. Certainly, these security issues continue to be the prime obstacle for the adoption of IoT in large-scale organizations. As per a survey, one of the most significant concerns in the deployment of solutions for different IoT use cases is security. Securing the communication among different entities and ensuring data privacy using encryption are the most commonly used methods to ensure IoT security [2]. However, the conventional security methods do not fully conform to the IoT systems because of the heterogeneity and limited resources of IoT devices. Moreover, most of the proposed solutions are centralized in which scalability becomes a matter of concern since thousands of devices work in an IoT use case [3]. Finally, each use case demands a different approach for system design, deployment, and ensuring security. Therefore, new approaches should be designed with the aim to facilitate the hassle-free addition of new services as well as new devices with add-on security benefits.

As a recently promising solution, the concept of Blockchain is suggested to provide a secure and efficient base for several IoT applications. With the growing popularity of digital currency, researchers have focused their attention on the different usages of Blockchain which is the key element behind Bitcoin. Coined by Nakamoto in 2008 [4], a Blockchain is essentially a distributed ledger that is inherently immutable, open, synchronized, and verifiable [5]. It facilitates distributed decision making so that all entities of the system share equal

Manuscript received September 10, 2020; revised December 10, 2020 and February 2, 2021; accepted February 28, 2021. Date of publication March 4, 2021; date of current version August 6, 2021. *(Corresponding author: Amir H. Gandomi.)*

Soumyashree S. Panda, Debasish Jena, and Bhabendu Kumar Mohanta are with the Department of Computer Science Engineering, International Institute of Information Technology Bhubaneswar, Bhubaneswar 751003, India (e-mail: c117011@iiit-bh.ac.in; debasish@iiit-bh.ac.in; c116004@iiit-bh.ac.in).

Somula Ramasubbareddy is with the Department of Information Technology, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad 500090, India (e-mail: svramasubbareddy1219@gmail.com).

Mahmoud Daneshmand is with the School of Business, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: mahmoud.daneshmand@stevens.edu).

Amir H. Gandomi is with the Faculty of Engineering Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: gandomi@uts.edu.au).

Digital Object Identifier 10.1109/JIOT.2021.3063806

privilege. Simply put, Blockchain networks enable a number of entities that do not share a trust relationship to coordinate, amalgamate, and associate in application development process or business intelligence process [6].

Most of the existing works solely depend upon the security attribute of Blockchain, which may not be enough for some of the IoT use cases. For example, the full anonymity provided by Blockchain does not ensure identification, which is crucial in most of the IoT use cases. Moreover, it remains unclear whether a low-power and resource-constrained IoT device will be able to perform transactions in Blockchain and participate in the Blockchain mining process. Most of the researchers are still in their elementary stage, whereby only an approach is presented but no proper implementation or analysis is given. Therefore, in this article, a distributed framework using two Blockchain structures is presented which enables secure communication among IoT devices. One-way hash chains are employed for authentication and key management.

Given the challenges in developing a distributed, reliable, and secure authentication scheme for a heterogeneous IoT network, a Blockchain-based distributed authentication, and key management scheme has been proposed in this article. The key contributions are given as follows.

1) A framework using two Blockchain structures is developed to provide a distributed and secure IoT network for communication.
2) A distributed authentication and key management using one-way hash chains to authenticate as well as to assign keys to the entities of the system. The introduction of Blockchain technology in the scheme facilitates distributed decision making without the need for a third party.
3) The scheme has been implemented on the Ethereum platform and an in-depth evaluation of the scheme proves its proficiency in making an IoT use case secure.
4) Further security analysis of the proposed authentication scheme is being compared with other existing schemes which prove the strength of the proposed scheme. The performance analysis shows that the scheme is highly efficient and scalable.

The remainder of this article is arranged as follows. Section II gives a brief introduction of Blockchain along with some popular platforms that combine it with IoT. Section III discusses the existing Blockchain-based security solutions for IoT systems. Section IV discusses the preliminaries required for the proposed scheme. Then, a detailed description of the proposed model is presented in Section V. The scheme is evaluated in terms of security and performance in Section VI. Finally, Section VII concludes this article with future research plans.

II. Blockchain for IoT

Blockchain, the key element of Bitcoin, has been growing at an unbelievable pace over the last few years with its application now extending beyond digital currency. As stated, Blockchain, as a distributed ledger that is inherently immutable, open, synchronized, and verifiable, can be thought of as a shared replicated ledger with smart contracts [6]. Smart contracts are nothing more than a piece of computer codes that provide the shared implementation of the business rules associated with each transaction. The following features of Blockchain describe how and why Blockchain can be used to handle the different issues related to data privacy and security in an IoT framework.

1) *Consensus:* The entities of the network will collectively agree that each transaction that is recorded in the Blockchain and the order of transactions in relation to others are valid.
2) *Provenance:* The entities know the history of the data and how it flows within the network.
3) *Immutability:* Entities cannot tamper with the transactions once they are agreed upon and recorded in the chain.
4) *Finality:* Once a transaction is committed, it cannot be reversed, i.e., data cannot be rolled back to the previous state. If a transaction is in error, then a new transaction must be used to reverse the error with both transactions visible.

