Computer Assisted Methods in Engineering and Science
30(2): 203–221, 2023, doi: 10.24423/cames.443

This article belongs to the *Special Issue on Scientific Computing and Learning Analytics for Smart Healthcare Systems* edited by Dr. C. Chakraborty, Dr. S. Barbosa and Dr. L. Garg

Decentralized Device Authentication for Cloud Systems with Blockchain Using Skip Graph Algorithm

F. SAMMY*, S. Maria Celestin VIGILA

Department of Information Technology, Noorul Islam Centre for Higher Education, Kumaracoil, Tamil Nadu, India, e-mail: celesleon@yahoo.com

* *Corresponding Author e-mail: fvr.sammy@gmail.com*

Cloud computing provides centralized computing services to the user on demand. Despite this sophisticated service, it suffers from single-point failure, which blocks the entire system. Many security operations consider this single-point failure, which demands alternate security solutions to the aforesaid problem. Blockchain technology provides a corrective measure to a single-point failure with the decentralized operation. The devices communicating in the cloud environment range from small IoT devices to large cloud data storage. The nodes should be effectively authenticated in a blockchain environment. Mutual authentication is time-efficient when the network is small. However, as the network scales, authentication is less time-efficient, and dynamic scalability is not possible with smart contract-based authentication. To address this issue, the blockchain node runs the skip graph algorithm to retrieve the registered node. The skip graph algorithm possesses scalability and decentralized nature, and retrieves a node by finding the longest prefix matching. The worst time complexity is $O(\log n)$ for maximum n nodes. This method ensures fast nodal retrieval in the mutual authentication process. The proposed search by name id algorithm through skip graph is efficient compared with the state-of-art existing work and the performance is also good compared with the existing work where the latency is reduced by 30–80%, and the power consumption is reduced by 32–50% compared to other considered approaches.

Keywords: authentication, blockchain, cloud computing, edge computing, fog computing, latency, power consumption, search by name ID algorithm, single-point failure, skip graph.

1. Introduction

Cloud computing system provides a centralized method of communication between devices connected to the system. It allows the system to access resources for computing facility on-demand with the help of the Internet. There are many

definitions of cloud computing. One such definition given by the National Institute of Standards and Technology (NIST) is as follows: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [1].

The major components of cloud computing [2] are clients, data centers and distributed servers. The end users of the application are called clients. Data centers consist of components to assist applications requested by the clients. To enable reliable communication, multiple servers are spread out geographically to provide service continuously. Cloud computing possesses the following characteristics [3]: pooling of resources, ubiquitous network access, on-demand service requests, and the ability to scale for resources as required. Based on the services, cloud computing defines three layers [4]: infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). NIST defines four cloud models: public cloud, private cloud, community cloud and hybrid cloud.

The major feature of the cloud is the concept of virtualization, which is very cost-efficient in deploying cloud services to cloud providers. Hypervisors of the cloud's major dynamic environment demand proper identification in service deployment. Unfortunately, in a hypervisor, the security menace in one host loses the reputation of the other host, which is a major security concern. In addition, the virtualization technique suffers from the backdoor channel and side-channel attack. which affects the privacy of the cloud systems [44, 45].

The specific security requirements of cloud computing are confidentiality, integrity and availability. The primary problem faced with cloud computing is lack of data privacy as there are many chances of stealing data from cloud service provider [5, 6], service level agreements (SLA) [7], denial of service attacks affecting the availability of service [8], botnet attack [9], address resolution protocol (ARP) spoofing attacks [10], and many more. The major aim of this paper is to authenticate communicating devices. This paper is focused on providing users data privacy with proper device authentication. Many authentication schemes are available as user name and password-based authentication [11–14], multifactor authentication [15, 16], physical, biometric and behavioral Authentication [17–23], public key infrastructure [24, 25], and single sign-on [26].

Although many authentication schemes are available, we aim to provide lightweight decentralized authentication without the need for a trusted third party. Hence, an alternate technology suggested for authenticating devices in cloud computing is to use blockchain technology. All nodes are connected and share the information. The blockchain nodes are searched through the skip graph algorithm where node search is performed by *search by name ID algorithm.*

The major contributions of this work are given below:

a) to address blockchain-based communication in a decentralized manner,
b) to reduce the number of authentication steps involved in blockchain-based communication using existing ECDSA,
c) to dynamically scale devices in the blockchain network, and, for fast nodal retrieval, the skip graph algorithm is effectively implemented in fog nodes to accelerate the authentication process,
d) to compare the effectiveness of the proposed work with the existing methods specified in [27, 28] in terms of time to authenticate and send a message plus the power consumed by the device in the case of worse time complexity in node search.

The work is organized as follows. Section 2 discusses blockchain technology and the existing survey conducted in this research work. Section 3 outlines the proposed architecture and authenticating mechanism used in the current work and Sec. 4 presents in detail the experimental setup and comparative analysis of the current work with the existing work described in [27, 28].

2. Blockchain technology

Blockchain technology is a distributed ledger concept providing an alternate solution to single-point failure as it shares the same transaction across each node. Information is shared in the form of blocks and each block is verified using a consensus mechanism by a mining node and gets added to the blockchain. Each block contains current information and the hash of previous data, and if anyone tries to alter the hashed data, it will be reflected in other nodes and automatically the malicious nodes will be detached from the network. Blockchain can be implemented as a public, private or consortium blockchain.

In the beginning, blockchain was used in financial transactions. Due to its essential characteristics of non-tampering, good fault tolerance and decentralized nature, it has been adopted in other sectors, involving privacy concern, such as health sectors, financial transaction flows, smart transportation and device communication of networks [29, 30] in providing authentication and access control policies to the requesting node.

