IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 February 2024, accepted 27 May 2024, date of publication 30 May 2024, date of current version 10 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3407103

RESEARCH ARTICLE

Design of an Efficient and Secure Authentication Scheme for Cloud-Fog-Device Framework Using Key Agreement and Management

MANJUNATH HEGDE[1], **ROHINI R. RAO**[1], **AND RADHAKRISHNA BHAT**[2], **(Member, IEEE)**

[1]Department of Data Science and Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India

[2]Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India

Corresponding author: Rohini R. Rao (rohini.rao@manipal.edu)

ABSTRACT IoT and Smart devices are typically deployed for real-time applications that need to communicate to the cloud infrastructure for data processing and storage. However, the cloud infrastructure has high network latency, and hence the fog has been introduced to form a layered cloud-fog-device framework. The layered architecture requires secure and efficient authentication between all the communicating entities. Secure authentication between fog nodes and cloud servers is not addressed in previous authentication schemes, which can result in severe threats like server masquerading and insider attacks. Ali et al. proposed an authentication key exchange scheme for the cloud-fog-device framework, which was found to be vulnerable to key revelation attacks and failed to provide user anonymity and session secrecy. To overcome the security issues identified, an improved authentication scheme based on key agreement and management was proposed. The scheme authenticates all the entities in the communication, including the cloud server. The scheme secures against privileged insider attacks, ensures user anonymity, untraceability, and session secrecy. The scheme was verified using rigorous cryptanalysis and its security was proved using the ROR model. Formal verification using scyther also confirmed its security against active and passive attacks. An efficiency analysis was performed by comparing the computation and communication costs with other relevant schemes. Functional analysis proved that the proposed scheme exhibits all the functionalities necessary for robust authentication in the cloud-fog-device framework. Overall, the new authentication scheme addresses the security concerns of the cloud-fog-device framework, making it a secure and reliable option for real-time applications.

INDEX TERMS Authentication, elliptic curve cryptography, fog computing, Internet of Things, key management, provable security, security.

I. INTRODUCTION

The emergence of wireless communication has revolutionized communication technology. The Internet of Things (IoT) consists of devices fitted with sensors that connect to networks and process data in cloud computing platforms. Cloud computing provides unlimited storage and processing infrastructure for applications like smart homes, vehicular networks, and smart grids. However, the cloud servers are physically far from edge devices, resulting in an increase in average network latency and jitter [1]. The cloud infrastructure cannot deal with big data generated from edge devices. It cannot scale for real-time applications, which require fast response time, mobility support, high bandwidth, and geo-distribution [2]. Fog servers can be deployed as a lightweight layer between edge devices and the cloud [3], [4].

Fog computing is a distributed computing layer that extends cloud computing services such as storage, processing, and network to edge devices, thereby decreasing service latency. The fog layer consists of small, independent computing entities called fog nodes that are close to the edge devices. These entities are connected to each other

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamad Afendee Mohamed.

as well as centralized cloud servers. The fog nodes work together to pre-process data and provide short-term data storage. Thereby reducing the interaction with cloud servers and improving overall efficiency. Fog nodes can be fog devices that store data, fog servers that can process data, or fog gateways that redirect information between fog devices and servers. Fog computing extends cloud services to large geographical areas. It has features like location awareness, mobility, geo-distribution, distributed control, and real-time interaction, which are required for real-time IoT applications. The fog layer is important because it governs the speed of processing and the flow of information [5], [6], [7], [8], [9], [10], [11], [12].

The Fog-enabled ecosystem improves the overall efficiency of edge devices; however, they have similar security and privacy challenges as the cloud infrastructure. Amongst the security issues, authentication continues to be the most significant challenge [11]. In the fog computing infrastructure, multiple participants interact, and multiple trust domains exist. The multiple entities interact through various layers, which may include untrusted domains [13]. In most authentication schemes, the cloud nodes are assumed to be trusted, whereas the fog nodes and edge devices are not trusted. Fog nodes can be deployed by malicious attackers, and even trusted fog nodes can be easily compromised. IoT devices are usually deployed in places that are not secure and can easily be stolen or invaded by attackers. The heterogeneity of edge devices and computing nodes, and the large geographical area for deployment, add to the authentication challenges in a fog computing ecosystem.