Bitcoin is an example of a permission-less public Blockchain. It is a peer-to-peer payment system that allows people to send currency to one another without requiring a centralized intermediary using a class of assets called cryptocurrency [5]. It uses a resource-intensive process known as Proof of Work (PoW) to achieve consensus. PoW in the Bitcoin system extends the hashcash-based PoW system and develops a mechanism to safeguard the Blockchain by applying the distributed consensus mechanism [4]. The hashcash system was proposed by Adam Back and uses the puzzle friendliness property of the cryptographic hash function [7]. Transactions in Bitcoin are public and visible, but the entities behind each transaction are largely anonymous making them very difficult to track.

To exploit the advantages of Blockchain technology, a number of platforms have been designed to integrate it with IoT to provide smart and usable foundations for future research and development. Some of the popular platforms include Ethereum, Hyperledger, Multichain, IOTA, Rootstock, IoT Chain, Atonomi, Lisk, Chain of Things, etc. Specifically, Ethereum was the first acknowledged platform for the development of decentralized or distributed systems using Blockchain technology, which supports smart contracts. These smart contracts executed on the Ethereum virtual machine (EVM), a type of operating system provided by the Ethereum platform [8]. Ethereum provides a type of cryptocurrency called Ether (ETH), that can be used for both financial transactions and executing smart contracts. Though most of the earlier versions of Ethereum used PoW as the consensus mechanism, the recent version employs Proof of Stake (PoS) as the consensus mechanism. The PoW-based consensus used in Ethereum is known as Ethash, a memory intensive and less power consuming consensus mechanism as compared to traditional PoW. Ethereum can be used to implement both permission less and permission-based frameworks over Blockchain. Lately, smart contracts have been extensively used for modeling and securing a number of IoT use cases. Ethereum was the first platform to provide a base for the development of distributed applications (DAPPs) [9]. As another popular platform, Hyperledger is a permission-based Blockchain framework that provides an Enterprise-grade foundation for transactional applications, where the nodes in the network need to know each other prior to setting up the network [10]. Practical Byzantine fault tolerance (PBFT) is used as the consensus mechanism used in Hyperledger fabric, which safeguards the network from crash faults, network faults, Sybil attacks, and Byzantine nodes. Hyperledger provides better performance in terms of higher transaction throughput and less power consumption compared to Bitcoin and Ethereum. Nevertheless, it has limitations, for instance, applications built on Hyperledger cannot be fully decentralized and will be less scalable. Multichain is another open platform to model and deploy private Blockchain within a closed environment, just like Hyperledger. It is forked from Bitcoin to broaden the functionality domain of Blockchain that provides users with more features, such as speed, permissions, multiple assets, and atomic exchanges. Another Ethereum like platform is Rootstock for Blockchain-based IoT developments. Since it is compatible with Ethereum, smart contracts written for the Ethereum environment can also be used over this platform. It also has a built-in infrastructure layer that provides users with better computing power, fast payment channels, and larger storage space. Atonomi is another platform that provides trust and identity that are essential for the increasingly connected world by securing a device's identity on the distributed ledger, tracking a device's reputation, and securing the communication between devices.

III. Related Work

Even though Blockchain is still in its infancy, substantial research has already been done in different areas of IoT using Blockchain technology. In this section, the authors discuss some of the existing works in the field.

Christidis and Devetsikiotis [11] proposed the advantages and disadvantages of Blockchain technology with respect to IoT, concluding that Blockchain promotes the secure and trustworthy sharing of resources and data in an IoT environment among multiple entities. Bahga and Madisetti [12] revealed shown how Blockchain technology can be used to design a trust-less, decentralized environment for industrial IoT. However, there was no formal proof for validation of their proposed model given in this article. In [13], a privacy-preserving mechanism was presented that helps to authorize IoT devices in cloud systems. The presented method allows stakeholders to share their data gathered from sensor devices with different service providers in a fully anonymous way. Yet, it was not adapted to the use cases where identification is essential. Another access control method using Blockchain technology known as "FairAccess" was proposed in [14], which works analogous to the role-based access control [15]. FairAccess was specially designed for IoT use cases where the policies are kept in a private Blockchain so that they cannot be tampered with. However, this method is not applicable to all IoT use cases since it was designed to work only for policy-based systems.

Dorri *et al.* [16] addressed the various challenges of Blockchain in the context of IoT, such as scalability, computational complexity, and storage overhead and propose a lightweight Blockchain having a simple consensus mechanism to address these issues. Huh *et al.* [17] shared an approach to combine IoT and Blockchain technology where smart contracts define the functionalities of each device. However, their approach lacks clarity in terms of the usage and the application of the approach to different use cases of IoT is also restricted. Pham *et al.* [18] utilized Ethereum Blockchain for secure analysis and management of medical sensors.

These sensors combined with IoT smart devices help in monitoring the health condition of a patient from remote locations. Another method to ensure mutual authentication among IoT devices is introduced in [3], which groups IoT devices into virtual zones within which they can share data securely. However, this method does not allow interzonal communication and is still in its elementary phase. In [19], a distributed storage system is presented for IoT applications that generate huge amounts of data. Although the work confirms that the storage system utilizes Blockchain technology to store the generated data in a distributed manner, other security and privacy needs of IoT applications are not addressed in the research article.