In the blockchain, to initiate a transaction and block creation, we have nodes called miner nodes. Their primary duty is to validate transactions and block creation. For validating transaction, they execute consensus algorithms such as proof-of-work (PoW) [31], proof of stake algorithm (PoS) [32], Byzantine fault tolerance algorithm [33] and ripple algorithms [34]. The choice of these algorithms depends on the application requirements as minimum power requires a method of easy access facility, less latency and good privacy.

2.1. Skip graph algorithms

A distributed hash table (DHT)-based data structure called skip graph [35] is used in this work to organize the structure of nodes. The *search by name ID algorithm* is used here to search for a node's presence and track its transaction. In this structure, all nodes are considered as a binary string, and based on the node ids the target machine is found using a common prefix length. The distributed recurrent nature of the *search by name ID* algorithm works by searching the node with its name ID matching the common longest prefix instead of doing it with the present level of search. Searching continues from the matching level of node ID to all the matching levels until the node is found. If the node is not present, it returns a NULL value. The worst-case time complexity of searching nodes at a certain level is $O(1)$ and $O(\log n)$ for upper levels with maximum n nodes.

Skip graph implements the search by name method to retrieve the presence of a node in a DHT. It is implemented as a doubly-linked list, and every node is identified with a unique ID in binary representation. It organizes the table in a hierarchical fashion, and nodes with the same binary prefix are present in one level. The higher prefix values are placed in the upper level. In the searching process, the node with the longest prefix is searched, and if not found in the current level, the searching is continued in the next higher level until the node is found. If a node is not present, it returns a NULL value. This process enables dynamic addition of nodes and scales with N nodes in the network. Blockchain-based authentication uses a smart contract procedure, but the rules are static and even smart contracts face security breaches. In this work, the skip graph algorithm is mainly preferred for dynamic scaling of nodes.

2.2. Related works

This section discusses the works on authentication mechanisms in different environments. Symmetric key cryptography is used [36] for device authentication, but the drawback of this approach is storing the symmetric key centrally, causing a single-point failure. Another approach based on physical unclonable functions (PUFs) is presented in [37]. This approach is also subject to a single-point failure as the physical function related to authentication is stored on a server. The work conducted in [38] for device authentication uses group signatures and Shamir's secret sharing scheme, which stores the secret key on every device to recognize the user anonymity, hence requiring more storage. Specific to the cloud environment, working against insider attack using ECC-based mutual authentication in the cloud environment is proposed in [38], but it produces more computation cost in providing mutual authentication. Research work on solution against insider attack using multifactor authentication is presented in [39], where it offers higher security compared to two-tier schemes. However, changing the password

at different tiers is not possible. Most cloud accessing services use two-factor authentication including passwords in first-tier and unique code send to the mobile [40] or email of the user. This type of mechanism suffers due to need for mobile to be continuously serviced. Another research work relied on trusted third party (TTP) [41, 42], where the user saves the data on the server and requests the third-party to verify the security credentials to access the data.

All these existing researches suffer from the drawbacks of insider attack, the need for an effective authentication mechanism and the requirement for extra software and hardware with each security tier. Hence to overcome the security issue due to authentication problem, this work focuses on applying blockchain for decentralized mutual authentication.

2.3. Problem identification

The main concern of devices communicating in the network is to provide identification. Hence authentication is a major concern. Authentication mechanism with centralized architecture in cloud systems suffers from a single-point failure [43] and demands a decentralized structure to overcome security problem. To eliminate these issues, decentralized authentication with less power computation is required [46]. This research work focuses on providing decentralized authentication of devices in the cloud environment using blockchain technique. The work proposed here reduces the time overhead incurred in [27, 28] with effective device registration and data transmission.

3. Proposed system

The objective of this research is to provide authentication between devices connected through blockchain. The communication happens either between devices of the same system or between devices on different systems. The following Subsec. 3.1 describes the architecture of the proposed work and nodes involved in the authentication process.

3.1. Architecture of the proposed work

The architecture consists of blockchain-connected edge devices belonging to different system environments. These edge devices maintain the DHT table and use the search by name ID algorithm for faster node search in authenticating devices. Figure 1 depicts the system architecture of the proposed approach.

The proposed architecture consists of a two-layered structure with a device and network layer. The device layer consists of a variety of components such as sensing devices, actuating devices and different types of systems present within a specific system such as smart home or in different system environments (smart

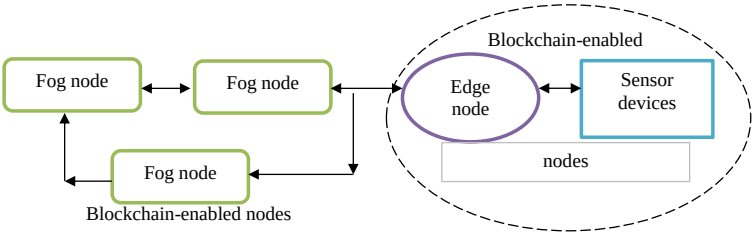

FIG. 1. The system architecture of the proposed approach.

transport). Blockchain-enabled edge devices operate at the network layer that maintains DHT for maintaining the device registration information. All the communications are maintained as transactions and these transactions are shared between all connected edge nodes.

The proposed architecture is well suited for smart healthcare systems where patients, doctors, and lab technicians play the role of end devices and the local server serves as an edge device. These edge devices take the role of admin in registering and authenticating internal device communication. All internal nodes are connected with a private blockchain. The fog node is a cloud node and is connected over the public chain with other fog nodes. Fog nodes are public cloud servers maintaining details about system interconnected. All the fog systems are connected with the public blockchain.

The communication in this proposed approach takes place between fog-fog nodes, fog-edge nodes, edge-device nodes, and between device to device. The fog devices communicate with other fog devices, which share the edge device IDs connected with them. The edge device registers itself with the fog node and shares its DHT with the fog node in fog-edge device communication. The device registers itself to the edge node in device-fog node communication. After stepwise authentication, a successful device-device communication occurs.

