Penerbit
Akademia Baru

Journal of Advanced Research Design

Journal homepage:
https://akademiabaru.com/submit/index.php/ard
ISSN: 2289-7984

Enhanced Mutual Authentication Scheme for Fog Computing using Blockchain Technology

Saif Salah Kareem[1], Bashar I. Hameed[2,*], Humam Khalid Yaseen[2]

[1] Al-Nahrain University, Baghdad 64074, Iraq
[2] Computer Science Department, Al-Imam Al-Adham University College, Baghdad, Iraq

ARTICLE INFO

Article history:
Received 7 February 2025
Received in revised form 24 March 2025
Accepted 15 July 2025
Available online 8 August 2025

Keywords:
Fog computing; cloud computing; mutual authentication; blockchain; security

ABSTRACT

Fog computing is an innovative concept that extends cloud services further and brings their capabilities even closer to end-users by extending them out as far as the network edge. Since edges operate on resources that are closer to the source of data, this goes a process in addressing some of the problems with traditional cloud computing - especially latency. Unfortunately, while this proposal offers some benefits it does not by itself ensure that edge devices are trustworthy or behaving securely. As a result, security continues to be one of the key focus points in fog computing deployment. In this context, authentication is an essential component of any security system. Traditional authentication systems do not work well in the fog computing context; therefore, an efficient mutual i.e. two-way authentication process should be used between edge devices and fog servers because of the low end-to-end latency constraints. This paper proposes an improved mutual authentication scheme specifically designed for fog computing environments to overcome these struggles. Our proposal scheme ensures mutual verification between fog servers and edge devices can effectively strengthen security against potential threats. It makes sure there is very low to no storage overhead on fog servers which increases efficiency, especially in resource-constrained environments. Our scheme is tested through extensive experiments and mathematical analysis.

1. Introduction

The connected web and powerful computers of the digital age have integrated convenience into our, life daily [1]. The most intuitive thing is knowledge sharing and acquisition. The rise of the internet, while providing an unprecedented level of connectivity, had also brought an unanticipated obstacle-how to secure all that information. In terms of data processing, high computing capacity, and data storage and management functions, cloud computing technology has demonstrated its performance. It can thus be an important player in the world's data/demand equation in the coming years. Cloud computing does have a centralized operating mode, however, which can become troublesome when the processed data is raised because it takes longer and therefore loses

[*] *Corresponding author*
E-mail address: bashar_ibrahim@imamaladham.edu.iq

https://doi.org/10.37934/ard.141.1.163188

productivity, especially for real-time applications. As a consequence, there recently emerged a new paradigm known as fog computing to solve these limitations. Fog paradigm is designed to spread cloud resources at the network edge while guaranteeing cloud interaction. Computation, coordination, storage and control operations are thus carried out by pooling local resources in close proximity to the end user. New problems arise with the new fog estimation paradigm. The security of data is one of the architecture's major challenges [2]. Data security is one of the key concerns of the user because of its completely distributed and untrusted nature. Authentication is the entryway to any system of protection, typically involves checking the identity of users. The certificate-based authentication framework provides an effective system for verifying the identity of any system identity in a variety of applications. When an entity uses a certificate, a confidence link between the certificate of the entity and the root certification authority must be verified. This trustworthy relationship is checked by confirmation to the root certificate authority of the contents of all the certificates in the certificate chain [3].

Most users and fog node certificates derive from various authorities in a heterogeneous fog computing architecture. In order to authenticate each other mutually between the users and the fog nodes, the intermediate authorities are awarding certificates in the route chain that meet a trusted user or fog node certificate authority [4]. The distributed system can be endangered by several attacks. An opponent can carry out scams, Dos attacks, replay attacks and other forms of attacks [5]. Many shared security authentication mechanisms were suggested to address these problems. However, the current protocols typically use a server-specified database that is vulnerable to SQL attacks and not suitable for the distributed system to support the authentication method. Claiming mutual authentication certificates would probably cause latency problems. In addition, certified solutions suffer from the effects of scalability because a large number of verification requests can be needed for the central authority. This paper therefore proposes an advanced mutual authentication security scheme based on Blockchain, which can meet the above requirements. In this paper we present the specifications that are not available in current protocols but that the distributed system requires. For instance, to illustrate a Blockchain-based distributed technology and explain the proposed mutual authentication scheme we take the Electronic Commerce network cooperation scenario. The following limitations are commonly found in current remote authentication approaches based on single server architectures:

i. Missing anonymity and privacy. A simple approach to password authentication does not secure the true identity of the user. Regular user operations can be predicted easily.
ii. Missing confidentiality and completeness. Without user's awareness, the historical logs of users stored in, say a local archive, can be changed, which can also lead to user privacy leakages.

The role of fog computing and Blockchain technology was outlined in order to solve these shortcomings. Thus an increased number of anonymous and even secret Blockchain authentications have been suggested for example [6-8]. Much worse, either all access policies or request transactions are in plaintext that can be obtained to evaluate user's everyday lives statistically and thereby impact user privacy. This paper can be summarized as following as its key contributions:

i. This paper proposes the mutual authentication scheme, which is a more advanced and efficient way to provide mutual authentication and secure communication of the network edge. The proposed gains access to the cloud platform through registration. In the next step, we allow mutual authenticate fog nodes with users on their provided credentials from the cloud and reduce reliance on the cloud. As well as the smart contract of Ethereum Blockchain governing the laws and logic of authentication.

ii. The proposed mutual authentication scheme incorporates authentication mechanisms between fog nodes. It leverages blockchain security, hashing methods and public key encryption.

iii. With ReMaster fog node implementation in the case of fog node failure, our proposed provides high device reliability.

iv. The proposed solution enables secure access to any architectural fog node by end-users.

v. The proposed solution enables the fog nodes to mutually authenticate. This needs at least two fog nodes to meet the particular storage criteria.

vi. In addition, our blockchains are privately held and not subject to massive processing powers, which are provided for in order to ensure synchronization, compared to the established public Blockchains such as Bitcoin [9].

vii. The proposed mutual authentication scheme performs exceptionally well in comparative studies based on real-world data, outperforming existing approaches in terms of features of security, functional systems and computational overhead.

The following benefits are offered by our scheme:

i. Dynamic and flexible for system users without having to contact between cloud servers and each fog node upon user registration.

ii. Secure, distributed authentication mechanism across multiple cloud providers.

iii. Low latency because authentication is carried out on the network edge, all users and fog nodes, communicating directly without cloud intermediation.

iv. Adaptive and compact system utilizing the established authentication system for preliminary user verification. In fact, our scheme does not need to construct a new public key infrastructure (PKI) but rather builds its security base on previously deployed cloud-level security systems and extends them to network edges.

v. Low authentication overhead.

vi. Opposition to common attacks and removes the current conceivable database attacks without additional security technologies.

2. Related Work

Many similar studies focused on the authentication issue and proposed different schemes, many of which suffer from some challenges. Hammi *et al.,* [10], an authentication scheme was proposed to enable any Fog user, under the authority of a cloud service provider, to provide mutual authenticate with any Fog server. This scheme requires every Fog Server to hold a secret key in the network for every user. Otherwise, the users can't authenticate these servers. Furthermore, authentication between fog servers was not considered [10]. Several solutions have been proposed, particularly for fog computing, for similar architecture. In a hybrid fog-cloud scheme, which offers a threefold protective mechanism [11], the authors introduced a Fog Computing security scheme. Next, the VPN secures the communication channels by using the machine learning to split traffic. It also uses virtual private networks (VPN). This suspicious source of traffic is validated by authentication of a challenge response. He *et al.,* [12], the author suggested a technique for scalable and enhanced key aggregates (SEKAC) The technique aims to give the parties involved in the contact high security. It encrypts data using dual encryption and the cipher text id is used for decryption. In relation with the data size, the cipher text classes are formed. Ibrahim [13], the author proposed a group authentication using a secret sharing scheme . Each tag and server have a secret key in the scheme. The entire group of tags can be verified simultaneously with just one-round contact between the tag and the server. The server uses its own private key for decryption. The protection feature is based on the secret sharing scheme feature, which is ideal for RFID systems for high computational efficient secret sharing. Khan *et al.,* [14], the authors suggested a method for the authentication of the parties. This is accomplished with a hypothetical deductive approach. This is achieved after checks on certain factors, such as the response time, challenge-response and freshness, the work carries out mutual authentication. The scheme was developed by means of a C++ framework which has been tested for Fog Computing and Cloud Computing in two IoT environments. Kiktenko *et al.,* [15], the author devices for end-users and fog-based computer environments a secure authentication scheme. This mechanism also uses elliptical cryptography, which to date is one of the most powerful algorithms. They are ideal for resource-controlled end devices with smaller keys in length. Kumar *et al.,* [16], the authors suggested a secure anonymous distribution mechanism for key distribution. The authentication of other users and servers is achieved through identity-based encrypting. A SK is developed for further communication after good checking. This method appears to be effective and productive, though. Li *et al.,* [17], the authors proposed an enhanced El-Gamal-based signature technology for ECC authentication. The authors concentrate in Lin *et al.,* [18] on possible hyperphysical risks. They suggested a reliable and stable framework for authentication of messages. Regrettably, Liu *et al.,* [19], the authors pointed out that does not avoid distributed DDoS and offers efficient reciprocal authentication. While multifactor authentication can provide us with greater reliability, the communication mechanism is complicated. In summary, protection is very important and very essential where confidential information is concerned. Protection can be given in a variety of ways, including authentication. In the last few years there have been several authentication schemes which have attempted in different ways to protect the system. Some were insufficiently secure, others were extremely complicated to use in low-power devices [15]. This paper analyses current authentication schemes and proposes a new, secure blockchain-based shared authentication system to address the limitations outlined in Table 1.

Table 1
Related studies comparison

Authors	Function	Limitations
Ibrahim [13]	• Allows any user from the Fog level to authenticate each other with any Fog server. • For each fog server on the network to authenticate users, each fog server must possess a unique secret key.	• Authentication between fog servers was not considered in the scheme.
Maharaja *et al.,* [20]	• A hybrid fog cloud computing security scheme utilizing a three-tiered security scheme to improve safety. • Using a private virtual network (VPN), communication networks are protected and computer traffic is categorized.	• Doesn't provide more security goals • There is no fog authentication.
Pugazhenthi *et al.,* [26]	• Use the dual data encryption method and the ciphertext id is used to decrypt. • In relation to the data size, ciphertext classes are generated	• Lack for fog level and as a result suffers from more computation time overhead.