Recently, an authentication scheme for IoT devices using gateway nodes and Blockchain technology has been proposed in [20], where gateway nodes are included to address the low computation power and resource-constrained nature of IoT devices. Similarly, in [21], IoT devices are connected to fog nodes that share a Blockchain structure. Even though the proposed design ensures a secure communication between fog nodes and devices, the applicability of the scheme is very restricted.

To summarize, most of the existing research works are not applicable to the wide range of IoT application areas. Apart from this, most of the works solely depend upon the security attribute of Blockchain, which may not be enough for some of the IoT use cases. For example, the full anonymity provided by Blockchain does not ensure identification, which is crucial in most of the IoT use cases. Moreover, it remains unclear whether a low-power and resource-constrained IoT device will be able to perform transactions in Blockchain and participate in the Blockchain mining process. Most of the researchers are still in their elementary stage, whereby only an approach is presented but no proper implementation or analysis is given.

IV. Preliminaries

A. System Variables

This section specifies the system variables that need to be accepted and used by all entities of the system. These variables are specified as follows.

1) Assume G to be a cyclic multiplicative subgroup of Z_p^* of prime order p, with identity element $e = 1$ and $g \in G$ is a generator of G. We assume that computing discrete logarithms in G with respect to g is computationally infeasible.

TABLE I
NOTATION TABLE

H	One-way hash function mapping the set ¡0,1,• • • ,p-1¿ onto itself
h	Cryptographic hash function
D_id	Unique identity given to the Device
AMN_id	Unique identity of the Access Managing Node
pk_D	Permanent Public Key of Device
prk_D	Permanent Private Key of Device
pk_AMN	Permanent Public Key of AMN
prk_AMN	Permanent Private Key of AMN
puk_k	Public key from the generated key set
prk_k	Private key from the generated key set
N	Number of key pairs generated per device
E_{key}	Encryption using key
D_{key}	Decryption using key

For example, G might be a large multiplicative subgroup of Z_p^* for some large prime p, where q is a large prime dividing $p-1$. Alternatively, G could be the group of points on an elliptic curve.

2) N is a positive integer that specifies the number of public/private key pairs available to an IoT device.

Table I contains the description of the notations used in this article.

B. One-Way Hash Chain

One-way hash chains are a kind of cryptographic hash used in many applications, such as micropayment systems [22], mobile *ad hoc* networks [23], etc., for providing a set of cryptographic keys from a single key. This technique was introduced by Lamport for securing passwords from intruders and malicious users [24]. As per the technique, given a number known as a seed and a cryptographic hash function such as SHA-1, the successive application of the hash function to the seed generates a set of hash values known as a hash chain. The characteristic of the hash chain is that it is computationally impossible to invert [25].

It works as follows. Initially, an entity will have to choose a secret number known as seed s and a number N, where $s \in (0, 1, \dots, p-1)$.

Then, it will repeatedly apply the one-way hash function defined above for $N-1$ times to produce a set of $N-1$ values denoted as $H_1, H_2, H_3 \cdots H_N$, where $H_1 = H(H_2)$, $H_{k-1} = H(H_k)$, and $H_N = H_{N-1}$, where $1 < k \leq N.H_1$ is named as the tip of the hash chain.

These values can be used as keys in the reverse order of creation, i.e., in the order $H_{N-1}, \dots, H_k, \dots, H_2, H_1$will be consumed by the entity. Thus, any hash value needs to be kept secret until it is used, and the validity of a particular hash value can be checked easily with a simple hash operation after receiving it. It is important to note that the disclosure of any key, say H_k, does not reveal any information about other keys. With that being said, if a one-way hash chain is used to uniquely bind a set of public key/private key pairs, a public key belonging to the chain can be validated using the hash function for the required number of times to the received public key [25]. For the proposed scheme, a system entity, say A, first chooses an integer $s \in 0, 1, \dots, p-1$, then uses a one-way hash function H on the value s for N times as shown in

$$H^N(s) \leftarrow H^{N-1}(s) \leftarrow H^{N-2}(s) \cdots \leftarrow H^k(s) \cdots \leftarrow H^2(s) \leftarrow H^1(s) \leftarrow s. \tag{1}$$

Then, A computes a hash value ϑ that will be used by other entities of the system to validate A as follows:

$$\vartheta = \underbrace{h\Big(g^{\prod_{j=0}^{N} H^j(s)}\Big)}_{}. \tag{2}$$

The kth private key (PrK_k) and its corresponding public key (PuK_k), where $0 < k < N$, are generated by the following equations:

$$PrK_k = \underbrace{\prod_{j=0}^{k} H^j(s)}_{} \tag{3}$$

$$PuK_k = \underbrace{g^{\prod_{j=0}^{k} H^j(s)}}_{}. \tag{4}$$

V. PROPOSED WORK

This section presents the proposed system architecture and scheme for IoT use cases implementing Blockchain technology in detail. The designed solution for the system employs a one-way hash chain for authentication and key management. The most important aspect is that this method significantly decreases the computational overhead and communication latency, which can drastically improve the efficiency, reliability, and scalability of the system.