3.1.1. Assumptions. We consider the following assumptions prior to describing our proposed mechanism:

a) the admin node is considered a trusted blockchain-enabled edge node that maintains the DHT table,
b) the signature of the blockchain-enabled fog node is communicated securely to the edge and other nodes in the network,
c) the DHT table is maintained by both fog and edge node, where node search is performed with the skip graph method.

3.1.2. Cryptographic algorithm used for key generation. In the current work, the elliptic curve digital signature algorithm (ECDSA) is used to generate public

and private key pairs for making communication between device, edge and fog nodes. ECDSA has the same security strength as that of Rivest-Shamir-Adleman (RSA) algorithm with shorter key pairs. Hashing offers high performance compared to asymmetric key, but one has to share a secret key with verifying party. which needs to be authenticated. ECC provides the same level of security with a shorter key length compared to RSA and a better signing time than RSA. Moreover, it is hard to generate a private key with the given public key and generator value in ECDSA. Hence, the considered approach is efficient as that of RSA with the same key length.

3.2. System process

This section explains the working procedure of the proposed approach and Table 1 details the notations used in the proposed approach.

Table 1. Notations used in the proposed work.

FID	fog node unique id
EID	edge node unique id
DID	device node unique id
PU_f	public key of fog node
PK_f	private key of fog node
PU_e	public key of edge node
PK_e	private key of edge node
PU_d	public key of end device
PK_d	private key of end device

3.2.1. Initialization phase. In this phase, the edge device registers with a blockchain-enabled fog node. The edge node transmits its last five digits hashed MAC to the fog node using the public key of fog node PU_f. The fog node generates a unique ID consisting of the fog system name and last five digits of the edge node hashed MAC and transmits to the edge node with the edge node public key PU_e. The fog node maintains details about the edge node in its DHT table and forwards this information to all other connected fog nodes.

Each device has to register with the edge node for further communication. Each device sends its last five digit hashed MAC to the edge device. This information is maintained as DHT in edge nodes. The edge device transmits the unique ID (certificate) obtained from the fog node to all the devices registering with the edge node. The edge node transmits all the registered device details with their corresponding fog node. Figure 2 explains the sequence diagram of fog-edge communication.

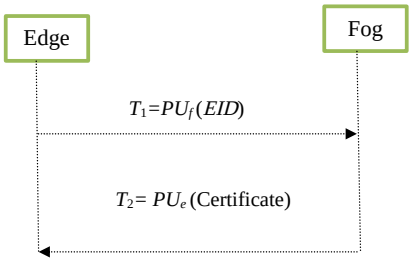

FIG. 2. Sequence diagram of edge–fog communication.

The stepwise procedure for registering a device is explained as follows:

1. Edge admin node sends its last five digit hashed MAC address to the fog node:

$$EID = (\mathrm{SHA}_1\,(\text{MAC address of edge node})),$$

$$T_1 = PU_f(EID).$$

2. On reception of *EID* by the fog node, the private key PK_f generates a certificate containing ids of both the fog and edge node and transmits to the edge node:

$$FID = (\mathrm{SHA}_1\,(\text{MAC address of fog node})).$$

FID contains the last five digits hashed value.

$$\text{Certificate} = (PK_f\,(FID\_EID)).$$

$$T_2 = PU_e\ (\text{Certificate}).$$

3. The edge node extracts the certificate with its private key PK_e.
4. Devices register themselves with the edge node with the last five digit hashed value with the edge node public key PU_e and the edge node provides the certificate using the device public key PU_d:

$$DID = (\mathrm{SHA}_1\,(\text{MAC address of device})),$$

$$T_3 = PU_e(\mathrm{SHA}_1\,(\text{MAC address of device})),$$

$$T_4 = PU_d\,PK_e\,(\text{Certificate}).$$

This is the initial registration phase where every edge node communicates with the fog node by sending its ID (MAC address). The fog node registers the node and returns a certificate for further communication. Similarly, all end nodes register themselves with the edge node and receive this certification for intra-device and inter-device communication. Transaction T_1 indicates the registration of the edge node with the fog node. Transaction T_2 indicates the reception of the certificate from the fog node where PU_f and PU_e represent the public address of the fog and edge node, respectively.

3.2.2. Authentication and communication phase. The authentication phase starts with the initiation of communication by the device. There are two types of device communication. The first type is internal to the same system, and the second type is between devices residing in different systems.

When a device wants to communicate between devices in the same system, it sends a communication request to the edge node. The edge node lookup takes place in the table and if present it authenticates the devices and maps the requested communication.

If the device communication is between devices on different systems, the communication request is made to the fog node for authentication. If the certificate provided by the edge node is presented in the request, the node is authenticated and the communication request is passed to the fog node, which forwards it to the edge node connected. This edge node verifies the requisition and maps the communication. Figure 3 shows the sequence diagram of device-device communication.