There is a requirement for a reliable and fast authentication mechanism that facilitates communication between smart devices and users with the cloud servers, using the fog node as a mediator. The authentication scheme should be lightweight and possess anonymity and untraceability features. To explore the available authentication schemes for fog computing ecosystems, a survey of existing schemes was performed. The authors compiled a table of important authentication schemes and identified major issues in the cloud-fog-device framework. Subsequently, the major contributions of this research has been listed.

A. RELATED STUDIES

In 2012 Bonomi et al. [14] were the first to present the concept of using fog computing for IoT. In 2015, Stojenovic et al. [15] investigated the security issues, by considering the stealthy features of the man-in-middle attack in a fog computing paradigm. Yi et al. [16] reviewed the security and privacy issues of fog computing and discussed issues such as network security, data privacy, secure computation, secure storage, and intrusion detection. In 2017 Ni et al. [17] presented the security and privacy threats in fog-enabled IoT applications. They also discuss the security and privacy requirements in the fog-enabled infrastructure.

The notable authentication schemes proposed for fog-enabled infrastructure are as follows: In 2018, Imine et al. [18] proposed an authentication scheme for fog computing architecture that adopted blockchain technology for the fog layer. Shamir's secret sharing technique was used to authenticate IoT devices and Fog nodes. The limitation of the authentication scheme is the dependency on a cloud broker, which may result in a trust issue in case the cloud broker is malicious. In 2018, Huang et al. [19] proposed a hardware-based authentication scheme using a physical unclonable function (PUF) to achieve authentication in the fog environment. In the same year, Salem [20] proposed an Elliptic Curve Cryptography (ECC) based, privacy-preserving, mutual authentication scheme for the ''publish-subscribe'' model of fog computing.

In 2019 Wazid et al. [21] proposed a key management and user authentication scheme, SAKA-FC, for a fog computing environment. They adopted a one-way hash function, fuzzy extractor, and ECC. SAKA-FC uses lightweight operations for resource-constrained smart devices, and the scheme also preserves anonymity and untraceability properties. In 2019 Dewanta et al. [22] proposed a mutual authentication scheme for the service handover process in the Vehicular Network Environment. The scheme was proposed for limited access to fog computing service and service reservation during login and service requests. In the same year, Jia et al. [23] proposed a key agreement scheme, using ECC bilinear pairings, for a fog-driven IoT healthcare system, which performs three-party authentication between device, fog, and cloud layer.

In 2020 Chen et al. [24] identified an ephemeral secret leakage attack in the Jia et al. [23] scheme and proposed an improvised authentication and key exchange scheme for fog computing. Wang et al. [25] proposed a lightweight, anonymous authentication scheme for fog computing infrastructure. In this scheme, the communication devices exchange ID and random numbers for registration. The information generated is anonymous and pseudonym information is used to match the recorded random numbers. In 2021 Ali et al. [11] identified that SAKA-FC, the authentication scheme by Wazid et al. [21] is vulnerable to traceability and user impersonation attack and is also inefficient. They proposed an improved authentication scheme for fog computing infrastructure to overcome the identified attacks. The scheme has a similar communication cost as SAKA-FC, but a minor increase in computation cost, and justify it in terms of the robustness of the scheme.

Lin et al. [12] proposed a cross-domain anonymous authentication for multiple servers in a fog-cloud environment. Kalaria et al. [26] proposed a mutual authentication scheme for fog computing in 2021 that utilizes elliptic curve cryptography and one-way hash functions. They propose a lightweight, secure mutual key exchange protocol between cloud, fog, and edge devices. The scheme regenerates the session key for different sessions for secure communication between the fog server (FS) and end-users. In 2021 Guo et al. [27] proposed an authentication scheme that performs mutual authentication between fog users and devices with the help of untrusted fog servers. The authentication scheme achieves low latency with untrusted fog servers and has less computation and communication costs.

In recent studies, Guo et al. [1] proposed an authentication scheme based on ECC and one-way hash functions for the fog-enabled smart home environment. In 2022, Hamada et al. [28] proposed an anonymous mutual authentication scheme for securing fog computing environments. The LAMAS scheme, based on ECC, performs authentication between device, fog, and cloud layers. The fog user stores only one secret key, a short ID, and elliptic curve parameters. Due to its low computation costs with lower storage capacity requirements, it is ideal for deployment in fog computing environments. Ogundoyin and Kamil [2] proposed an authentication and key agreement scheme based on ECC for secure, trust-based communication for fog-to-fog services. In the study of cloud-fog-device authentication, Chatterjee et al. [29] identified that the Wazid et al. authentication scheme, SAKA-FC, is susceptible to insider attacks by fog servers, message intercept attacks, and replay attacks. To address these vulnerabilities, they proposed a lightweight and enhanced remote user authentication and key agreement scheme for IoT communication in a fog-centric setting.