A. System Architecture

The proposed system design of the Blockchain-based distributed architecture for IoT use cases is shown in Fig. 1. The architecture consists of three layers, namely, device, fog, and cloud layers. The device layer consists of the smart devices used in various IoT use cases, for example, different wearable medical devices to sense, monitor, and observe patient's health status from a remote location. These include temperature sensors, gas sensors, and surveillance cameras for home or organization automation. Since the devices are resource constrained by nature, the fog layer was added to improve the performance and reduce the computation time and overhead of the devices. The fog layer contains a number of access managing nodes (AMNs) with standard computational and storage capabilities to manage the devices of the device layer. Devices belonging to similar use cases are grouped together into domains, where each domain is managed by an AMN. Similarly, a set of AMNs are grouped together to form a network in the fog layer and are responsible for generating, distributing, and managing the secret keys for the devices linked to them. AMNs belonging to a network share a Blockchain structure to store the transactions related to authentication and key management of the same network. AMNs also act as miners to pack the transactions of the devices that occurred within a certain time interval into a new block.

Next, the fog layer is connected to the cloud layer via high-speed network connectivity. The cloud layer manages

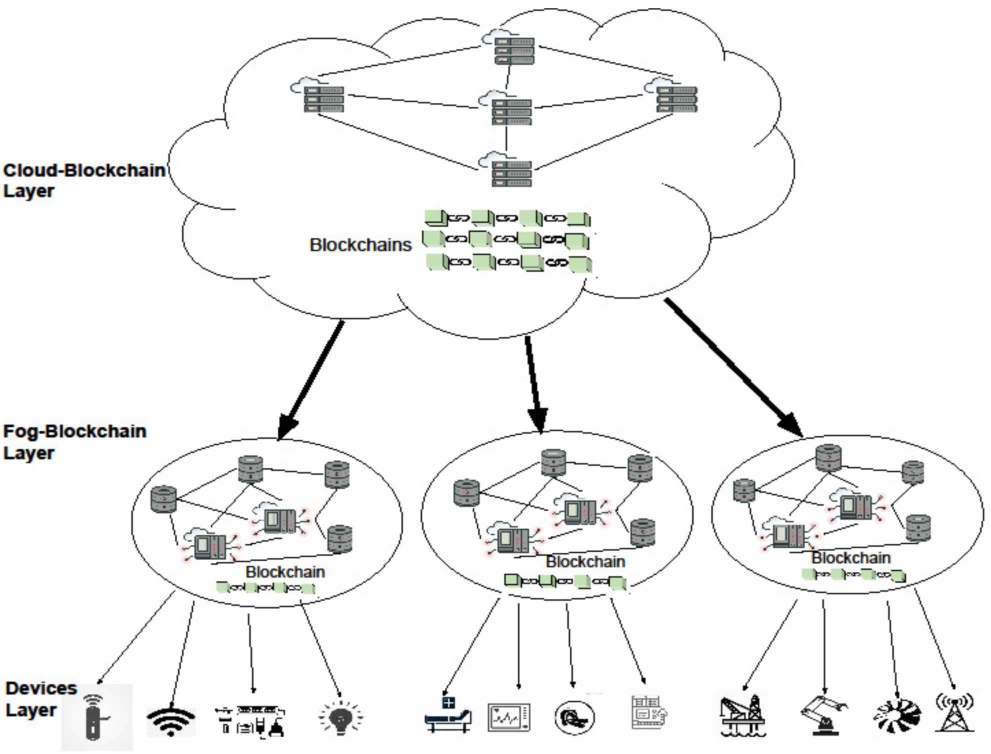


Fig. 1. Blockchain-based IoT architecture.

multiple Blockchains; each from the AMN network of the fog layer. For this, a number of nodes, known as manager nodes (MNs) possessing immense computing capabilities are introduced in the cloud layer to handle the constrained resources and highly scalable IoT use cases. Communications within the same network are handled by the AMNs of the network while internetwork transactions are handled by the MNs of the cloud layer. Moreover, the MNs also store the data generated by the devices of the lower layer in an encrypted manner, and the data can be accessed after proper authentication.

The proposed scheme assumes that all entities constituting the architecture are furnished with a highly correct atomic clock, whereby the clocks of the AMNs and devices belonging to the same network are synchronized.

B. Authentication and Key Management Scheme

This section gives a detailed description of the proposed scheme, which functions in three phases as shown in Fig. 2. The detailed procedure of each phase is described as follows.

1) System Initialization and Device Registration Phase: The MNs at the top layer are responsible for selecting the system variables as defined in Section III-A and will announce these values to the AMNs at the fog layer. As already stated, AMNs function as network managers to generate and manage the keys of the devices connected to them. Each device at the device layer generates a public/private key pair.