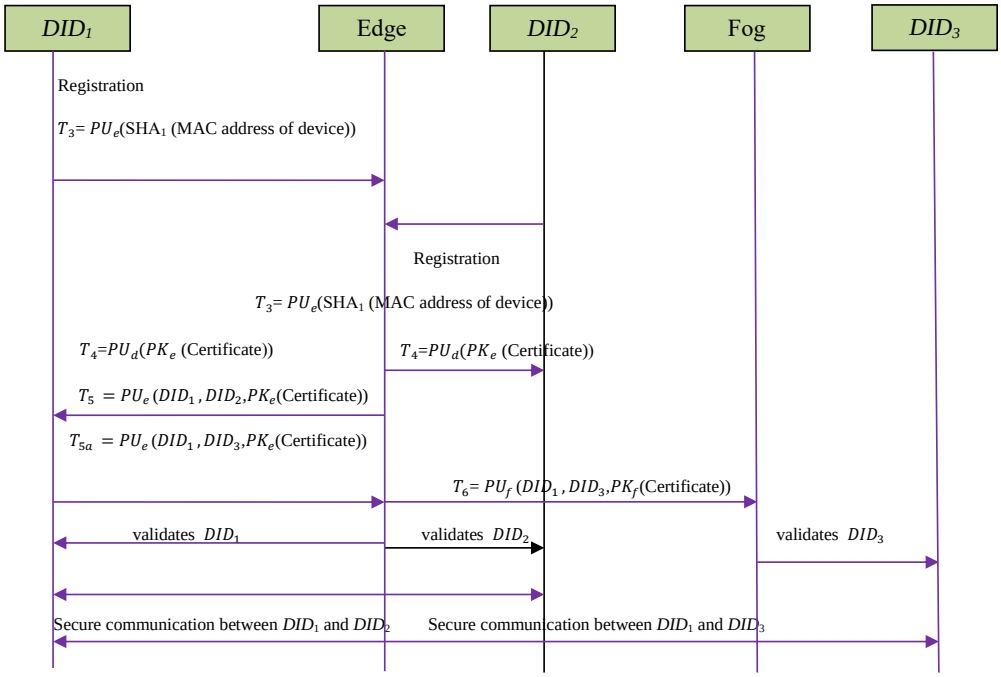

FIG. 3. Sequence diagram of device-to-device communication.

The stepwise procedure for authentication pass is given below:

1. Device D_1 sends a transaction request to the edge device to communicate with device D_2 on the same system:

$$T_5 = PU_e(DID_1, DID_2, PK_e\,(\text{Certificate})).$$

The edge device validates the existence of DID_1 and DID_2 and their corresponding public address in the table. If they exist, then the obtained certificate is validated. If it is valid, it provides an authentication pass and creates a block for communicating between these devices. This block enables them in future communication without re-authenticating.

2. Device D_1 of system A wants to communicate with device D_3 of system B, the device D_1 sends a transaction request to the fog node. The fog node verifies the existence of D_1 and its certificate. If it is true, it forwards the request to all connected fog nodes. The fog node validating D_3 will respond after authenticating D_3. A new block is created and device D_1 in system A communicates with device D_3 in system B. Mapping these devices with a block enables future communication without the need of re-authentication:

$$T_{5a} = PU_e(DID_1, DID_3, PK_e\,(\text{Certificate})),$$

$$T_6 = PU_f(DID_1, DID_3, PK_f\,(\text{Certificate})).$$

The fog device checks the existence of DID_2 and validates it. If validation is true, it maps the communication between DID_1 and DID_2.

3.2.3. System specification. Table 2 shows the system specification used in our work. The fog and edge devices are implemented with laptops, whereas end devices use mobile device.

Table 2. System specification.

Node name	Fog/edge node laptop	End device/mobile device
CPU model	Intel® Core™ i5-4210U CPU @ 1.7 GHz	8 core ARM processor 546.0 MHz – 1.59 GHz
CPU MHz	2400	1590
RAM	1 TB harddrive, 4.00 GB	4 GB

4. Results and discussions

In this study, we compare the performance of our proposed method with earlier work presented in [27] and [28], and evaluated the security performance against the attacks as discussed in the same works.

4.1. Experimental analysis

The work is conducted with four laptops where three laptops act as fog nodes, one laptop as the edge node and three mobile phones as end devices. The

experiment is implemented with C++ similar to [27] for performing interaction among devices with the JsonRPC library.

Figure 4 shows the overall authentication process proposed in this work. This method reduces the overall steps in authenticating process compared to approaches specified in [27, 28].