Liu *et al.,* [19]	• Every tag and server hold a secret key in the scheme. • The whole tag group can be checked all at once with just one round contact. • Secret sharing mechanism based on cryptography.	• Doesn't provide mutual authentication between user and the server and less security goals are provided.
Yao *et al.,* [34]	• Uses the hypothesis-deductive authentication approach. • On the basis of verification of such variables, such as the time of response, challenge-response and freshness.	• Doesn't provide reliability of fog nodes. • More computation overhead for authenticating user to the fog nodes.
Singh *et al.,* [28]	• System with a safe end-user and fog-computing authentication scheme. • Use elliptical cryptography that is actually one of the most effective algorithms.	• The authentication mechanism used consume more computation time. • Doesn't provide reliability and more security goal.
Tsai *et al.,* [31]	• A secure anonymous mechanism is proposed for the key distribution. • Leverages identity-based encryption to authenticate users and servers.	• Does not offer privacy to for user credentials using Canetti-Krawczyk opponents (CK-adversary) template.
Odelu *et al.,* [23]	• Propose an ECC-based El-Gamal style signature technological authentication scheme.	• Suffer from computation cost in the communication process. • fog authentication doesn't exist
Li *et al.,* [17]	• The authentication scheme is possibly stable and anonymous.	• Fails the prevention and efficient mutual authentication of distributed DDoS.
Wu *et al.,* [32]	• Offers efficient reciprocal authentication. • Multifactor authentication provides greater reliability.	• The communication mechanism is complicated.

3. Background

This section presents a brief review of authentication technologies related to the proposed mutual authentication scheme. An introduction about Blockchain technology and framework used in implementation Blockchain is discussed in section 3.1. The proposed mainly depends on fog computing technology thus an overview of fog computing and the difference between cloud computing and fog computing is presented in section 3.2. The implications result from embedding security concerns in fog computing distributed systems and the main security pillars related to the authentication process are discussed in section 3.3.

3.1 Review on Blockchain

Blockchain is a promising emerging technology that has revolutionized the cryptocurrency environment in recent years. The main objective of this technology is the communication and sharing of assets between heterogeneous nodes. This is without any trustworthy central authority depending on this exchange. Every blockchain node does not trust any other node, but trusts the entire Blockchain network. Blockchain and Bitcoin are used in some literature interchangeably, while Bitcoin is a specific implementation of blockchain technology. Here we consider a public blockchain made up of nodes held publicly. Blockchain is also allowed (i.e. new block generation determined by the number of trustworthy nodes), with copyright management applications, authentication, data storage, etc. The Blockchain has six block tables as shown in Figure 1. The blocks from bottom to top are cryptographically linked in chronological order. Blockchain technology uses timer proofing, cryptography and other technologies combined with the distributed block storage system to

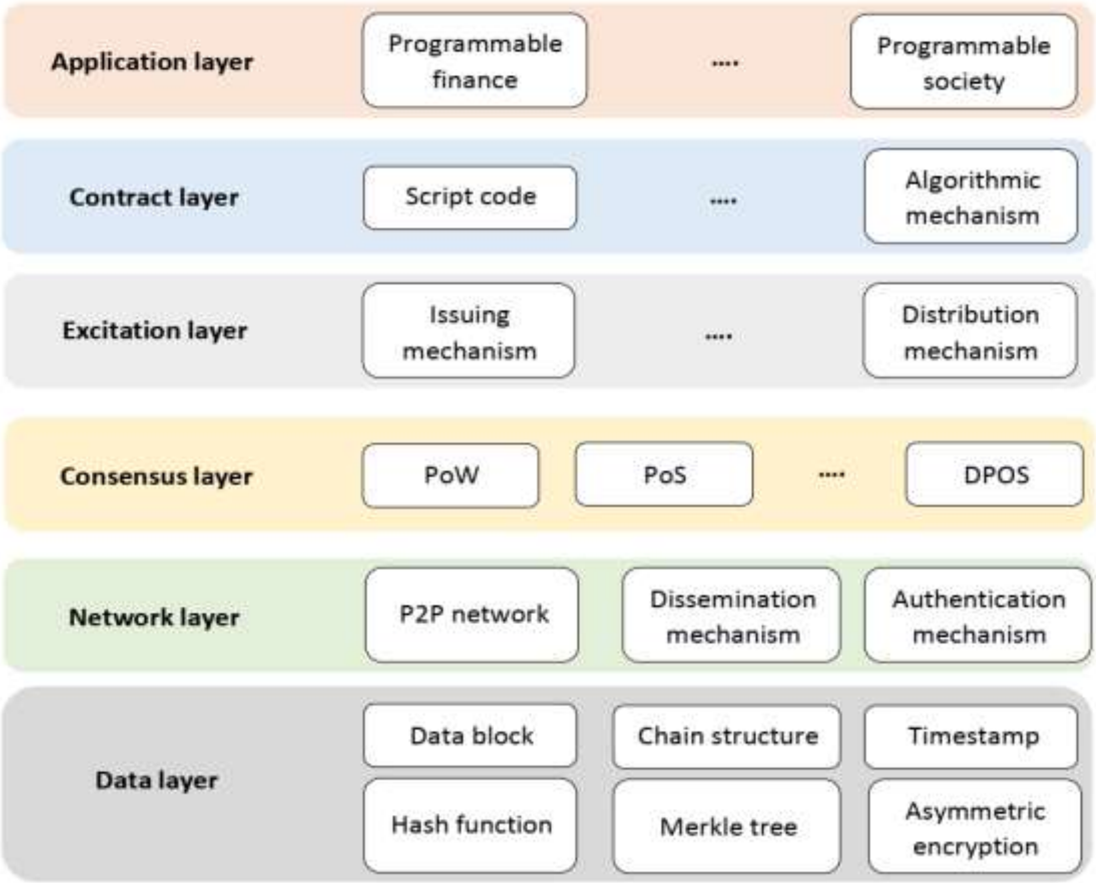

Fig. 1. Framework levels of blockchain

decentralize and difficult for falsifying the Blockchain forgery and collective management, ensuring the protection and confidentiality of important information in the block. In the structure, the data layer is the lower layer and the blocks contain transactional data in the basic unit and use cryptography as a means to build the related data in sequential order such as Hash algorithms and cryptography algorithms. The cryptographic algorithm has two types: symmetrical and asymmetric. In the former one key is encrypted and decrypted and in the latter one private key and public one is split. This is an irreversible derivation of the two. The primary manifestation of Blockchain decentralization is the network layer. The kit comprises a network architecture, a communication interlock protocol and the authentication method [20].

The Blockchain typically has a point-to-point (P2P) network architecture after a long period of growth. The consensus layer covers all algorithms of the consensus process between Blockchain nodes. The data layer, network layer and consensus layer described above are necessary and critical factors in the Blockchain. The incentive layer is often used to recompense the layout of the nodes involved in Blockchain 'mining.' The contract layer covers a piece of contract code which will be executed when the predefined conditions are met. Without a central network manager, Blockchain nodes complete transaction verification in different ways such as Raft [21]. Each node contains one pair of encrypted keys (both private and public) in the Blockchain that allow transactions to be produced and communicate with other nodes in the network. Moreover, certain transactions are unchanging. The majority of validation nodes named miners have to be checked to inject a transaction into a Blockchain. In general, the validation process requires the resolution of a serious computing problem. This function strengthens the immunity property of the Blockchain. Indeed, if you want one block to be fake but validated, for this block and all of its subsequent blocks, you need to apply the same strong validation procedure [22]. Bitcoin, blockchain has become the foundation of several of the most common cryptocurrencies today. However, the possible usage of Blockchain space is endless with the advent of Ethereum Blockchain, which implements intelligent contracts [23]. Ethereum uses its own Ether cryptocurrency. The Ethereum Blockchain network consists of a decentralized platform of thousands of Ethereum Virtual Machine (EVM) and is responsible for the decentralization without the trusted third party (TTP) of application codes (smart contracts) [24]. Smart contracts are a digital Protocol which seeks to reach agreement on the basis of predefined rules between communicating parties without need of a trusted third party. The smart contract is carried out by outside calls and operations, by applying the smart code and events are communicated to all stakeholders.

3.2 Fog Computing and Cloud Computing

The cloud has facilitated the growth of internet access, enabling people to connect from virtually anywhere [25]. Cloud computing strategies are common in many organizations seeking to improve efficiency and innovation [26]. In big data structures as well as in large cloud systems, fog computing can be visible and references the increasing difficulties in the accurate recovery of data [27]. Fog computing extends some cloud capabilities to the edge of IoT networks. This distributed computing scheme puts the ability for processing and data storage right into devices such as mobile phones or IoT sensors directly, leading to reduced latency and speed-up responses in case of data-intensive applications [28]. Fog makes major enhancements to its capabilities, strong security controls and procedures and carefully and flexibly builds data transmission capacities. As patterns are now modified and all businesses demonstrate their importance in technological innovation, Fog Computing's advantages over cloud are massive.

The data that can be sent to the server and which can be handled locally is controlled by Fog computing. Fog and the cloud are related and fog is nearer than cloud to the earth and the same steps are taken in technology. Fog computing enhances the cloud's advantages of stability and productivity [28]. Multiple edge nodes are the components of Fog that can be directly connected to physical devices. The nodes of the fog edge are closer to the physical devices and so fog can supply instance connections. The critical computing capacity of the edge nodes enables a large amount of data to be computed alone without being sent to far away servers [26]. Cloud computing is developed to use remote servers or machines across the internet, rather than use the local computer or server. The service can be provided via the Internet with cloud computing. The facilities cover storage, networking, software, data, etc. From a security perspective, fog computing is very complex since it is connected to multiple nodes and thus retains high security. Fog computing offers all security tests and processes to address and minimize safety risk problems [26]. At various times, in Fog computing, the knowledge pieces are spread over different networks, a massive example of wide bandwidth systems. Low latency in cloud computing is not compared with Fog. Fog computing addresses the challenge of high latency in cloud computing by bringing processing and data storage closer to the edge of the network. Efficiency of power is one of the key advantages. Due to its consistency, performance, speed and processes, many people like Fog computing. Fog computing is very costly since the company has to procure equipment such as gateways, hubs and routers. However, Fog computing spreads cloud capabilities of analysis, computation and storage beyond centralized data centres at the network edges -edges that are often connected to smart objects like sensors- which makes it a good partner for those thinking about Cloud-based services [29]. Fog computing accelerates data processing by decreasing the time needed for applications to respond. Unlike classical cloud computing, which runs on centralized data centres, fog computing provides cloud capabilities to the network edge, accelerating user interactions and improving overall application performance. Cloud technology has three models such as IaaS (infrastructure as a service), PaaS (platform as a service) and SaaS, in a scalable infrastructure (Software as a Service). The management of resources is centralized in Fog computing and is centralized or distributed in the cloud. Table 2 outlines the key differences between cloud and fog computing. The proposed scheme leverages fog computing because its advantages over cloud computing are greater.

Table 2
Comparison of cloud fog computing concepts

Parameter	Fog computing	Cloud computing
Response time	Low	High
Transparency level	High	High
Latency	Minor	High
Distribution of the regional network	Medium	High
Degree of Scalability	High	High
Customization of services	Medium	Low
Generation of content at	Edge device	Central server
Local network dependency	Medium	Low

Figure 2 illustrates the environment of the fog computing network for authentication. In this case, it was presumed that the given scheme initially had a centralized cloud representing the trusted e-commerce site and that F_i $(i = 1,2,...,n)$ and terminals U_j $(j = 1,2,...,t)$ are placed in the network.