An AMN registers an unregistered device, by providing a structure known as a "license" that will be used as a permit to take part in the network. The license includes: 1) the unique device identity (*D_id*); 2) the unique identity of the AMN to which the device will belong (*AMN_id*); 3) the permanent public key of the device (*pk_D*); and 4) a signature using the private key (*prk_AMN*) of the AMN. Then, the AMN issues transactions in the Blockchain regarding the registration of the device. Subsequently, the smart contracts written for the registration of devices in the Blockchain check for the uniqueness of the device's identity. If the transaction is correct, then the registration details (license) of the device are stored in the Blockchain, which can only be accessed by the AMNs connected to that particular network

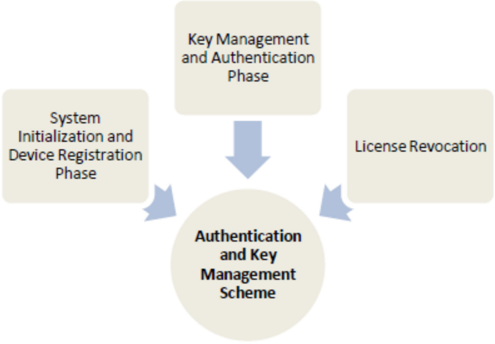

Key Management and Authentication Phase
System Initialization and Device Registration Phase
License Revocation
Authentication and Key Management Scheme

Fig. 2. Authentication and key management phases.

$$AMN \xrightarrow{\text{license}=\left(D\_\text{id}||AMN\_\text{id}\|pk\_D\|\text{Sign}_{prk\_AMN}\right)} D.$$

2) Key Management and Authentication Phase: Whenever a registered device, say D_p, wants to communicate with another device, say D_q. belonging to the same domain, it requires encryption keys for the secure sharing of data. For that, D_p generates a seed s where $s \in (0, 1, \ldots, p-1)$. Then, it encrypts the s using the public key of the AMN to which it belongs, say AMN_x (pk_AMN_x), and sends it to an AMN. It also sends the license with the above message for verifying itself to the AMN_x. Then, the AMN_x generates the N number of public key/private key pairs and the hash values ϑ using the one-way hash chain described in Section III-B. The AMN_x issues a transaction to the Blockchain to store $H^N(s)$, ϑ, the current timestamp value, and the duration for which the key set will be valid, corresponding to D_p. Once the transaction is verified by all AMNs of the network, AMN_x encrypts the generated key set using a public key of D_p, signs the message, then sends to D_p.

When D_p wants to connect to D_q to access data or share information, first it has to prove its authenticity to D_q, then they will establish a session key for further communication. For this, D_p initiates the communication during time interval $t_k, 0 \leq k < N$ by sending a message directly to D_q that includes: 1) $PuK_k = g^{\prod_{j=0}^{k} H^j(s)}$; 2) the license; 3) current time (T_p); and 4) cipher text of a random number $\Re_p$, $(0 < \Re_p < p-1)$ using public key of D_q $[E_{pk\_D_q}(\Re_p)]$

$$D_p \overset{PuK_k,\text{license},T_p,E_{pk\_D_q}(\Re_p)}{\rightarrow} D_q.$$

On receiving the above message, D_q first verifies whether $T_q - T_p < \Delta\Gamma$, where T_q is the current system time at D_q and $\Delta\Gamma$ is the maximum tolerable time interval. If it holds, it then verifies the correctness of the license through its AMN. If found correct, then it computes PrK_i, $(k+1) \leq i \leq N$ (since D_q is supposed to know k because of time synchronization)

$$\begin{aligned}\vartheta^* &= (PuK_k)^{\prod_{j=0}^{k} H^j(s)} \\ &= g^{\prod_{j=0}^{k} H^j(s) \prod_{j=0}^{k} H^j(s)} \\ &= g^{\prod_{j=0}^{N} H^j(s)}.\end{aligned}$$

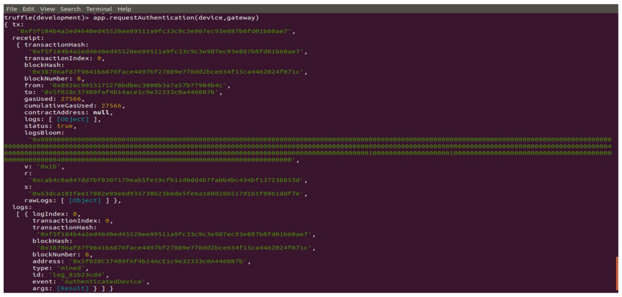
To verify the authenticity of D_p, D_q checks whether $h(\vartheta^*) = \vartheta$. (The value of ϑ can be obtained from the corresponding AMN for D_q.) If it is not true, then D_q rejects the request and reports to its AMN. If it matches, then D_q successfully verifies D_p as a valid entity of the system. Next, it decrypts $D_{prk\_D_q}[E_{pk\_D_q}(\Re_p)] = \Re_p^*$ using its private key and selects a random number $\Re_q$, $(0 < \Re_q < p-1)$. Then, it forms the reply message, which includes: 1) license; 2) $PuK_l = g^{\prod_{j=0}^{l} H^j(s)}$ (valid public key of D_q); 3) $[E_{pk\_D_p}(\Re_q)]$; and 4) $h(\Re_q || \Re_p^*)$

$$D_p \overset{PuK_l,\text{license},E_{pk\_D_p}(\Re_q)}{\leftarrow} D_q.$$

When D_p gets this message, it follows the same process as D_p to verify D_q. If verified then it computes $D_{prk\_D_p}[E_{pk\_D_p}(\Re_q)] = \Re_p^*$ and verifies whether $h(\Re_q || \Re_p^*) = h(\Re_q^* || \Re_p)$. If it holds then it verifies D_q and sends acknowledgment.