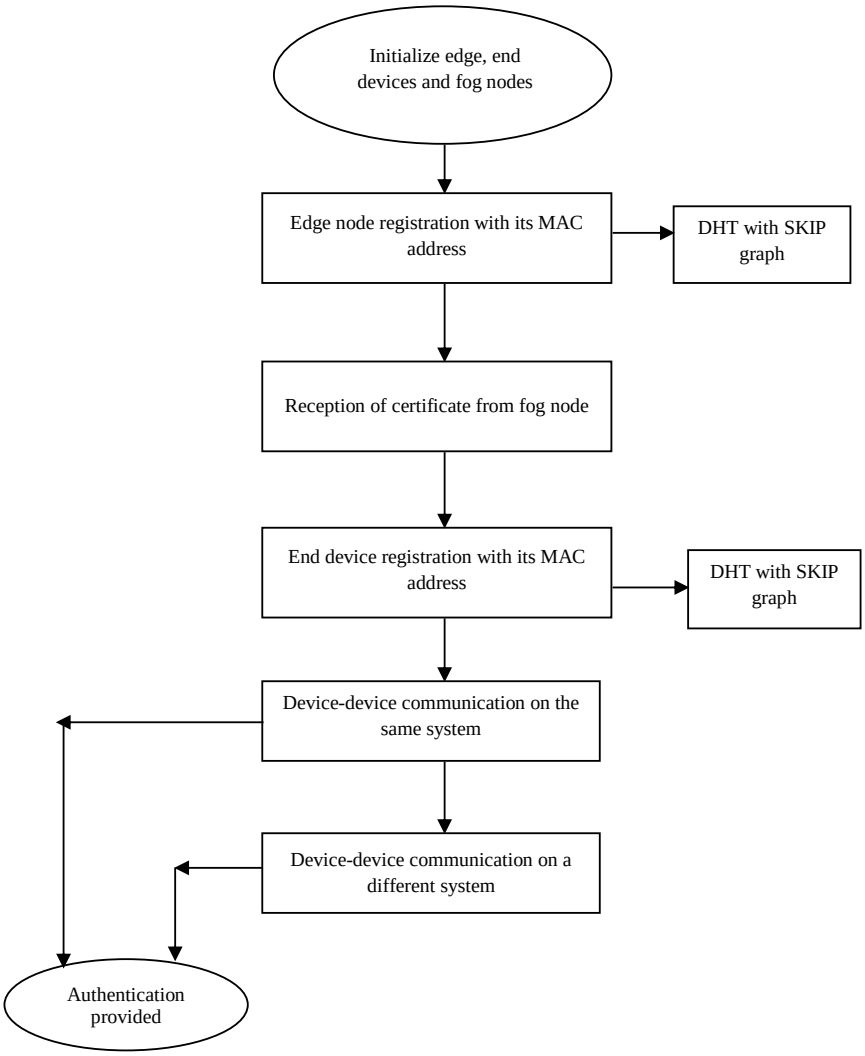

Fig. 4. Flow chart of the overall authentication process.

4.2. Security analysis

This section discusses the security features provided by the proposed work. The security requirements are discussed below.

4.2.1. Identification. Each device in the communication registers its hashed MAC address with the verifying authority. This uniqueness avoids forging as another device.

4.2.2. Non-repudiation and integrity. Transactions are signed by the private key; hence no device can perform repudiation and the data cannot be modified as the private key of the sender signs the hashed data. Hence, integrity is maintained in the system.

4.2.3. Against spoofing and Sybil attack. The forging device cannot get the private key of the original device; hence spoofing attack cannot be launched. Further, the device cannot create a false message or provide fake identities; hence they cannot launch Sybil attacks.

4.2.4. Authentication. After device registration, the device ID is maintained in the DHT table. On validation, the existence of the device is verified in the DHT using the skip graph algorithm. If a node exists and has auth_pass, it is mutually authenticated and establishes trust among the communicating nodes.

4.2.5. Against replay attack. Each transaction has a transaction ID and timestamp; hence a forging node cannot resend the same message, thus preventing a replay attack.

4.2.6. Against substitution attack. Signatures of the senders cannot be forged; hence message modification is not possible, protecting the system against a substitution attack.

4.2.7. Against single-point failure. Decentralized communication with blockchain provides security against a single-point failure.

4.3. Evaluation parameters

The proposed work is evaluated with the same parameters as mentioned in [27, 28] to show the improved performance of the proposed method. The first parameter taken into consideration is the time taken to generate a registration request and sending a message by the nodes. The second parameter is the power consumed by the node in generating a registration request and sending a message.

Tables 3 and 4 show the time taken by the nodes to register and generate a message after authentication. As per the study similar to [27], the experimentation revealed that the proposed work reduces the computational steps and the time taken to conduct registration and data sending.

Table 3. Time taken to register a device.

Compared approach	Laptop [ms]	Mobile device [ms]	Raspberry Pi [ms]
Proposed method	0.87	∼1.12–1.19	–
Khalid *et al.* [27]	1.069	–	24.77
Hammi *et al.* [28]	1.56	–	28.03

Table 4. Time taken to send a message.

Compared approach	Laptop [ms]	Mobile device [ms]	Raspberry Pi [ms]
Proposed method	0.023	0.005	–
Khalid *et al.* [27]	0.03	–	0.0042
Hammi *et al.* [28]	0.04	–	0.029

4.4. Complexity analysis

The time complexity involved in the node search by skip graph at one level is $O(1)$ as it searches with a higher prefix length compared with the target node. If not found, it continues its search at the upper level with the worst time complexity of $O(\log\ n)$. Because of this complexity level, the time taken to register is reduced compared to [27] and [28] by 22% and 80%, respectively. Similarly, the time taken to send a message is reduced by 30% and 42% compared to [27] and [28], respectively.

Tables 5 and 6 show the power consumed by the device to register its identity with the node and proceeding communication through the block chain. As the computational steps are reduced in the proposed approach, the power spent on device registering takes approximately 4.9 mW, and is 32% more efficient than in [27] and 50% more efficient than in [28]. Similarly, the power consumed in

Table 5. Power consumed in generating registration request.

Compared approach	Laptop [mW]	Mobile device [mW]	Raspberry Pi [ms]
Proposed method	4.9	4.87	–
Khalid *et al.* [27]	7.24	–	58.69
Hammi *et al.* [28]	9.76	–	64.16

Table 6. Power consumed in sending a message.

Compared approach	Laptop [mW]	Mobile device [mW]	Raspberry Pi [ms]
Proposed method	2.10	2.23	–
Khalid *et al.* [27]	2.91	–	10.37
Hammi *et al.* [28]	3.35	–	16.29

sending a message by the proposed work is 32% and 52% more efficient than in [27] and [28], respectively.

Figures 5 and 6 show the time taken by the laptop and the mobile device for the proposed approach in registering the device and sending a message in a communication. It is shown that the time taken is shorter than d in the compared approaches.

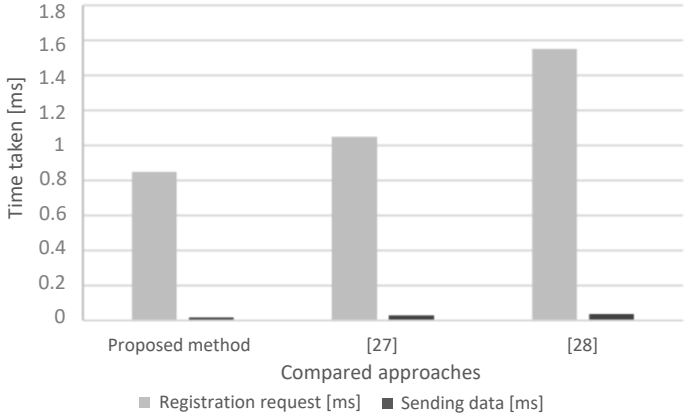

Fig. 5. Time taken by the laptop to register and send data [ms].