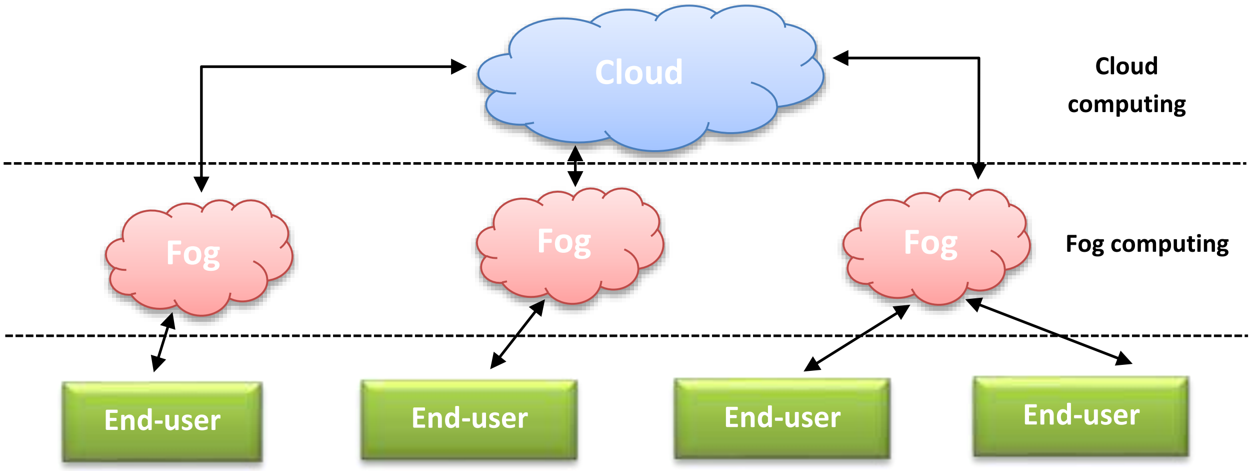

Fig. 2. Fog computing environment

The contact between fog node and cloud will take place sequentially after that end user and fog node. A user initially communicates with a fog node in the standard model of a fog computing architecture. The fog node communicates with the cloud for more resources or services as required. In fog computing, any end user trying to get real-time data should register within the system and receive an authentication to verify their identity. It ensures that only legitimate users can access data and recipients of data are correct nodes within the fog network. Therefore, reliable mutual authentication is necessary, because no one is trusting in the network and all communication is made through an unsecured network. And if an intruder gets the information or the access, it could interrupt the information. Thus, users and fog can each authenticate each other in order to avoid such scenarios. A private blockchain with the consensus mechanism of Raft could be a suitable option to satisfy the requirements of strong consistency and high-speed consensus performance that is a suitable choice. Raft is a widely accepted distributed consensus algorithm with features of high safety and efficiency. This model will ensure that confidential tag information is exchanged for authentication and that the end user's secret data is safeguarded.

3.3 Security Implications of Fog Computing

In today's world, data is quite invaluable. It should be properly maintained from the point of generation to its storage. While Fog computing offers several features, some satiety problems still have to be critically evaluated to secure the data. Table 3 demonstrates the security consequences of the most relevant security concepts for user devices using fog computing. In addition to the features listed in the table, there are many more factors that affect the security of user systems and fog nodes (most significant are CIA, i.e. confidentiality, integrity and availability). In case of failure (capture) in the protection of the fog computing server and user interface, the following security principles are taken into consideration [30]:

i. <u>Access Control:</u> Fog computing devices offer a bridge between the cloud and the user network. Fog devices are in theory gateways that would enable user networks to connect with the data and services available on public cloud resources, but also a potential entry point for threats that expose end users. For instance, a fob server gone rogue might intercept and tamper with data on its path between user devices and the cloud. Hence, there need to be stringent access control policies in place for the data exchange between different devices and users whoever is interacting with these fog environments.

ii. <u>Availability:</u> We expect major effects on the availability of user network services if contact is blocked, depending on the critical location of the fog server. This does not, however, slightly impact the cloud side.

iii. <u>Integrity:</u> We expect to see no fog server effect on the integrity of the messages, depending on the communications system, of any small number (if there is no end-to-end encryption).

iv. <u>User privacy:</u> As with all services, users' privacy is important on the user network and any leakage through fog server devices can have serious implications. All users of the compromised fog server system are affected. The remaining cloud data will, however, remain private and secure.

v. <u>Location Privacy:</u> The end system is normally discharged/communicated to the nearest node. If there is a risk that a fog node is hacked, the hacker would know where the end device has communicated to that specific node is located. Thus, safeguarding the user location is critical.

vi. <u>Authentication:</u> Before launching communication, it is important to check the validity of the fog node so that confidential data can be secured against unauthorized access and only the legit user has access to the limited resources. The exchanging parties should also be authenticated.

vii. <u>Confidentiality:</u> The authentication mechanism is complemented by this feature. This ensures that important information cannot be exposed to any organization unless it is allowed to use it.

viii. <u>Authorization:</u> This feature allows the generated data to only be forwarded to legitimate end users.

ix. <u>Forward secrecy:</u> No more messages from this user are entertained or taken into account after the session is over or the user leaves / moves.

x. <u>Backward secrecy:</u> When a user enters the party, the messages previously transmitted should not be identified.

Table 3
Possible security consideration when the fog computing device is compromised

Feature	Impact on Cloud	Risk on user device
Confidentiality	Minor	Moderate
Integrity	Minor	Minimal
Access control	Minor	Moderate
Availability	Minor	Significant
Authentication	Major/Minor	Moderate
Privacy	Major/Minor	Significant

The combination of heterogeneous terminal devices with Fog computing provides various opportunities and facilities. In order to ensure data confidentiality, connectivity and storage, the above issues must be tackled.

4. Proposed Mutual Authentication Scheme

This section briefly describes a fog computing mutual authentication scheme, which is proposed to address the security holes discussed in some works quantitatively, similar. Table 4 illustrates the abbreviations used in the proposed scheme in section 4.2, which, explained every step of the proposed scheme.

Table 4
Proposed mutual authentication scheme notations

Notations	Meaning
PK_i	Cloud *Bi*'s public key.
SK_i	Cloud *Bi*'s private key.
PK	Public validation key shared among all clouds
SK	Private validation key shared among all clouds
$Ukey$	The public key address generated by the blockchain using user's password
$UsrID$	The user' ID (username) of the end-use account stored at the end-user device.
BC_UsrID	The real user' ID (username) of the end-use account stored at the blockchain.
M	The user'ID (username) of each transaction between end-user device and the fog nodes
F_gPK_i	Public key of Fog node *i*
F_gSK_i	Private key of Fog node *i*
F	Fog node server
C	The blockchain
U	The end-user
$H(*)$	One-way hash function

4.1 Architecture Components

We consider an architecture made up of the following elements in our proposed:

i. Cloud servers that distribute both user and fog nodes authentication credentials so that they are authenticated at the network edge.
ii. Master Fog nodes providing network edge computer resources and device authentication obligations.
iii. ReMaster Fog nodes which considered as a replicate node for the computational services provided at traditional fog nodes at the case of fog node failure to allow more reliability in our proposed.

iv. End-users' devices which request services from fog nodes.

4.2 Proposed Mutual Authentication Scheme Description

The proposed mutual authentication scheme ensures that end-users and fog servers have a good mutual authentication. Our scheme is divided in three phases: Initialization phase, Registration phase and Authentication phase. The three phases description will be discussed below to underline the main contribution of the scheme.

We remember that we do not consider additional access control problems in our scheme as to whether the user is entitled to use an application in the fog node or which services he is entitled to use. In the following, we illustrate how our proposed mutual authentication scheme can be implemented, which enables users and fog nodes to check their authenticity on the edge of the network. In certain points of the authentication process, we note that our approach uses public-core cryptography and therefore consider RSA algorithm [31] as a proposed of key public cryptography for the purpose of illustrating what follows. Our method is based on three basic stages of mutual authentication. Starting stage that initializes the Blockchain and generates the cryptographic keys necessary for authentication. The registration stage for fog nodes at the cloud server level is the responsibility for validating the fog nodes as a Blockchain transaction. This stage is also responsible for registering cloud users to receive the authentication credentials required at further fog level of computation. The authentication process authenticates both fog nodes and end-users. Figure 3 displays the phases of the proposed mutual authentication scheme.

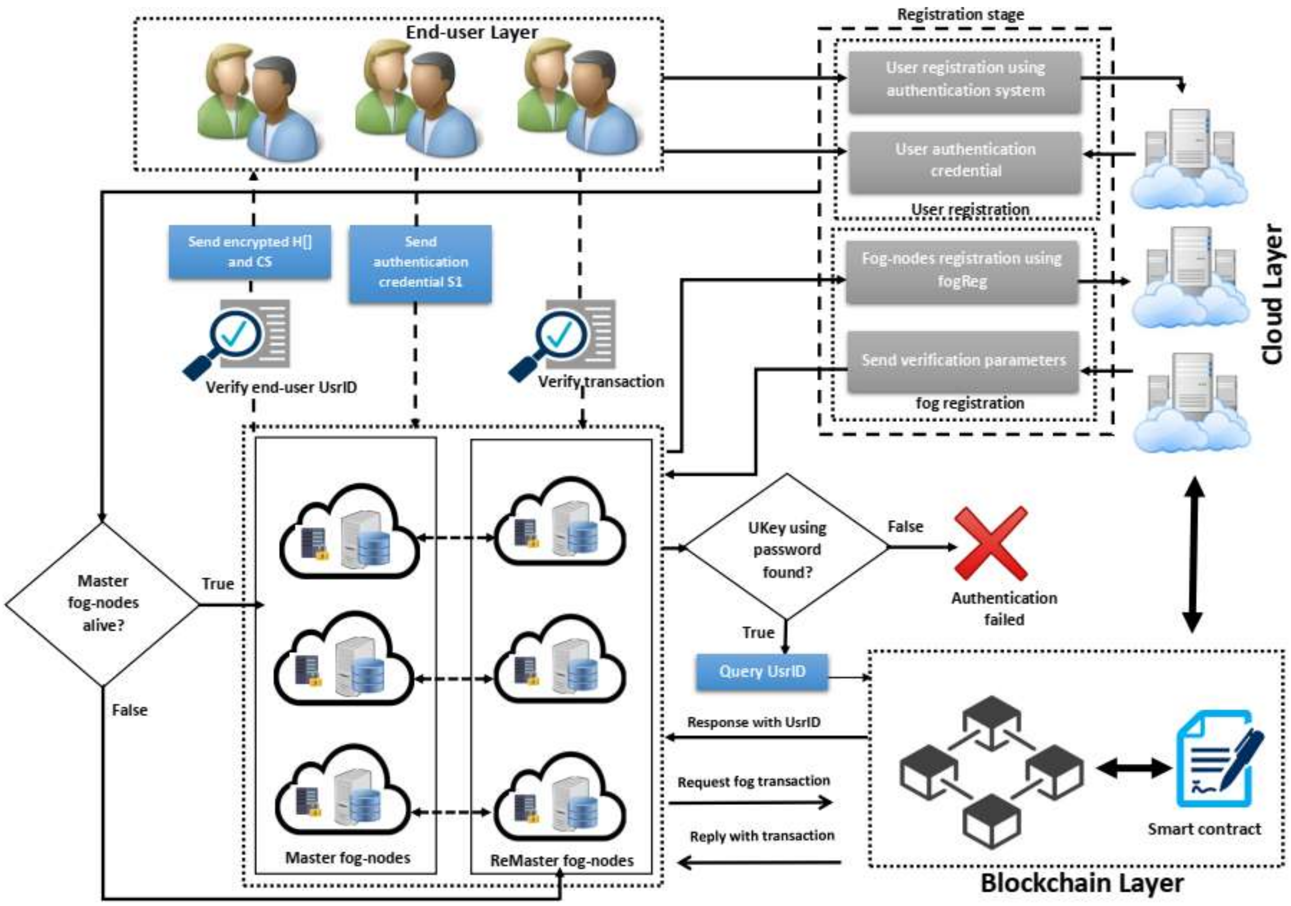

Fig. 3. Proposed mutual authentication scheme

4.2.1 Initialization stage

The cloud server sets the system parameters that will be used in the future authentication and registration phases at this point.