Finally, both D_p and D_q compute the session key as $h(PuK_k || \Re_q || \Re_p || PuK_l)$.

3) License Revocation: If a device is found to be malicious, then the AMN issues a new transaction to revoke the license of that particular device. The transaction stores the identity of the device into a new block so that any further messages from this particular device will be rejected.



```
truffle(development)> app.requestAuthentication(device,gateway)
{ tx:
   '0xf5f184b4a2ed4640ed45528ee89511a9fc33c9c3e987ec93e887b0fd01b60ae7',
  receipt:
   { transactionHash:
      '0xf5f184b4a2ed4640ed45528ee89511a9fc33c9c3e987ec93e887b0fd01b60ae7',
     transactionIndex: 0,
     blockHash:
      '0x38706af67f9641b6876face4497bf27889e778dd2bce634f15ca4462024f071c',
     blockNumber: 8,
     from: '0x0926c9953175278bdbec3800b3a7a57b77904b4c',
     to: '0x5f028c37409faf4b14ace1c9e32333c0a446087b',
     gasUsed: 27566,
     cumulativeGasUsed: 27566,
     contractAddress: null,
     logs: [ [Object] ],
     status: true,
     logsBloom:
      '0x000000000000000000000040000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000800000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000004
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000100000000000000000000001000000000000000000000000000000000000000
0000000000004000000000000000000000000000000000000000000000000000000000000',
     v: '0x1b',
     r:
      '0xcab4c0a847dd7bf8307179eab5fe19cfb11d0dd4b7fabb4bc434bf13723b853d',
     s:
      '0x63dca181fee17882e89e6d935730623b6de5fe6a180d26b517d161f0961ddf7e',
     rawLogs: [ [Object] ] },
  logs:
   [ { logIndex: 0,
       transactionIndex: 0,
       transactionHash:
        '0xf5f184b4a2ed4640ed45528ee89511a9fc33c9c3e987ec93e887b0fd01b60ae7',
       blockHash:
        '0x38706af67f9641b6876face4497bf27889e778dd2bce634f15ca4462024f071c',
       blockNumber: 8,
       address: '0x5f028C37409fAf4b14AcE1c9e32333c0A446087b',
       type: 'mined',
       id: 'log_81b23cd4',
       event: 'AuthenticatedDevice',
       args: [Result] } ] }
```

Fig. 3. Device authentication in Blockchain.

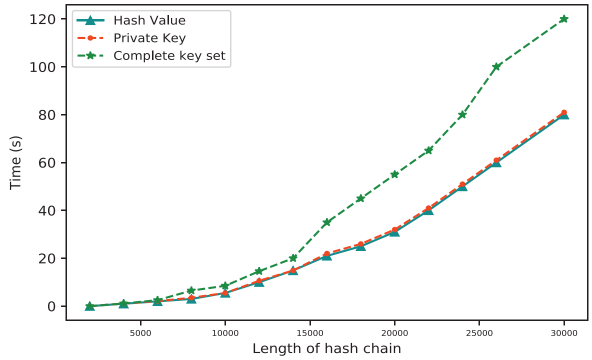


Fig. 4. Time required for key pair generation.

VI. Evaluation

A. Performance Analysis

In this section, a detailed description is provided of how the proposed scheme is implemented using smart contracts. The experimental setup consists of two cloud servers to simulate the MNs, four laptops to function as AMNs, and two Raspberry Pi to connect the devices. The Ethereum platform is used to realize the Blockchain network. Smart contracts that serve as the core of the proposed system are developed using Solidity language [8]. These smart contracts were implemented and verified using Remix IDE before deploying them in the Blockchain platform [26]. In fact, Ropsten Testnet was used as an Ethereum tool for testing and development purposes. The output of the authentication process is shown in Fig. 3.

As the proposed approach uses one-way hash chains for validation and key management, its performance was evaluated on a system with specification Intel Core i5, CPU-3.30 GHz, 8 GB of RAM, Win 8, 64-bit OS. Fig. 4 shows that the time required to generate the private/public key pairs is analogous to the generation of corresponding hash values. This is because both key pairs and hash values require multiplication operations for their computation but the number of multiplication operations required for private key generation is inversely proportional to those for hash value generation. In addition, the time required to generate the entire key pairs and hash values is 120 s for about 29 780 key pairs.

Fig. 5 displays the plot of transaction time with respect to the rate of issuing a transaction and the number of devices. At

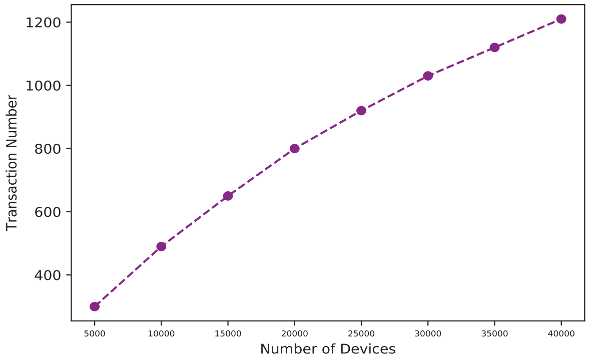


Fig. 5. Transaction number with respect to the number of devices.