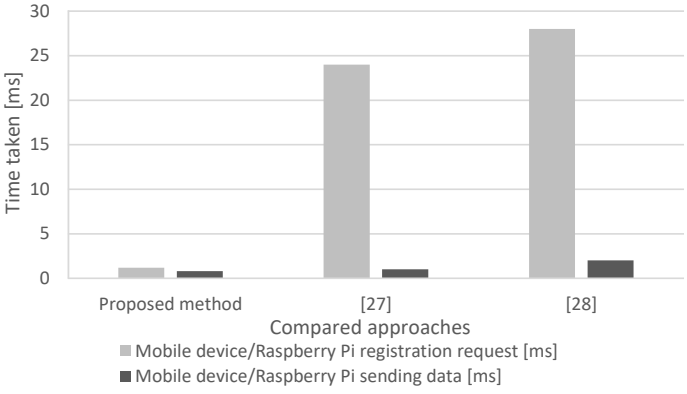

Fig. 6. Time taken by the mobile device to register and send data [ms].

Figures 7 and 8 show the power consumed by the laptop and the mobile device in mW for the transactions taken to complete the registration process and data sending. As the number of computational steps is considerably reduced compared to those of other considered approaches. The proposed approach can be adopted in the future authentication process of cloud systems and light power consuming devices used in IoT.

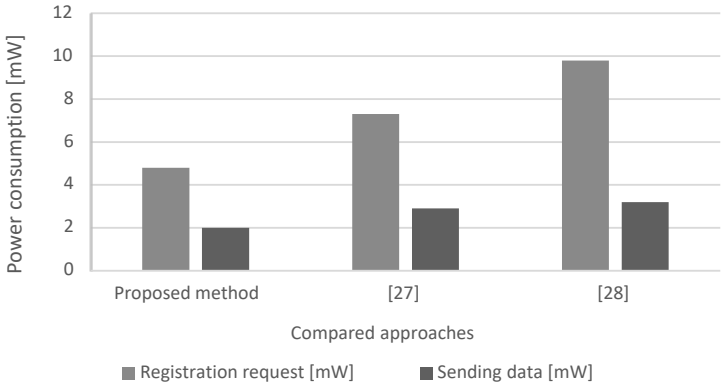

Fig. 7. Power consumed by the laptop to register and send data [mW].

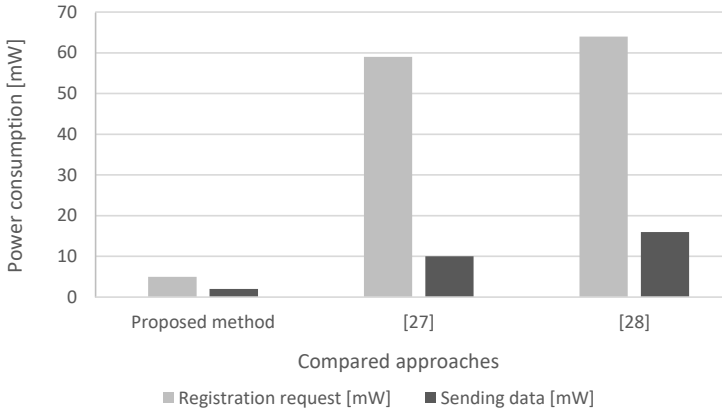

Fig. 8. Power consumed by the mobile device to register and send data [ms].

5. Conclusion

The proposed approach uses the blockchain-based device authentication with the skip graph algorithm for a faster node search problem. Moreover, the computational steps are highly reduced compared to other considered approaches, thereby reducing the power consumption and time of the process. This approach also addresses the security issues of integrity, confidentiality, non-repudiation, replay attacks, Sybil and spoofing attacks. Also it solves a single-point failure. The order of nodal search in authentication takes $O(1)$ for search in the same level and $O(\log n)$ for nodes in the upper prefix level. The latency taken by the proposed work is reduced by 22% and 80% compared with [27] and [28], respectively, for device registration. Further, the time to authenticate and send message is reduced by 30% and 42% compared with that of the other considered approaches. Similarly, the power consumed to register and authentication is reduced by 32%

and 52% compared to that of other approaches. The proposed work contributes to faster retrieval and accelerates the authentication process. In the future, the work will be focussed on applying machine learning techniques and novel cloud architecture to further minimize the latency of the communicating devices.

References

1. P. Mell, T. Grance, The NIST definition of cloud computing, *National Institute of Standards and Technology Special Publication, NIST Special Publication 800-145*, **53**: 1–7, 2011.

2. A.T. Velte, T.J. Velte, R. Elsenpeter, *Cloud Computing: A Practical Approach*, McGraw-Hill, 2010.

3. M. Ahronovitz *et al.*, *Cloud Computing Use Cases*, A white paper produced by the cloud computing use case discussion group version 4.0, 2010.

4. M. Jensen, J. Schwenk, N. Gruschka, L.L. Iacono, On technical security issues in cloud computing, [in:] *2009 IEEE International Conference on Cloud Computing*, 21–25 Sept., Bangalore, India, pp. 109–116, 2009, doi: 10.1109/CLOUD.2009.60.

5. A. Mxoli, M. Gerber, N. Mostert-Phipps, Information security risk measures for cloud-based personal health records, [in:] *International Conference on Information Society (i-Society 2014)*, 1–12 Nov., London, UK, pp. 187–193, 2014, doi: 10.1109/i-Society.2014.7009039.