We notice that only one cloud server will be enough to initialize and share parameters with the remaining cloud servers. This will be sufficient for us. Let us therefore take into account that one of the cloud servers is running the following stage:

i. Start a Blockchain that contains each legitimate fog node's public key. We believe each $\mathrm{B_i}$ cloud server has a couple of keys ($\mathrm{PK_i}$ public key and $\mathrm{SK_i}$ private key). The $\mathrm{PK_i}$ key should be known to the other servers as each transaction it produces uses $\mathrm{SK_i}$ to. Furthermore, cloud servers can share other keys in common $(\mathrm{PK}, \mathrm{SK})$. SK is used to sign legitimate transactions, while Pk is used for fog-authentication by users to validate the SK signature.

ii. The Blockchain uses the login password to create public key address as an identifier for account. The record $(\mathrm{Ukey}, \mathrm{UsrID})$ is held by any end-user. As the user is the source of the transaction, the UsrID must initially be accessible to the end-user account, which can be performed in account creation. Blockchain (BC_UsrID) user ID for each end-user account is initialized to a random number between (0, 10), whereas the end-user user ID (UsrID) is initialized with $\mathrm{BC\_UsrID} + \mathrm{M}$.

iii. Two primes u_1, u_2 are chosen and calculate two values $\varphi(n) = (u_1 - 1) \times (u_2 - 1)$ besides $n = u_1 \times u_2$ to be used in the authentication stage.

4.2.2 Registration stage

4.2.2.1 Fog registration

Fog nodes can initially be registered in the cloud by supplying one of the cloud servers with its certificate. Then fog Reg function is used by the cloud server as in the following:

i. Check the fog node's certificate.

ii. Prepare a transaction that includes the public key, the node's current valid state, signed by the private key $\mathrm{SK_i}$.

iii. Put the transaction in a new block and fill in the difficulty field, most importantly, that defines the math issue to be solved in the validation phase.

iv. Transmit the transaction among the pairs so that it is checked.

v. The cloud servers perform a proof of a stack algorithm to validate the transaction, which selects one of the $\mathrm{B_j}$ cloud servers to verify and validate the transaction as follows:

- Check the transaction signature with $\mathrm{PK_i}$ Public Key of $\mathrm{B_i}$ cloud server.
- Once the signature has been validated successfully, solve the math problem in the $\mathrm{B_i}$ block specified by the problem area.
- Fill in the nonce field the solution for the math problem and then sign the transaction with SK.
- Insert into the Blockchain a new block. We notice that the verification in this phase is unrelated to the verification of certificates previously carried out by the broker, only to verify the transaction was produced by one of the relevant brokers.

By inserting the public key of the fog node FPK_i into the Blockchain, The Cloud Server $\mathrm{B_i}$, which has validated its certificate, sends authentication parameters to this legit fog node, computed in the initialisation stage, so that it enables users to authenticate without resorting to the cloud. Figure 4 displays the series of fog registration.

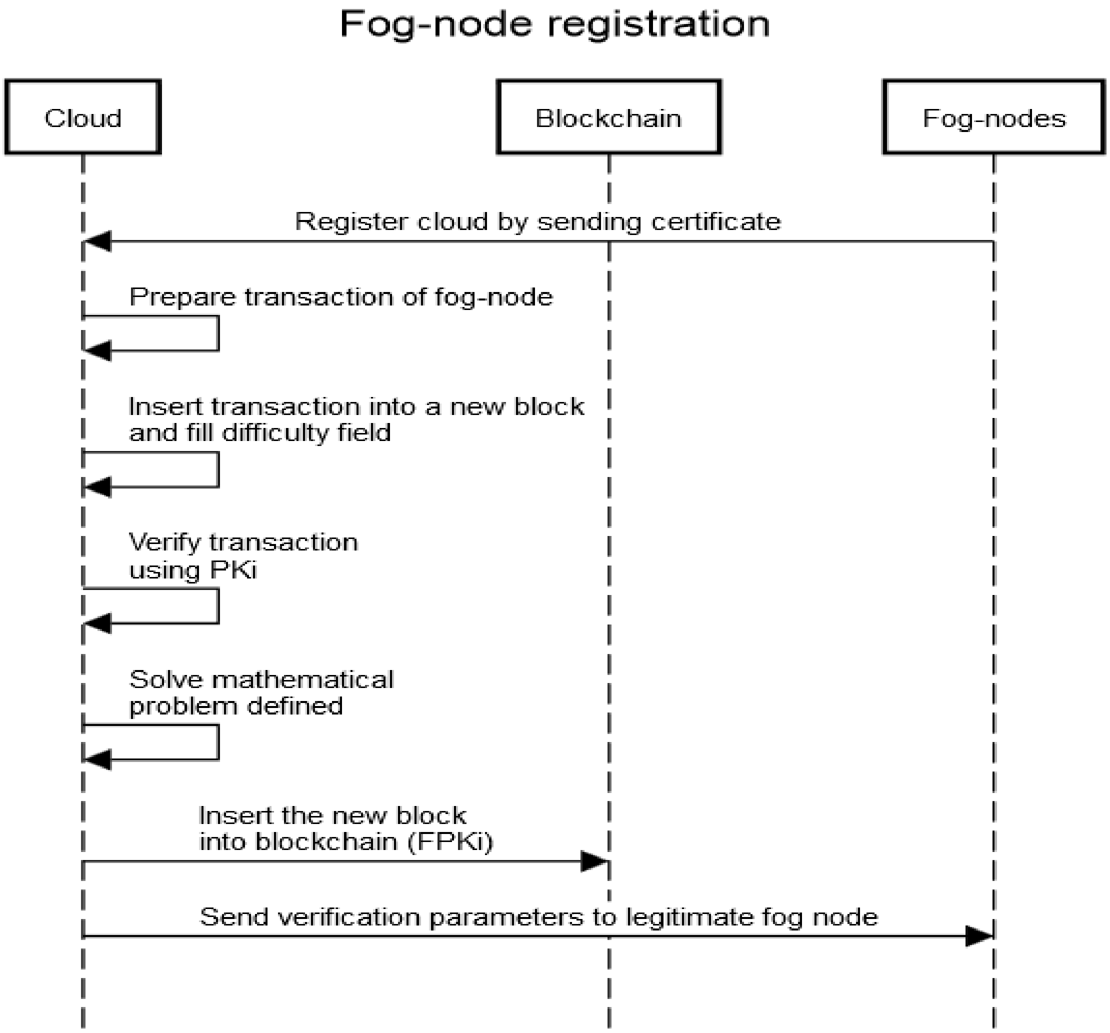
Fog-node registration
Cloud
Blockchain
Fog-nodes
Register cloud by sending certificate
Prepare transaction of fog-node
Insert transaction into a new block and fill difficulty field
Verify transaction using PKi
Solve mathematical problem defined
Insert the new block into blockchain (FPKi)
Send verification parameters to legitimate fog node

Fig. 4. Sequence diagram of fog-node registration

4.2.3 User registration

As Figure 5 indicates, users should also make an initial registration at the cloud level for the user registration stage. A user has to be authenticated successfully using the previously adopted authentication method in the cloud to validate their identity. Afterwards, the cloud server creates new user credentials in order to allow him, at the end of the network (fog node level), to perform any possible authentication as follows:

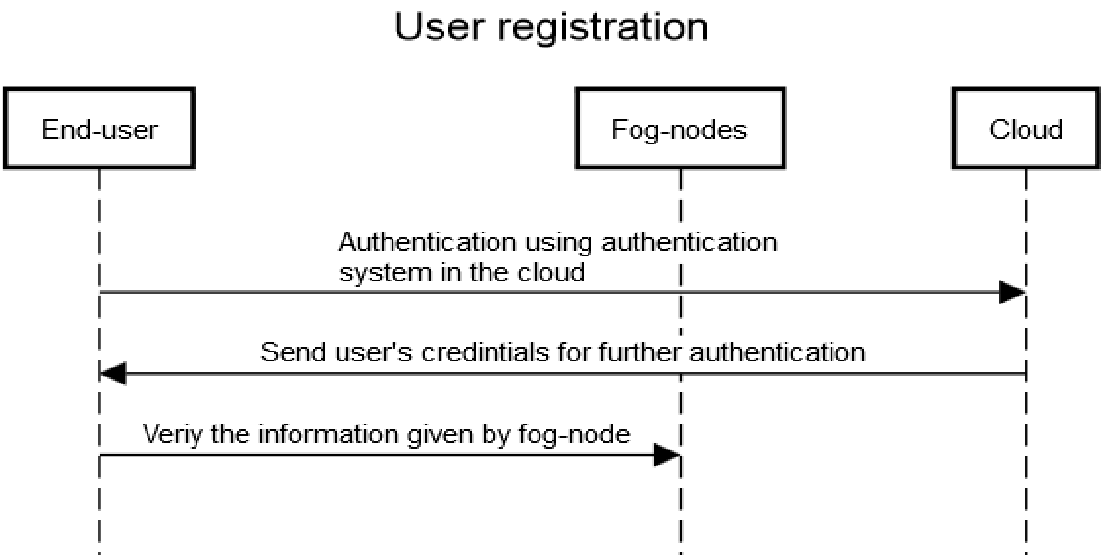
User registration
End-user
Fog-nodes
Cloud
Authentication using authentication system in the cloud
Send user's credintials for further authentication
Veriy the information given by fog-node

Fig. 5. Sequence diagram of the user registration

i. A unique but also a random value $\mathrm{T_i}$ defined from $\mathrm{Z_p}$ that is coprime with $\varphi(\mathrm{n})$ Then, $\mathrm{T_i^{-1}}$ is computed in which it is the modular multiplicative inverse of $\mathrm{T_i}$.
ii. The credentials of the user are prepared accordingly: User's credential = (PK, T_i, T_i^{-1})

Where, PK is the public key to validation. Notice that we want to allow the user to check by sending the public PK key that information provided by the fog node comes from the legitimate and not a falsified Blockchain during the mutual authentication point.