In 2022, Wazid et al. proposed a user authentication and key agreement scheme BUAKA-CS [30] designed for crowdsourcing systems. The scheme is based on the elliptic curve cryptography (ECC) algorithm, provides a secure and efficient authentication process, and ensures the confidentiality and integrity of data during exchange. The scheme uses a blockchain-based smart contract to manage user credentials and facilitate key agreements. Vangala et al. [31] proposed an authentication protocol for IoT-enabled smart agriculture in 2022. The protocol is based on blockchain technology and uses symmetric and asymmetric encryption algorithms to secure communication between sensors, gateway, and the cloud server. The authentication process for IoT-enabled devices is managed through a smart contract.

In 2023, Gowda et al. [32] proposed a blockchain-based secured key management scheme for the Fog Computing Environment (BSKM-FC). BSKM-FC is a decentralized system that does not involve third parties for authentication. Instead, it uses a private blockchain in the fog layer for the generation of private and public key pairs and ECC (Elliptic Curve Cryptography) for secured sharing. Gowda et al. [33] proposed a two-way authentication between edge devices with key management in fog computing environments (TAKM-FC), which uses public-key cryptography and a trusted registration authority to authenticate all entities. In 2023, Akram et al. [34] suggested a fog-based low-latency and lightweight authentication protocol for vehicular communication. The scheme uses elliptic-curve cryptography and hash functions and supports efficient revocation and rekeying mechanisms for improved security. In 2023, Mahesh and Muthumanickam [35] propose a secure authentication scheme for fog environments by identifying forged edge data centers based on cloud-reliant credentials. In 2023, Huo et al. [36] discuss existing authentication schemes for Industrial Wireless Sensor Networks (IWSNs) with Fog Computing and analyze their security strengths and weaknesses. The authors propose an improved scheme to address the limitations of the scheme of Sahoo et al. [37], which is not resistant to user impersonation, tracking, DoS, and replay attacks.

B. MOTIVATION AND RESEARCH CONTRIBUTIONS

The authors have gone through several authentication schemes and found that they are not completely secure, and cannot meet all the security requirements. The authors have compiled some important authentication schemes in Table 1 to provide the clear overview. Upon analyzing Table 1, the authors have identified three major issues in the cloud-fog-device framework.

- The current authentication schemes lack comprehensive solutions for authenticating all entities involved in communication, including devices, users, fog nodes, and servers, leading to vulnerabilities and susceptibility to attacks.
- Some existing schemes rely on the assumption of secure cloud servers, neglecting the importance of authentication between fog nodes and cloud servers, leaving the system vulnerable to server masquerading and insider threats. Furthermore, communication between fog nodes and servers is not always secure, highlighting the need for comprehensive authentication of all entities involved in communication.
- The authentication schemes must minimize computation, communication, and storage overhead, while allowing new devices to be added securely and efficiently.

In this article we analyzed the Ali et al. [11] authentication scheme and identified the security pitfalls. The authentication scheme is vulnerable to key revelation attacks on the smart device, fog node, and cloud server. It is also observed that the scheme does not provide user anonymity, and the session secret key is not secure. Hence, the contributions of the proposed scheme are:

- An efficient and secure authentication scheme for cloud-fog-device framework using key agreement and management. The proposed scheme authenticates all participating entities using the keys the trusted authority (TA) generated. After authentication, registered users and devices can establish a secure session for further communication. Symmetric trivariate polynomial, ECC, and cryptographic hash functions have been used for authentication.
- Rigorous cryptanalysis has been done on the proposed authentication scheme, which is secure against key revelation attacks and provides user anonymity and session secrecy. The proof of security against the adversary model is presented using the real-or-random

TABLE 1. Summary of literature review.