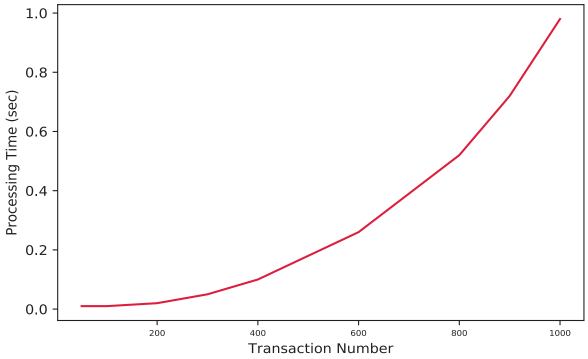


Fig. 6. Block preparation time.

TABLE II
COMPARISON BASED ON CHARACTERISTIC PARAMETERS

Characteristic	**[20]**	**[22]**	**[24]**	**Current Study**
Mutual Authentication	Yes	Yes	–	**Yes**
Resists Replay Attack	No	No	–	**Yes**
Resists MITM Attack	Yes	Yes	–	**Yes**
Resists DOS Attack	Yes	Yes	–	**Yes**
Scalablility	No	No	No	**Yes**
Implementation	No	No	Yes	**Yes**

an average rate (0.03), the transaction number increased from 112 for 5000 devices to 1200 for 40 350 devices, which proves the scalability of the system. The time required to prepare a block is shown in Fig. 6, revealing that preparation time slowly increased up to 400 transactions. Preparation time over 0.4 s when the number of transactions was larger than 700. Finally, the preparation time reached 1 s when there were 990 transactions.

B. Security Analysis

An extensive analysis of the proposed scheme proves that the scheme is highly accomplished to meet the privacy and security needs of an IoT use case. This section shows how the proposed scheme secured against various network attacks and makes the comparison to some existing works that have similar objectives. The comparison detail is provided in Table II.

1) *Mutual Authentication and Message Integrity:* Authenticating the source as well as the destination before starting a communication is crucial in an IoT system to avoid impersonation and Man-in-the-Middle (MITM) attacks. Each device holds a license that is digitally signed by a trusted authority. During communication, each device uses its license to verify itself to the other device, whereby only a valid device can correctly compute ϑ. Finally, both the communicating devices use unique random numbers to verify each other.
2) *Resistance to Replay Attack:* An adversary can use already sent messages to gain knowledge about the confidential information of the entities. The use of timestamp values with each request message ensures that the proposed approach is secure against replay attacks.
3) *Resistance to Sybil Attack:* In a Sybil attack, the attacker disturbs a system by creating multiple identities. These fake identities share wrong information and hence affect the decision making of the system. To address this issue, each device of the proposed model can have only a single pair of keys at a particular time, which is mentioned in the license. Besides, each device has been assigned a unique device identity in the registration phase that is stored in the Blockchain. Thus, a malicious node will not be able to fake identities to disturb the system.
4) *Resistance to Man-in-the-Middle Attack:* In the proposed work, random numbers and public/private key pairs generated from the hash chain are used to successfully resist the system from this attack. Suppose an attacker (D_a) starts a session parallel to a valid session by sending the same message as $D_p < PuK_k, \text{license}, T_p, E_{pk\_D_q}(\Re_p) >= < PuK_k, \text{license}, T_p, E_{pk\_D_q}(\Re_p) >$. When D_q receives this message, it follows the procedure as described in Section V-B2 and replies with messages that includes $[E_{pk\_D_p}(\Re_q), h(\Re_q||\Re_p)]$ and $[E_{pk\_D_a}(\Re_a), h(\Re_a||\Re_p)]$ to D_p and D_a, respectively. At this point, D_a blocks the message meant for D_p and $[E_{pk\_D_a}(\Re_a), h(\Re_a||\Re_p)]$ to D_p. Then, D_p decrypts $\Re_a$ and sends it to D_p for final verification. But it fails and thus it proves that the proposed approach resists the MITM attack.
5) *Resistance to Denial-of-Service (DoS) Attack:* In a DOS attack, the adversary attempts to prevent the use of a network resource or a valid service by temporarily or permanently blocking the server of the system. In a Distributed DOS (DDoS) attack, multiple attackers consume the resources of the system to disrupt its normal functioning. This can be done by flooding the target device with unnecessary messages. If the target device is the central node of a centralized system, then the failure of the central node affects the whole system. In the proposed approach, both the use of Blockchain technology and the large number of miners in the Ethereum platform increases the resistance to such an attack. Furthermore, the high cost of making a transaction in a Blockchain network, discourages an attacker from launching an attack.
6) *Scalability:* In the context of this article, scalability is characterized by the ability to guarantee that the size of the system does not affect its performance. In other words, if the number of devices increases, then it should not affect the time required for authentication and key

management. In the proposed work, the AMNs store the information related to authentication and key management of their own network. All Blockchains belonging to different AMN networks are handled by MNs of the cloud layer. Apart from this, a device has to store very minimal information required only for validating its authenticity and securing its communication with other devices. Moreover, using peer-to-peer networks like Blockchain, the scalability issue can be handled very easily [27]. Due to all these features, the proposed approach can achieve a good security performance.