4.2.4 Authentication stage

In the next stage if the master fog-nodes is not available for use because of failure such as network overhead, server maintenance or etc., the ReMaster nodes are used instead. ReMaster nodes are the same as traditional master nodes but they are used to provide system reliability for user requests. Once ReMaster nodes are used, the steps of fog and user authentication can be performed. Using the credentials given by the cloud server and the information in the blockchain, both the users and the fog nodes can mutually authenticate each other at the edge of the network as follows:

i. Fog node authentication: As shown in Figure 6, the user begins by authenticating the fog node by following steps:
- Request the blockchain transaction that inserted and signed the public key and state of the fog node on the cloud server.
- When the fog node returns its transaction block, the user verify that the transaction obtained originates from the blockchain valid for the publication of legitimate fog node by cloud servers.
- A transaction block has therefore been specified as in Eqs. (1) to (3):

$$\mathrm{Block_i = (header, T_x, H(T_x)\sigma_{SK})} \tag{1}$$

$$\mathrm{T_x = (R_i, H(R_i)\sigma_{SK_i}} \tag{2}$$

$$R_i = (F_g PK_i, \text{state}, \text{time}_{\text{stam}}) \tag{3}$$

Where, $header$ refers to the block $Block_i$ header in the blockchain and $state$= valid or not valid.

- H_1 is computed by the user as follow: $H_1 = H(T_x)$.
- Check the signature of the block with the public PK validation key as follows in its credentials as in Eq. (4):

$$H_2 = (H(T_x)\sigma SK_i)^{PK}. \tag{4}$$

- The fog node transaction is verified when H_1 is equal to H_2. The user would otherwise note the fog node does not have a block in the legitimate blockchain because the signature does not match the cloud server's public key PK. Algorithm 1 also provides the steps of authentication of the fog node.

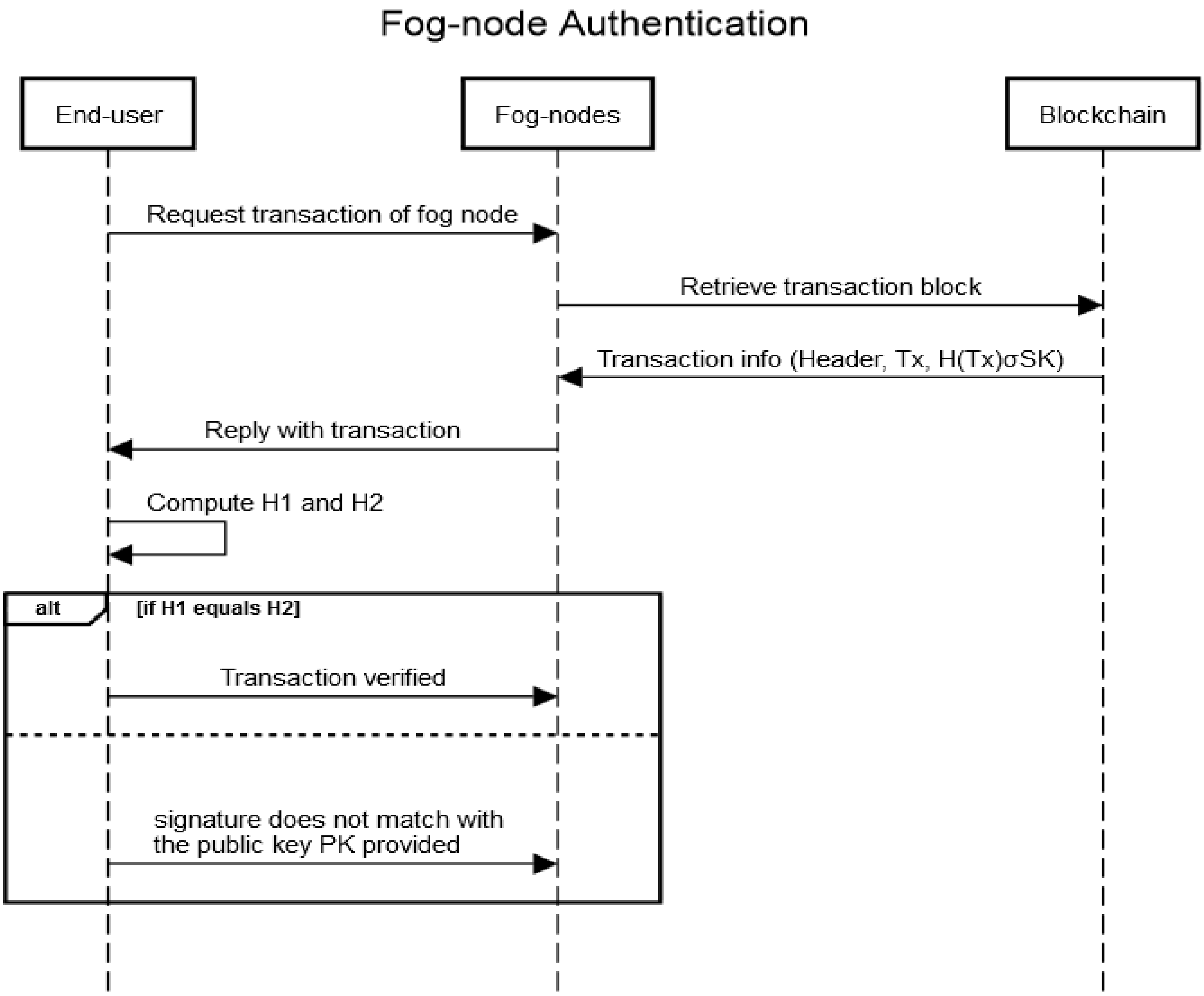
Fog-node Authentication
End-user
Fog-nodes
Blockchain
Request transaction of fog node
Retrieve transaction block
Transaction info (Header, Tx, H(Tx)σSK)
Reply with transaction
Compute H1 and H2
alt
[if H1 equals H2]
Transaction verified
signature does not match with the public key PK provided

Fig. 6. Sequence diagram of fog-node authentication

Algorithm 1 fog-node authentication

```
Function FNAuthen(UsrReq)
//FogAuthenSubPhase
 U request transaction from C
 Transaction info (header, $T_x$, $H(T_x)\sigma_{SK}$) eq. (1),(2),(3)
U computer H1 as A1=H(Tx)
Verify the block's signature using PK as:
   Calculate A2;
   if(A1==A2)
     fog node transaction is verified
   else
     A valid Blockchain fog node was not supplied
//end
```

4.2.5 User authentication

If the user verifies the fog node transaction, the authentication process begins, as shown in Figure 7. The fog node produces a random W number and sends it to the user as follows in Eq. (5):

i. A1 is computed by the end-user computes as follow:

$$A1 = H(Ukey \oplus UsrID \oplus W) \quad (5)$$

Then $S1 = (A1, Ukey, W)$ is sent to the fog node.

ii. At the fog node, when $(A1, Ukey, W)$ is obtained from the end user, the smart contract on the blockchain will immediately commence to continue authentication. Next, look at $Ukey's$ blockchain. The user authentication is failed when $Ukey$ does not found and the server will stop the session; if not, ask for the user's $UsrID$ as BC_UsrID and take the following steps:

- A2 is computed by the fog node as follow in Eq. (6):

$$A2 = H(Ukey \oplus BC\_UsrID \oplus W) \quad (6)$$

- When $A1 == A2$, the last authentication message has been lost or intercepted by the competitor, otherwise the server calculates in Eq. (7):

$$A2' = H(Ukey \oplus BC\_UsrID \oplus M) \oplus W) \quad (7)$$

- If $A1 == A2'$ this will mean a good final authentication.
- If not, the authentication of user is not satisfied, the server stops the session then steps go to step e, Send and step f) is no longer performed.
- If the results of the comparison are consistent with the first, record on Blockchain the interception log for that end-user and BC_UsrID will stay the same (i.e., $BC\_UsrID' = BC\_UsrID$). The fog server unlocks account credentials if it matches the second one. A3 is computed by the fog server as follow in Eq. (8):

$$A3 = H(BC\_UsrID' \oplus W\ ), \quad (8)$$

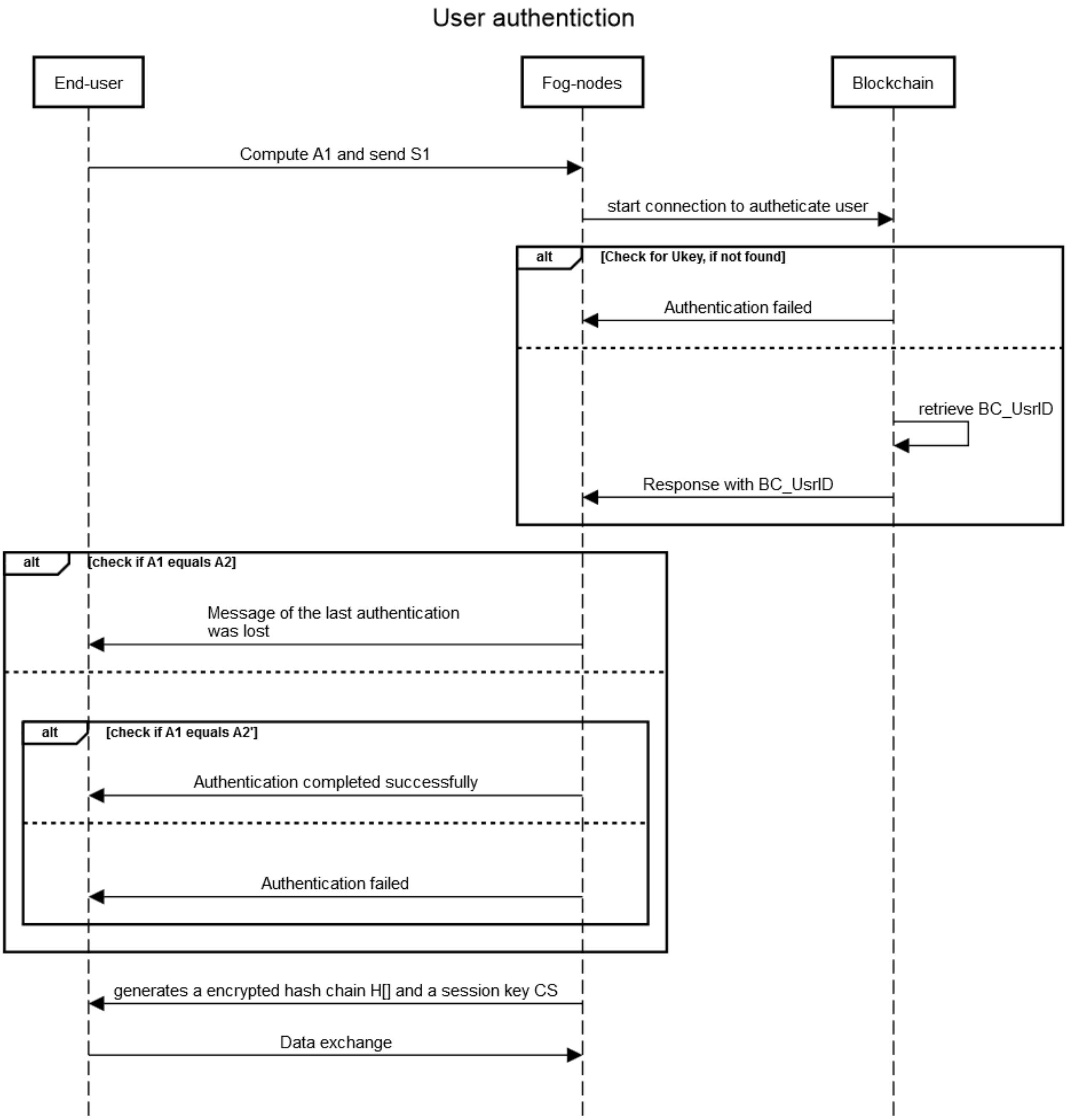

Fig. 7. Sequence diagram of user authentication

and A3 is sent to the end-user.