Author	year	Participants	Authenticated Entities	Limitation
Wazid et al.[21]	2019	• Cloud servers, • Fog servers, • Smart devices, • User	• Fog servers • Smart devices • User	• Not all entities are being authenticated • Traceability attack • Clogging attack • insecure parameters • Fog server insider attack • Message intercept attack
Ma et al. [38]	2019	• Device • Fog server • Cloud server	• Device • Fog server • Cloud server	• Privileged-insider attacks • Lost/stolen mobile device attacks • Doesn't preserves anonymity and untraceability
Jia et al. [23]	2019	• Cloud servers, • Fog servers, • Smart devices, • User	• Cloud servers, • Fog servers, • Smart devices, • User	• Secret key leakage • Known session-specific temporary information attack
Amin et al. [39]	2020	• Device • Fog server • Cloud server	• Device • Cloud server	• Not all entities are being authenticated • Privileged-insider attacks • No anonymity and untraceability • No multi-party authentication
Chen et al. [24]	2020	• Cloud servers, • Fog servers, • Smart devices, • User	• Cloud servers, • Fog servers, • Smart devices, • User	• Resistance to intruder node • Replay attack • Masquerading attack • Password guessing attack • Privileged insider attack • Denial of service attack • No clock synchronization
Guo et al. [27]	2020	• User • Device • Fog node • Cloud server	• User • Device • Fog node	• Not all entities are being authenticated • No resistance to unauthorized user • No resistance to on-and-off attack
Guo et al. [1]	2022	• User • Device • Fog node • Cloud server	• User • Device • Fog node	• Not all entities are being authenticated • Device stolen attack • Gateway compromised attack • No Un-traceability • Desynchronization attack • Perfect forward secrecy
Li et al. [40]	2022	• Device • Fog server • Cloud server	• Device • Fog server • Cloud server	• No anonymity and untraceability
Chatterjee et al. [29]	2022	• Cloud servers, • Fog servers, • Smart devices, • User	• Fog servers • Smart devices • User	• Not all entities are being authenticated • replay attack • No multi-level authentication
Gowda et al. [32]	2023	• Cloud servers, • Fog servers, • Smart devices	• Fog servers, • Smart devices	• Not all entities are being authenticated • High latency • Considered fog node as fully trusted

(ROR) model. Also, the security verification was done using the scyther tool, and the results were presented.

- The performance of the proposed authentication scheme is evaluated based on its computation and communication costs and compared with relevant schemes. The functional analysis shows that the proposed scheme reduces the trade-off between security and functionality.

The paper is structured as follows: Section II outlines the system models of the cloud-fog-device framework, including the network model (II A) and the threat model (II B). Section III provides the cryptographic preliminaries necessary for proposing the authentication scheme. Section IV reviews the Ali et al. authentication scheme, detailing the various phases involved. Section V presents the cryptanalysis of the Ali et al. scheme, identifying potential vulnerabilities. Section VI proposes a new authentication scheme for the cloud-fog-device framework. Section VII conducts a security analysis of the proposed scheme, including an informal and formal analysis using the ROR model. Also, this section presents the results of formal security verification using the Scyther tool. In Section VIII, the efficiency of our scheme is evaluated by comparing computation and communication costs, and a functional analysis is also performed. Finally, Section IX concludes the paper.

II. SYSTEM MODELS

This section illustrates mainly the traditional fog-cloud-device network model and the attacker capability where $\mathcal{A}$ can perform various attacks.

A. NETWORK MODEL

A classic cloud-fog-device framework includes a cloud server, fog nodes, and edge devices. The complete system architecture is presented in Figure 1. The Devices could be smart, mobile, or IoT devices that generate real-time data. Fog nodes include gateways, fog servers, and network devices