6. A. Bouayad, A. Blilat, N.E.H. Mejhed, M. El Ghazi, Cloud computing: Security challenges, [in:] *2012 Colloquium in Information Science and Technology*, 22–24 Oct., Fez, Morocco, pp. 26–31, 2012, doi: 10.1109/CIST.2012.6388058.

7. B.R. Kandukuri, Ramakrishna Paturi V., A. Rakshit, Cloud security issues, [in:] *2009 IEEE International Conference on Services Computing*, 21–25 Sept., Bangalore, India, pp. 517–520, 2009, doi: 10.1109/SCC.2009.84.

8. D. Riquet, G. Grimaud, M. Hauspie, Large-scale coordinated attacks: Impact on the cloud security, [in:] *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, 4–6 July, Palermo, Italy, pp. 558–563, 2012, doi: 10.1109/IMIS.2012.76.

9. K. Kourai, T. Azumi, S. Chiba, A self-protection mechanism against stepping-stone attacks for IaaS clouds, [in:] *2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, 4–7 Sept., Fukuoka, Japan, pp. 539–546, 2012, doi: 10.1109/UIC-ATC.2012.13.

10. H. Wu, Y. Ding, C. Winer, L. Yao, Network security for virtual machine in cloud computing, [in:] *5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, 30 Nov. – 2 Dec., Seoul, South Korea, pp. 18–21, 2010, doi: 10.1109/ICCIT.2010.5711022.

11. T. Acar, M. Belenkiy, A. Küpçü, Single password authentication, *Computer Networks*, **57**(13): 2597–2614, 2013, doi: 10.1016/j.comnet.2013.05.007.

12. P. Liu, S.H. Shirazi, W. Liu, Y. Xie, pKAS: A secure password-based key agreement scheme for the edge cloud, *Security and Communication Networks*, **2021**: Article ID 6571700, pp. 1–10, 2021, doi: 10.1155/2021/6571700.

13. S.M. Gurav, L.S. Gawade, P.K. Rane, N.R. Khochare, Graphical password authentication: Cloud securing scheme, [in:] *2014 IEEE International Conference on Electronic Systems, Signal Processing and Computing Technologies*, 9–11 Jan., Nagpur, India, pp. 479–483, 2014, doi: 10.1109/ICESC.2014.90.

14. A.A. Yassin, H. Jin, A. Ibrahim, D. Zou, Anonymous password authentication scheme by using digital signature and fingerprint in cloud computing, [in:] *2012 Second IEEE International Conference on Cloud and Green Computing*, 1–3 Nov., Xiangtan, China, pp. 282–289, 2012, doi: 10.1109/CGC.2012.91.

15. M. Karnan, M. Akila, N. Krishnaraj, Biometric personal authentication using keystroke dynamics: A review, *Applied Soft Computing*, **11**(2): 1565–1573, 2011, doi: 10.1016/j.asoc.2010.08.003.

16. K. Abhishek, S. Roshan, P. Kumar, R. Ranjan, A comprehensive study on multifactor authentication schemes, [in:] N. Meghanathan, D. Nagamalai, N. Chaki [Eds.], *Advances in Computing and Information Technology, Advances in Intelligent Systems and Computing*, **177**: 561–568, Springer, Berlin, Heidelberg, 2013, doi: 10.1007/978-3-642-31552-7_57.

17. E.T. Anzaku, H. Sohn, Y.M. Ro, Multi-factor authentication using fingerprints and user-specific random projection, [in:] *IEEE 2010 12th International Asia-Pacific Web Conference*, 6–8 April, Busan, South Korea, pp. 415–418, 2010, doi: 10.1109/APWeb.2010.44.

18. S. Ziyad, A. Kannammal, A multifactor biometric authentication for the cloud, [in:] G. Krishnan, R. Anitha, R. Lekshmi, M. Kumar, A. Bonato, M. Graña [Eds.], *Computational Intelligence, Cyber Security and Computational Models*, **246**: 395–403, Springer, New Delhi, 2014, doi: 10.1007/978-81-322-1680-3_43.

19. X.C. Jiang, J.D. Zheng, An indirect fingerprint authentication scheme in cloud computing, *Applied Mechanics and Materials*, **484**–**485**: 986–990, 2014, doi: 10.4028/www.scientific.net/AMM.484-485.986.

20. M. Babaeizadeh, M. Bakhtiari, M.A. Maarof, Keystroke dynamic authentication in mobile cloud computing, *International Journal of Computer Applications*, **90**(1): 29–36, 2014, doi: 10.5120/15541-4274.

21. M.A. Ferrer, A. Morales, C.M. Travieso, J.B. Alonso, Low cost multimodal biometric identification system based on hand geometry, palm and finger print texture, [in:] *2007 41st IEEE International Carnahan Conference on Security Technology*, 8–11 Oct., Ottawa, Canada, pp. 52–58, 2007, doi: 10.1109/CCST.2007.4373467.

22. B. Cui, T. Xue, Design and realization of an intelligent access control system based on voice recognition, [in:] *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, 8–9 Aug., Sanya, China, pp. 229–232, 2009, doi: 10.1109/CCCM.2009.5270462.

23. R. Jafri, H.R. Arabnia, A survey of face recognition techniques, *Journal of Information Processing Systems*, **5**(2): 41–68, 2009, doi: 10.3745/JIPS.2009.5.2.041.

24. A.K. Jain, S. Prabhakar, L. Hong, S. Pankanti, Filterbank-based fingerprint matching, *IEEE Transactions on Image Processing*, **9**(5): 846–859, 2000, doi: 10.1109/83.841531.

25. D. Zissis, D. Lekkas, Addressing cloud computing security issues, *Future Generation Computer Systems*, **28**(3): 583–592, 2012, doi: 10.1016/j.future.2010.12.006.