- A4 is computed also by the end-user as follow in Eq. (9):

$$A4 = H(UserID \oplus W) \tag{9}$$

and checks whether the $A4 == A3$ holds, if it is true, then successful authentication is found and consequently updates $\mathrm{UsrID}' = \mathrm{UsrID} + \mathrm{M}$. The fog node produces the $\mathrm{H}[]$ hash chain and a CS key for use when exchanging data between the fog node and the legitimate user. Then they are encrypted by $\mathrm{T_i}$ as follows in Eq. (10):

$$credential\ access = \{CS, H[]\}T_i \tag{10}$$

Where, $\{*\}\mathrm{T_i}$ is a public $\mathrm{Z_n}$ encryption form that uses a public key of $\mathrm{T_i}$. The end-user declines further contact if it does not hold. The steps mentioned above is summarized in both algorithm 2 and algorithm 3. Algorithm 2 presents the user authentication steps from the view point of the end-user device. Also, algorithm 3 presents steps involved in user authentication phase from the view of fog node server of receiving and sending authentication messages to end-user.

We used public key encryption *via* $\mathrm{T_i}$ value, as can be observed. This value plays an important part in the process which ensures an assurance that only an individual that actually has access credentials from the cloud server can decrypt the fog node. The fog node then sends access credentials and starts the timer that sets the duration for receiving a user's first service request. Then, the user and the cloud providers may create normal secure SSL link for data exchange.

Algorithm 2 user authentication (user level)

```
Function FNAuthen()
//UsrAuthSubPhase-user
Initially: UsrID=BC_UsrID+M
Operations: Hash function, Xor op
After receiving W:
  Calculate:
   $\mathrm{A1 = H(Ukey \oplus UsrID \oplus W)}$
  Send A1, Ukey, W
Upon receiving A3:
  Calculate:
   $\mathrm{H(UsrID \oplus W)}$
 Check If $\mathrm{H(UsrID \oplus W) == A3}$
   Update UsrID'=UsrID+M
  Else: fail
//end
```

Algorithm 3 user authentication (fog level)

```
Function FNAuthen()
//UsrAuthSubPhase-fog
 F send request, W
 After receiving the $\mathrm{(A1, Ukey, W)}$ sent from end-user, check blockchain:
If Ukey not exist: fail
Else, get BC_UsrID:
    Calculate:
      $\mathrm{A2 = H(Ukey \oplus BC\_UsrID \oplus W)}$
      $\mathrm{A2' = H(Ukey \oplus BC\_UsrID \oplus M) \oplus W)}$
     Check If A1==A2, successful:
         record log, $\mathrm{BC\_UsrID' = BC\_UsrID}$
         Calculate:
           $\mathrm{A3 = H(BC\_UsrID' \oplus W\ )}$
      If A1==A2', successful:
          Successful completion of last authentication
                                  $\mathrm{BC_{UsrID}' = BC_{UsrID} + M}$
          Calculate:
             $\mathrm{A3 = H(BC\_UsrID' \oplus W\ )}$
      Else, fail
create a hash chain $\mathrm{H[]}$ and a session key $\mathrm{CS}$ encrypted with $\mathrm{T_i}$
//end
```

5. Proposed Mutual Authentication Scheme Evaluation and Performance

5.1 Threat Model

In the proposed, we differentiate between two separate opponent's models in which each model reflects the following particular situation:

i. In the case of impersonating fog nodes: Let Z be a polynomial time opponent who communicates with a disclosure of signature. It then transmits $\mathrm{m_i}$ to the disclosure arbitrary messages in order to obtain their signature. The opposing person eventually sends out a message m which is never addressed with its signature to the disclosure. Opponent Z wins the safety game if he gives the message m a legitimate signature.
ii. In the case of impersonating end-users: Let A be an opponent of polynomial time that communicates with a random disclosure of encryption. A send $(\mathrm{mg_0}, \mathrm{mg_1})$ to the disclosure two messages. The disclosure then selects a random coin b ∈ $(0,1)$ and responds by sending $\mathrm{E(m_b)}$ where E is a public coding function. Finally, the opponent sends a guess b which has been encrypted by E in one of the two messages sent $(\mathrm{mg_0}, \mathrm{mg_1})$. The gain of opponent A in the game is:

$$Av = P[x = x] - ½ \tag{9}$$

5.2 Security Analysis

In this section, we identify the threats and attacks on the system as a whole briefly and explain the safety needs of our authentication solution. In order to achieve the security objectives, we discuss handling strategies. Availability, Confidentiality and integrity security objectives can be accomplished by sound authentication, monitoring of access and encryption of data [32]:

i. Confidentiality: The need for confidentiality is accomplished by preventing unauthorized access to user data and systems. Using secure SSL login after effective user authentication, also by encrypting the final user credentials at the user authentication stage, is the common approach to achieving confidentiality. The costly public key infrastructure (PKI) for key distribution is ease with our Blockchain based architecture. As explained by Wu *et al.,* [33], any participant with almost no collision, a powerful feature of Blockchain, can be assigned to specific 20 byte Ethereum Addresses (EA) immediately.
ii. Mutual authentication: Both end-user nodes and fog nodes can be authenticated by validating their reported transactions (i.e. the transaction is valid or not).
iii. Perfect forward secrecy: The proposed guarantees that if an attacker gets the public and private keypairs they still cannot decrypt your past communication. Each session will use its own, one-time encryption key.
iv. No verifier table: As stated earlier, only user devices and fog nodes for the public/private key pairs are required for mutual authentication. Obviously, the registration centre has no verifier table.
v. Resilience to interception and modifications: The transactions released have been signed. Once an attacker changes the signature, the transaction will be invalidated, which will avoid unauthorized changes to the reply. Therefore, interception and adjustment resilience are the architecture proposed.

vi. Resilience to hijacking attacks: All signed transactions can also withstand hijacking attacks as any probabilistic polynomial enemy cannot distort the transaction context before rejecting the signatures.

vii. Resilience to other attacks: Our proposal is designed to be resilient against the following types of attacks.

5.2.1 Replay/impersonation attack

The fog node tests the user's credentials and gives him the session key, encrypted by $\mathrm{T_i}$, as a public key in our authentication scheme. Finally, a timer is set and the user is expected. On the other hand, before the timeout is reached, the user must collect the session key and send a service request to the fog node. The authentication session will otherwise expire. The customer must submit a service request within a limited time span as we can see. Therefore, the attempt to replay the user authentication request is futile for any party, since any Party who wants this attack to be successful needs to recover the user's private $\mathrm{T_i^{-1}}$ key and obtain the session key to use it in the eventual service request. $\mathrm{T_i^{-1}}$ is a secret key created by an established, stable public key system, such as RSA [34], which preserves its security. It also makes no use of the identity of the user to be authenticated using their credentials in the fog node, since the attacker also demands that the user recovers the private key $\mathrm{T_i^{-1}}$. In the other hand, if an intruder embodies an established fog node identity, the private key of that fog node used to sign in access identities must be retrieved (the session key and the hash chain). Similarly, if an attacker is attempting to persuade a user that it is a legitimate fog node, a valid Blockchain transaction signed by one of the established cloud servers must be given that includes its public key. Consequently, the assailant must forge the cloud server signature key. A formal proof of its protection was given in respect of the signature of the RSA [35].

5.2.2 Compromise of user/fog

If a fog node is compromised, it does not affect the validity of users in close proximity to other fog nodes as fog nodes have only authentication parameters and no user's $\mathrm{T_i^{-1}}$ key information. A compromised fog node cannot, therefore, execute any sort of attack to use the credentials of each user to gain access to other fog nodes. Any other fog node can still authenticate a user who has been compromised. Therefore, in the event of breaches, users could request a revocation near one of the cloud servers. If a fault is found by the device in any user/Fog node, a cloud server has to cancel it. The block enchain can be an appropriate solution for managing this situation as a repository for the revocation list of all revoked fog nodes/users.

5.2.3 DDoS attacks

The proposed inherits the resilience of Bitcoin to DDoS attacks, which limit the block size.

5.2.4 Modification attacks

If an attacker alters the transaction, it is detected and discarded.

5.2.5 Man-in-the-middle attacks

It is obvious from the above that our proposed ensures effective mutual authentication. It is therefore also the resilience of the attack of man in the middle.

The defence and efficiency capabilities of our proposed are defined in Table 5, as well as, illustrates a comparison of features of security and functionality of our proposed with other literature methods. In contrast to the related protocols Tsai *et al.,* [31], Odelu *et al.,* [23], He *et al.,* [12] and Lin *et al.,* [18]. We pick some common features to compare our proposed. Most literature methods other than He *et al.,* [12] address basic safety needs, such as replay mitigation, anonymity and mutual authentication.

Table 5
Illustrates a comparison of features of security and functionality of our proposed with other literature methods

Feature	Our Proposed	Tsai *et al.,* [31]	He *et al.,* [12]	Odelu *et al.,* [23]	Lin *et al.,* [18]
DDoS prevention	√	×	×	×	×
Replay mitigation	√	√	√	√	√
Resistance to impersonate attack	√	×	√	×	√
Confidentiality	√	√	√	√	√
MITM attack defence	√	×	√	×	×
Dynamic join-and-exit	√	×	×	×	×
Mutual authentication	√	√	√	√	√
Anonymity	√	√	×	√	√

5.3 Security Correctness Proof

We use GNY logic in this paper to demonstrate the consistency of the user authentication process. GNY Review shall be divided into formalizing the messages, defining the presumptions, determining the aims of the protocol and applying the logical postulates. Remember that since the information storage and smart contracting associated with authentication is also in Blockchain, then we combine F and C defined as Y. The research objectives are described below. The security proof is provided in Table 6 and the GNY law is complied with in symbolic forms and written rules and V7 represent the line to prove G1 and G2 proved to step number V13.