VII. Conclusion

In this article, a novel approach for distributed authentication and key management is presented. The approach exploits the advantages of Blockchain technology, cloud computing, and fog computing to achieve a secure and efficient architecture for IoT use cases. The entire system is divided into layers of Blockchain to speed up the validation process and to increase the scalability of the system, whereby the Ethreum platform was used to develop the Blockchain network. The scheme was thoroughly evaluated, confirming the high efficiency and scalability of the scheme. The security analysis further demonstrates the scheme's compliance to the security requirements of IoT use cases. Future works to improve the proposed approach are suggested to: 1) design schemes for internetwork communication among the AMNs as well as devices and 2) implement and evaluate the schemes to verify their ability in providing security and performance requirements.

References

[1] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019.

[2] M. Ma, G. Shi, and F. Li, "Privacy-oriented blockchain-based distributed key management architecture for hierarchical access control in the IoT scenario," *IEEE Access*, vol. 7, pp. 34045–34059, 2019.

[3] M. T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni, "Bubbles of trust: A decentralized blockchain-based authentication system for IoT," *Comput. Secur.*, vol. 78, pp. 126–142, Sep. 2018.

[4] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[5] S. S. Panda, B. K. Mohanta, U. Satapathy, D. Jena, D. Gountia, and T. K. Patra, "Study of blockchain based decentralized consensus algorithms," in *Proc. TENCON IEEE Region 10 Conf. (TENCON)*, 2019, pp. 908–913.

[6] B. K. Mohanta, S. S. Panda, and D. Jena, "An overview of smart contract and use cases in blockchain technology," in *Proc. 9th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2018, pp. 1–4.

[7] A. Back, *Hashcash-a denial of service counter-measure*, 2002. [Online]. Available: http://www.hashcash.org/ papers/hashcash.pdf

[8] C. Dannen, *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*. New York, NY, USA: Apress, 2017.

[9] B. K. Mohanta, A. Sahoo, S. Patel, S. S. Panda, D. Jena, and D. Gountia, "Decauth: Decentralized authentication scheme for iot device using ethereum blockchain," in *Proc. TENCON IEEE Region 10 Conf. (TENCON)*, 2019, pp. 558–563.

[10] E. Androulaki *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, 2018, pp. 1–15.

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

[12] A. Bahga and V. K. Madisetti, "Blockchain platform for industrial Internet of Things," *J. Softw. Eng. Appl.*, vol. 9, no. 10, pp. 533–546, 2016.

[13] T. Hardjono and N. Smith, "Cloud-based commissioning of constrained devices using permissioned blockchains," in *Proc. 2nd ACM Int. Workshop IoT Privacy Trust Secur.*, 2016, pp. 29–36.

[14] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman, "FairAccess: a new blockchain-based access control framework for the Internet of Things," *Secur. Commun. Netw.*, vol. 9, no. 18, pp. 5943–5964, 2016.

[15] D. Ferraiolo, J. Cugini, and D. R. Kuhn, "Role-based access control (RBAC): Features and motivations," in *Proc. 11th Annu. Comput. Secur. Appl. Conf.*, 1995, pp. 241–248.

[16] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A lightweight scalable blockchain for IoT security and privacy," 2017. [Online]. Available: arXiv:1712.02969.

[17] 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.

[18] H. L. Pham, T. H. Tran, and Y. Nakashima, "A secure remote healthcare system for hospital using blockchain smart contract," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.

[19] Q. Xu, K. M. M. Aung, Y. Zhu, and K. L. Yong, "A blockchain-based storage system for data analytics in the Internet of Things," in *New Advances in the Internet of Things*. Cham, Switzerland: Springer, 2018, pp. 119–138.

[20] S. S. Panda, U. Satapathy, B. K. Mohanta, D. Jena, and D. Gountia, "A blockchain based decentralized authentication framework for resource constrained iot devices," in *Proc. 10th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2019, pp. 1–6.

[21] B. K. Mohanta, D. Jena, S. S. Panda, and D. Gountia, "Decentralized secure fog computing in cloud-fog-iot infrastructure using blockchain," in *Int. Conf. Emerging Technologies Inf. Commun. (ETIC)*, Bhutan, 2019, pp. 24–29.

[22] R. L. Rivest and A. Shamir, "PayWord and MicroMint: Two simple micropayment schemes," in *Security Protocols*, Lomas M. (eds), Lecture Notes in Computer Science, vol. 1189, Heidelberg, Germany: Springer, 1996, pp. 69–87, doi: 10.1007/3-540-62494-5_6.

[23] Q. Huan, I. C. Avramopoulos, H. Kobayashi, and B. Liu, "Secure data forwarding in wireless ad hoc networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 5, 2005, pp. 3525–3531.

[24] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[25] G. Kounga, C. J. Mitchell, and T. Walter, "Generating certification authority authenticated public keys in ad hoc networks," *Secur. Commun. Netw.*, vol. 5, no. 1, pp. 87–106, 2012.

[26] Remix. *Remix Description*, Apr. 1, 2018. [Online]. Available: http://remix.ethereum.org

[27] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, "Applications of blockchains in the Internet of Things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019.