26. J. Chen, G. Wu, L. Shen, Z. Ji, Differentiated security levels for personal identifiable information in identity management system, *Expert Systems with Applications*, **38**(11): 14156–14162, 2011, doi: 10.1016/j.eswa.2011.04.226.

27. U. Khalid, M. Asim, T. Baker, P.C.K. Hung, M.A. Tariq, L. Rafferty, A decentralized lightweight blockchain-based authentication mechanism for IoT systems, *Cluster Computing*, **23**: 2067–2087, 2020, doi: 10.1007/s10586-020-03058-6.

28. M.T. Hammi, B. Hammi, P. Bellot, A. Serhrouchni, Bubbles of Trust: A decentralized blockchain-based authentication system for IoT, *Computers & Security*, **78**: 126–142, 2018, doi: 10.1016/j.cose.2018.06.004.

29. C.H. Lau, K.-H.Y. Alan, F. Yan, Blockchain-based authentication in IoT networks, [in:] *2018 IEEE Conference on Dependable and Secure Computing (DSC)*, 10–13 Dec., Kaohsiung, Taiwan, pp. 1–8, 2018, doi: 10.1109/DESEC.2018.8625141.

30. D. Li, W. Peng, W. Deng, F. Gai, A blockchain-based authentication and security mechanism for IoT, [in:] *2018 27th IEEE International Conference on Computer Communication and Networks (ICCCN)*, 30 July – 2 Aug., Hangzhou, China, pp. 1–6, 2018, doi: 10.1109/ICCCN.2018.8487449.

31. G. Kumar, R. Saha, M.K. Rai, R. Thomas, T.H. Kim, Proof-of-work consensus approach in blockchain technology for cloud and fog computing using maximization-factorization statistics, *IEEE Internet of Things Journal*, **6**(4): 6835–6842, 2019, doi: 10.1109/JIOT.2019.2911969.

32. J. Kang, Z. Xiong, D. Niyato, P. Wang, D. Ye, D.I. Kim, Incentivizing consensus propagation in proof-of-stake based consortium blockchain networks, [in:] *IEEE Wireless Communications Letters*, **8**(1): 157–160, 2019, doi: 10.1109/LWC.2018.2864758.

33. J. Sousa, A. Bessani, M. Vukolic, A Byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform, [in:] *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 25–28 June, Luxembourg, Luxembourg, pp. 51–58, 2018, doi: 10.1109/DSN.2018.00018.

34. B. Chase, E. MacBrough, Analysis of the XRP ledger consensus protocol, *arXiv*, 2018, doi: 10.48550/arXiv.1802.07242.

35. Y. Hassanzadeh-Nazarabadi, A.U. Şahin, Ö. Özkasap, A. Küpçü, SkipSim: Scalable skip graph simulator, [in:] *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2–6 May, Toronto, Canada, pp. 1–2, 2020, doi: 10.1109/ICBC48266.2020.9169426.

36. F. Wu, X. Li, L. Xu, S. Kumari, M. Karuppiah, J. Shen, A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server, *Computers & Electrical Engineering*, **63**: 168–181, 2017, doi: 10.1016/j.compeleceng.2017.04.012.

37. M.N. Aman, K.C. Chua, B. Sikdar, Mutual authentication in IoT systems using physical unclonable functions, *IEEE Internet of Things Journal*, **4**(5): 1327–1340, 2017, doi: 10.1109/JIOT.2017.2703088.

38. P. Gope, B. Sikdar, Lightweight and privacy-preserving two-factor authentication scheme for IoT devices, *IEEE Internet of Things Journal*, **6**(1): 580–589, 2019, doi: 10.1109/JIOT.2018.2846299.

39. A. Singh, K. Chatterjee, A secure multi-tier authentication scheme in cloud computing environment, [in:] *2015 International Conference on Circuits, Power and Computing Technologies*, 19–20 March, Nagercoil, India, pp. 1–7, 2015, doi: 10.1109/ICCPCT.2015.7159276.

40. S.M. Bellovin, M. Merritt, Encrypted key exchange: password based protocols secure against dictionary attacks, [in:] *Proceedings of 1992 IEEE Computer Society Symposium on Research in Security and Privacy (SRSP92)*, 4–6 May, Oakland, California, pp. 72–84, 1992, doi: 10.1109/RISP.1992.213269.

41. P.S. Kumar, R. Subramanian, An efficient and secure protocol for ensuring data storage security in cloud computing, *IJCSI International Journal of Computer Science Issues*, **8**(6): 261–274, 2011.

42. K. Gunjan, G. Sahoo, R.K. Tiwari, Identity management in cloud computing – A review, *International Journal of Engineering Research & Technology*, **1**(4): 1–5, 2012.

43. K. Alhamazani *et al.*, An overview of the commercial cloud monitoring tools: Research dimensions, design issues, and state-of-the-art, *Computing*, **97**(4): 357–377, 2015, doi: 10.1007/s00607-014-0398-5.

44. C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, A survey of intrusion detection techniques in cloud, *Journal of Network and Computer Applications*, **36**(1): 42–57, 2013, doi: 10.1016/j.jnca.2012.05.003.

45. J. Tong, G. Xiong, Y. Zhao, L. Guo, A research on the vulnerability in popular P2P protocols, [in:] *2013 8th International Conference on Communications and Networking in China (CHINACOM)*, 14–16 Aug., Guilin, China, pp. 405–409, 2013, doi: 10.1109/ChinaCom.2013.6694630.

46. K. Amit, C. Chinmay, J. Wilson, A novel fog computing approach for minimization of latency in healthcare using machine learning, *International Journal of Interactive Multimedia and Artificial Intelligence*, **6**(7): 7–17, 2020, doi: 10.9781/ijimai.2020.12.004.

Received December 23, 2021; revised version January 28, 2022;
accepted February 10, 2022