Goals:

$\mathrm{G1: Y|\equiv U|\sim\#(H(Ukey\oplus UsrID\oplus W))}$

$\mathrm{G2: U|\equiv Y|\sim\#(H(BC\_UsrID\oplus W))}$

Table 6
Security correctness proof of user authentication mechanism

Number	Proof notation	Postulate
V1	$\mathrm{Y} \lhd^{*} \mathrm{H}(\mathrm{Ukey}\oplus\mathrm{UsrID}\oplus\mathrm{W})$	$\mathrm{Y} \lhd^{*} \mathrm{H}(\mathrm{Ukey}\oplus\mathrm{UsrID}\oplus\mathrm{W}),^{*} \mathrm{Ukey},^{*} \mathrm{W}, \mathrm{T2}$
V2	$\mathrm{Y}\vert\equiv \#(\mathrm{Ukey}\oplus\mathrm{UsrID}\oplus\mathrm{W})$	$\mathrm{Y}\vert\equiv \#(\mathrm{W}), \mathrm{F1}$
If Y authenticates U successfully after judgment, then BC_UsrID = UsrID after update, which means Y∋ UsrID.		
V3	$\mathrm{Y} \ni (\mathrm{Ukey}\oplus\mathrm{UsrID}\oplus\mathrm{W})$	$\mathrm{Y} \lhd^{*} \mathrm{H}(\mathrm{Ukey}\oplus\mathrm{UsrID}\oplus\mathrm{W}), \mathrm{T1}, \mathrm{T2}, \mathrm{P1}, \mathrm{P2}, \mathrm{B} \ni \mathrm{UsrID}$
V4	$\mathrm{Y}\vert\equiv \#\mathrm{H}(\mathrm{Ukey}\oplus\mathrm{UsrID}\oplus\mathrm{W})$	V2,V3,F10
V5	$\mathrm{Y}\vert\equiv \mathrm{U} \overset{\mathrm{UsrID}}{\longleftrightarrow} \mathrm{Y}$	$\mathrm{Y}\vert\equiv \mathrm{U} \Longrightarrow \mathrm{U} \overset{\mathrm{UsrID}}{\longleftrightarrow} \mathrm{Y}, \mathrm{U}\vert\equiv \mathrm{U} \overset{\mathrm{UsrID}}{\longleftrightarrow} \mathrm{Y}, \mathrm{J1}$
V6	$\mathrm{Y}\vert\equiv \mathrm{U}\vert\sim \mathrm{H}(\mathrm{Ukey}\oplus\mathrm{UsrID}\oplus\mathrm{W})$	V1,V2,V3,V5,I3
V7	$\mathrm{Y}\vert\equiv \mathrm{U}\vert\sim \#\mathrm{H}(\mathrm{Ukey}\oplus\mathrm{UsrID}\oplus\mathrm{W})$	V4,V6,F1
V8	$\mathrm{U}\vert\equiv \#(\mathrm{BC\_UsrID}\oplus\mathrm{W})$	$\mathrm{U}\vert\equiv \#(\mathrm{W}), \mathrm{F1}$
If U authenticates Y successfully after judgment, then UsrID = BC_UsrID which means U∋ BC_UsrID.		
V9	$\mathrm{U} \ni (\mathrm{BC\_UsrID}\oplus\mathrm{W})$	$\mathrm{U} \lhd^{*} \mathrm{N}, \mathrm{T1}, \mathrm{P1}, \mathrm{P2}, \mathrm{U} \ni \mathrm{BC\_UsrID}$
V10	$\mathrm{U}\vert\equiv \mathrm{Y}\vert\sim \mathrm{H}(\mathrm{BC\_UsrID}\oplus\mathrm{W})$	V8, V9,F10
V11	$\mathrm{U}\vert\equiv \mathrm{Y} \overset{\mathrm{BC\_UsrID}}{\longleftrightarrow} \mathrm{U}$	$\mathrm{U}\vert\equiv \mathrm{Y} \Longrightarrow \mathrm{Y} \overset{\mathrm{BC\_UsrID}}{\longleftrightarrow} \mathrm{U}, \mathrm{Y}\vert\equiv \mathrm{Y} \overset{\mathrm{BC\_UsrID}}{\longleftrightarrow} \mathrm{U}, \mathrm{J1}$
V12	$\mathrm{U}\vert\equiv \mathrm{Y}\vert\sim \mathrm{H}(\mathrm{BC\_UsrID}\oplus\mathrm{W})$	$\mathrm{U} \lhd^{*} \mathrm{H}(\mathrm{BC\_UsrID}\oplus\mathrm{N}), \mathrm{V8}, \mathrm{V9}, \mathrm{V11}, \mathrm{I3}$
V13	$\mathrm{U}\vert\equiv \mathrm{Y}\vert\sim \#(\mathrm{H}(\mathrm{BC\_UsrID}\oplus\mathrm{W}))$	V10,V12,F1

5.4 Performance Evaluation

This section evaluates the efficiency of the proposed mutual authentication scheme using three distinct configurations for user nodes, fog nodes and cloud instances:

i. User Node: It has at least 1.2 GHz CPU, 1 GB RAM and x86 architecture.
ii. Fog Computing: Fog node consists of an interlayer that is also situated at the edge of a network or just inside the same network [10]. Each image runs Ubuntu 18.04 LTS with vCPU i7 and 8 GB RAM. In turn, we can consider approximately 100m distance between the server and the client.
iii. Cloud Computing: The cloud server is an environment with high and sparsely distributed computing capacity [37]. A Cloud Compute Engine instance has been used that runs Ubuntu 18.04 LTS and at least 8 GB RAM and vCPU i7 was explicitly used as a server that can be approximately 4758 kilometres distance from the server to the clients.

We first calculate the time spent by the server in developing user credentials during the registration process. Then we have our set of authentication measurements and compare it to a solution based on the multi-level certificate. We note that all arithmetic operations take place in $\mathrm{Z_n}$ or $\mathrm{Z_p}$, where p and n are 1024 bits encoded (128 bytes).

5.4.1 Cloud server registration

This process of the cloud server registration verifies that it is indeed the customer and creates user credentials. Note: The major part of this phase involves some multiplication operations giving O(n), time complexity where n is the number of applications simultaneously put by messenger.

5.4.2 Edge level authentication

As shown in Figure 8 edge-level authentication is dynamic. The fog node executes some multiplication and additional processes with our solution only to verify the user's authenticity.

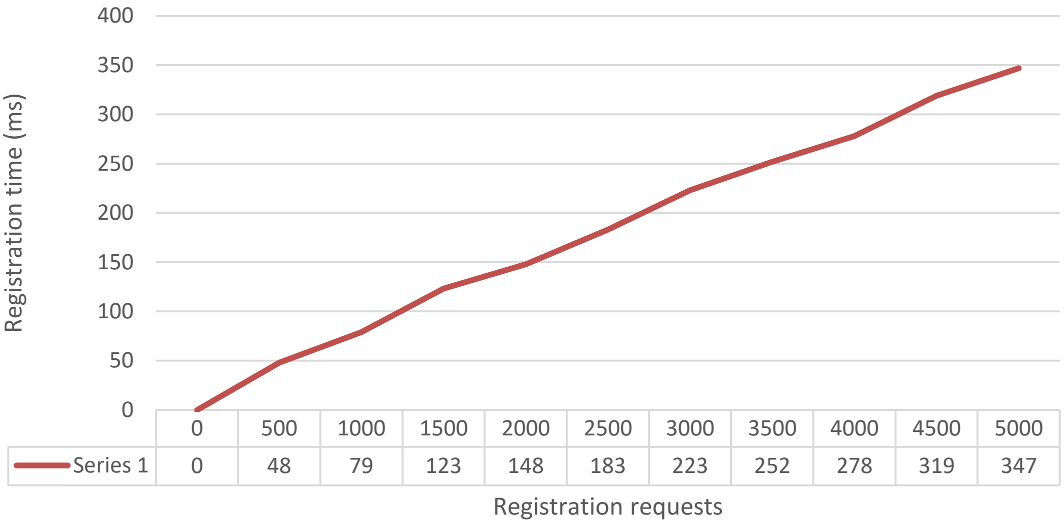

Fig. 8. Registration time at the registration stage

The user simply verifies that an authorized broker has signed the transaction. This verification process is relatively fast, as indicated in Table 6. The authentication process often takes place at the network edge so that no major latency takes place. With the help of current certificate-based techniques, the fog node verifies a series of intermediate certificates up to another trusted CA (Certificate Authority) issued one. Another crucial element to consider is that latency scales with the number of intermediate authorities involved in this verification process. These middle components are located in the same network as each other but not fixed to one part of said network like an electric crack. But if the parties were in different networks, it could introduce some latency.

5.4.3 Computation overhead

Since most costs are created during the period of authentication, we exclude costs during the period of registration. The run-time for Tsai *et al.,* [31] and Odelu *et al.,* [23] are 264.7 and 212.3 ms, is as shown in Figure 9. The total measurement cost of their system is reduced for Yin *et al.,* [36] substituted bilinear coupling by ECC. But Yin *et al.,* [36] scheme's cannot address the centralized problems of the cloud server. In our proposed protocol, we are also developing an authentication protocol which, will simplify, not just the contact round to one round. The results of the experiment also show that among the schemes examined, our proposed protocol demonstrates the lowest overhead.

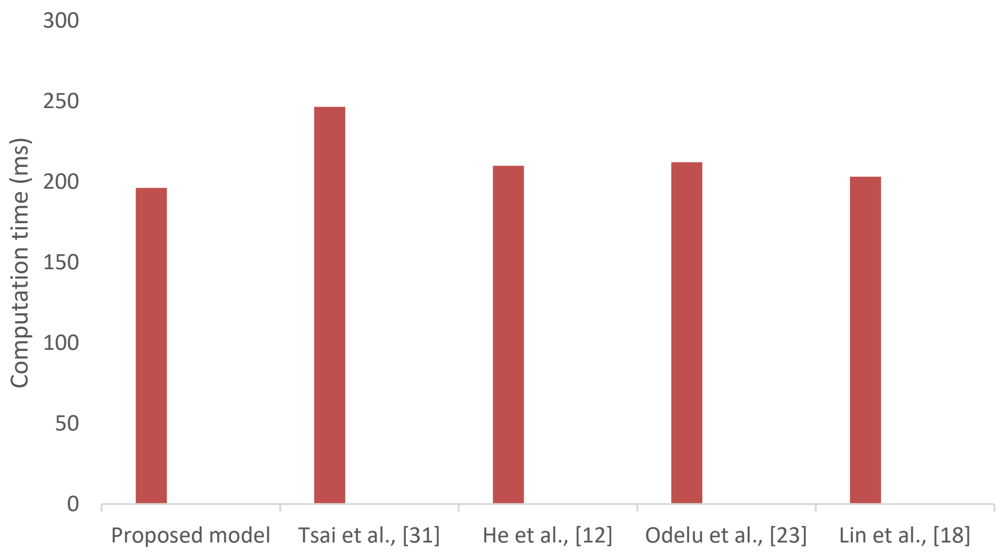

Fig. 9. Computational overhead

6. Conclusion and Future Work

This paper proposes an improved mutual authentication scheme based on blockchain security. An equally critical feature of the blockchain that is also inherited by our solution, thanks to this decentralized and uncontrollable quality which ensures all data remains public yet tamper-proof and every node in a network keeps an up-to-date copy, we use neither trusted third parties nor data centralization. Through our analysis. Our analysis demonstrates that the authentication scheme effectively satisfies security requirements and is free from design flaws. The proposed mutual authentication scheme functional protection is thoroughly demonstrated through a systematic review functional protection is thoroughly demonstrated through a systematic review. Finally, the high performance of our proposed solution has been demonstrated by comparative studies based on real data. We will continue to use the cryptographic elements of secure multiparty computing in our scheme in future efforts in order to improve security in communication processes.

Acknowledgement

This research was not funded by any grant.

References

[1] Amin, Ruhul, Neeraj Kumar, G. P. Biswas, Rahat Iqbal, and Victor Chang. "A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment." *Future Generation Computer Systems* 78 (2018): 1005-1019. https://doi.org/10.1016/j.future.2016.12.028

[2] Andrea, Ioannis, Chrysostomos Chrysostomou, and George Hadjichristofi. "Internet of Things: Security vulnerabilities and challenges." In *2015 IEEE symposium on computers and communication (ISCC)*, pp. 180-187. IEEE, 2015. https://doi.org/10.1109/ISCC.2015.7405513

[3] Berentsen, Aleksander. "Aleksander berentsen recommends “bitcoin: a peer-to-peer electronic cash system” by Satoshi Nakamoto." *21st Century economics: Economic ideas you should read and remember* (2019): 7-8. https://doi.org/10.1007/978-3-030-17740-9_3

[4] Christidis, Konstantinos, and Michael Devetsikiotis. "Blockchains and smart contracts for the internet of things." *IEEE access* 4 (2016): 2292-2303. https://doi.org/10.1109/ACCESS.2016.2566339

[5] Cloud, Hybrid. "The nist definition of cloud computing." *National institute of science and technology, special publication* 800, no. 2011 (2011): 145.

[6] Cramer, Ronald, and Victor Shoup. "Signature schemes based on the strong RSA assumption." *ACM Transactions on Information and System Security (TISSEC)* 3, no. 3 (2000): 161-185. https://doi.org/10.1145/357830.357847

[7] Das, Resul, and Muhammad Muhammad Inuwa. "A review on fog computing: Issues, characteristics, challenges, and potential applications." *Telematics and Informatics Reports* 10 (2023): 100049. https://doi.org/10.1016/j.teler.2023.100049

[8] ElDahshan, Kamal A., AbdAllah A. AlHabshy, and Bashar I. Hameed. "Meta-heuristic optimization algorithm-based hierarchical intrusion detection system." *Computers* 11, no. 12 (2022): 170. https://doi.org/10.3390/computers11120170

[9] Galla, Lavanya K., Venkata SreeKrishna Koganti, and Nagarjuna Nuthalapati. "Implementation of RSA." In *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 81-87. IEEE, 2016. https://doi.org/10.1109/ICCICCT.2016.7987922

[10] Hammi, Mohamed Tahar, Badis Hammi, Patrick Bellot, and Ahmed Serhrouchni. "Bubbles of Trust: A decentralized blockchain-based authentication system for IoT." *Computers & Security* 78 (2018): 126-142. https://doi.org/10.1016/j.cose.2018.06.004

[11] Hazra, Abhishek, Pradeep Rana, Mainak Adhikari, and Tarachand Amgoth. "Fog computing for next-generation internet of things: fundamental, state-of-the-art and research challenges." *Computer Science Review* 48 (2023): 100549. https://doi.org/10.1016/j.cosrev.2023.100549

[12] He, Debiao, Huaqun Wang, Muhammad Khurram Khan, and Lina Wang. "Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography." *Iet Communications* 10, no. 14 (2016): 1795-1802. https://doi.org/10.1049/iet-com.2016.0091

[13] Ibrahim, Maged Hamada. "OCTOPUS: An edge-fog mutual authentication scheme." *Int. J. Netw. Secur.* 18, no. 6 (2016): 1089-1101.

[14] Khan, Minhaj Ahmad, and Khaled Salah. "IoT security: Review, blockchain solutions, and open challenges." *Future generation computer systems* 82 (2018): 395-411. https://doi.org/10.1016/j.future.2017.11.022

[15] Kiktenko, Evgeniy O., Nikolay O. Pozhar, Maxim N. Anufriev, Anton S. Trushechkin, Ruslan R. Yunusov, Yuri V. Kurochkin, A. I. Lvovsky, and Aleksey K. Fedorov. "Quantum-secured blockchain." *Quantum Science and Technology* 3, no. 3 (2018): 035004. https://doi.org/10.1088/2058-9565/aabc6b

[16] Kumar, Vishal, Asif Ali Laghari, Shahid Karim, Muhammad Shakir, and Ali Anwar Brohi. "Comparison of fog computing & cloud computing." *Int. J. Math. Sci. Comput* 1, no. 1 (2019): 31-41. https://doi.org/10.5815/ijmsc.2019.01.03

[17] Li, Xiong, Fan Wu, Saru Kumari, Lili Xu, Arun Kumar Sangaiah, and Kim-Kwang Raymond Choo. "A provably secure and anonymous message authentication scheme for smart grids." *Journal of Parallel and Distributed Computing* 132 (2019): 242-249. https://doi.org/10.1016/j.jpdc.2017.11.008

[18] Lin, Chao, Debiao He, Xinyi Huang, Kim-Kwang Raymond Choo, and Athanasios V. Vasilakos. "BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0." *Journal of network and computer applications* 116 (2018): 42-52. https://doi.org/10.1016/j.jnca.2018.05.005

[19] Liu, Yanxiao, Qindong Sun, Yichuan Wang, Lei Zhu, and Wenjiang Ji. "Efficient group authentication in RFID using secret sharing scheme." *Cluster Computing* 22 (2019): 8605-8611. https://doi.org/10.1007/s10586-018-1929-1

[20] Maharaja, Rajaputhri, Prashant Iyer, and Zilong Ye. "A hybrid fog-cloud approach for securing the Internet of Things." *Cluster Computing* 23, no. 2 (2020): 451-459. https://doi.org/10.1007/s10586-019-02935-z

[21] Mukherjee, Sourav. "Benefits of AWS in modern cloud." *arXiv preprint arXiv:1903.03219* (2019). https://doi.org/10.2139/ssrn.3415956

[22] Niya, Sina Rafati, Florian Schüpfer, Thomas Bocek, and Burkhard Stiller. "A Peer-to-peer Purchase and Rental Smart Contract-based Application (PuRSCA)." *it-Information Technology* 60, no. 5-6 (2018): 307-320. https://doi.org/10.1515/itit-2017-0036

[23] Odelu, Vanga, Ashok Kumar Das, Mohammad Wazid, and Mauro Conti. "Provably secure authenticated key agreement scheme for smart grid." *IEEE Transactions on Smart Grid* 9, no. 3 (2016): 1900-1910. https://doi.org/10.1109/TSG.2016.2602282

[24] Ongaro, Diego, and John Ousterhout. "In search of an understandable consensus algorithm." In *2014 USENIX annual technical conference (USENIX ATC 14)*, pp. 305-319. 2014.

[25] Pan, Jianli, and James McElhannon. "Future edge cloud and edge computing for internet of things applications." *IEEE Internet of Things Journal* 5, no. 1 (2017): 439-449. https://doi.org/10.1109/JIOT.2017.2767608

[26] Pugazhenthi, A., and D. Chitra. "Secured and memory overhead controlled data authentication mechanism in cloud computing." *Cluster Computing* 22, no. Suppl 6 (2019): 13559-13567. https://doi.org/10.1007/s10586-018-2000-y

[27] Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM* 21, no. 2 (1978): 120-126. https://doi.org/10.1145/359340.359342

[28] Singh, Sunakshi, and Vijay Kumar Chaurasiya. "Mutual authentication scheme of IoT devices in fog computing environment." *Cluster Computing* 24, no. 3 (2021): 1643-1657. https://doi.org/10.1007/s10586-020-03211-1

[29] Stojmenovic, Ivan, Sheng Wen, Xinyi Huang, and Hao Luan. "An overview of fog computing and its security issues." *Concurrency and Computation: Practice and Experience* 28, no. 10 (2016): 2991-3005. https://doi.org/10.1002/cpe.3485

[30] Trnka, Michal, Amr S. Abdelfattah, Aishwarya Shrestha, Michael Coffey, and Tomas Cerny. "Systematic review of authentication and authorization advancements for the Internet of Things." *Sensors* 22, no. 4 (2022): 1361. https://doi.org/10.3390/s22041361

[31] Tsai, Jia-Lun, and Nai-Wei Lo. "Secure anonymous key distribution scheme for smart grid." *IEEE transactions on smart grid* 7, no. 2 (2015): 906-914. https://doi.org/10.1109/TSG.2015.2440658

[32] Wu, Libing, Jing Wang, Sherali Zeadally, and Debiao He. "Anonymous and efficient message authentication scheme for smart grid." *Security and Communication Networks* 2019, no. 1 (2019): 4836016. https://doi.org/10.1155/2019/4836016

[33] Wu, Longfei, Xiaojiang Du, Wei Wang, and Bin Lin. "An out-of-band authentication scheme for internet of things using blockchain technology." In *2018 International conference on computing, networking and communications (ICNC)*, pp. 769-773. IEEE, 2018. https://doi.org/10.1109/ICCNC.2018.8390280

[34] Yao, Yingying, Xiaolin Chang, Jelena Mišić, Vojislav B. Mišić, and Lin Li. "BLA: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services." *IEEE Internet of Things Journal* 6, no. 2 (2019): 3775-3784. https://doi.org/10.1109/JIOT.2019.2892009

[35] Yaseen, Humam Khalid, and Ahmed Mahdi Obaid. "Big data: Definition, architecture & applications." *JOIV: International Journal on Informatics Visualization* 4, no. 1 (2020): 45-51. https://doi.org/10.30630/joiv.4.1.292

[36] Yin, Wei, Qiaoyan Wen, Wenmin Li, Hua Zhang, and Zhengping Jin. "An anti-quantum transaction authentication approach in blockchain." *IEEE Access* 6 (2018): 5393-5401. https://doi.org/10.1109/ACCESS.2017.2788411

[37] Zhang, Wenping, and Zhonghua Deng. "Enquiring semantic relations among rdf triples." In *2012 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science*, pp. 477-481. IEEE, 2012. https://doi.org/10.1109/DCABES.2012.48