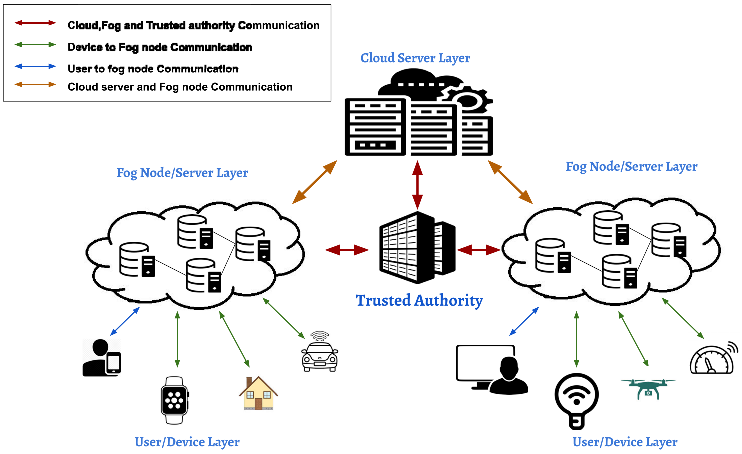

Cloud,Fog and Trusted authority Communication
Device to Fog node Communication
User to fog node Communication
Cloud server and Fog node Communication
Cloud Server Layer
Fog Node/Server Layer
Fog Node/Server Layer
Trusted Authority
User/Device Layer
User/Device Layer

FIGURE 1. System Model - Cloud-fog-device communication framework.

responsible for data processing prior to communicating data to the cloud servers. Cloud servers are made up of data centers and application servers that store and further process data received from fog nodes. The proposed scheme includes Trusted authority (TA), cloud server (CS_l), fog server (FS_j), Smart device (D_k), and user (U_i). These participants are well connected hierarchically, smart devices that generate data, which needs to be transferred to cloud servers via a fog server. A trusted authority is used to generate identities and keys, which are shared with participating entities for authentication.

B. THREAT MODEL

The widely-used Dolev–Yao threat (DY) threat model [41] defines the adversary's capacity to perform any attack. The defined capabilities of the attacker are as follows:

- Adversary $\mathcal{A}$ has complete control over an insecure communication channel. That means $\mathcal{A}$ can intercept, interrupt, forge, and eavesdrop on the messages.
- Edge devices and fog nodes are assumed to be untrusted in the model. They are vulnerable to various attacks; devices can be compromised or stolen by $\mathcal{A}$,
- Attacker $\mathcal{A}$ may perform a dictionary or guessing attack to find the user's password, but $\mathcal{A}$ cannot steal both user and device passwords simultaneously.
- The secret credentials, session states, and session keys in the sessions also may be compromised by Adversary $\mathcal{A}$.
- Adversary $\mathcal{A}$ can extract stored information from fog nodes and smart devices using various techniques, such as differential power analysis attacks and side-channel attacks.
- Hence adversary $\mathcal{A}$ may tamper with fog nodes or smart devices. However, the registration authority is trusted and therefore, cannot be compromised.

In addition to the DY model, the Real or Random (ROR) model is also considered. According to the ROR-adversary model, an attacker can potentially gain access to confidential credentials, session keys, and states within sessions. Moreover, the attacker can capture smart devices and perform a power analysis attack to retrieve stored information.

III. CRYPTOGRAPHIC PRELIMINARIES

In this section, we discuss some important cryptographic concepts that form the foundation of the proposed scheme. Specifically, we'll describe the basics of hash functions, symmetric trivariate polynomials, and the fuzzy extractor.

A. CRYPTOGRAPHIC HASH FUNCTIONS

A cryptographic hash function $h = \{0, 1\}^* \rightarrow \{0, 1\}^n$ is a deterministic function that inputs a string x of random size and produces a fixed-length hash value y.

B. SYMMETRIC TRIVARIATE POLYNOMIAL

The trivariate polynomial $GF(p)$, of degree t over the finite field, is:

$$f(x_1, x_2, x_3) = \sum_{i_1,i_2,i_3=0}^{t} a_{i_1,i_2,i_3}(X_1)^{i_1}(X_2)^{i_2}(X_3)^{i_3}$$

where the polynomial coefficients a_{i_1,i_2,i_3} are randomly picked from $GF(p)$, and p is a large prime for accommodating the cryptographic key.

A symmetric polynomial has the following properties:

Symmetry: A trivariate polynomial $f(x_1, x_2, x_3)$ is said to be symmetric if $f(x_1, x_2, x_3) = f(x_{\sigma(1)}, x_{\sigma(2)}, x_{\sigma(3)})$ for any permutation $\sigma : 1, 2, 3 \rightarrow 1, 2, 3$.

Security: A symmetric polynomial is t-secure, If all coefficients are picked uniformly over the finite field $GF(p)$ [42], [43].

C. FUZZY EXTRACTOR

A fuzzy extractor is a set of randomized procedures that extract an l-bit random string σ in an error-tolerant manner from the biometric characteristic ω that serves as input [44],

[45]. The two randomized procedures of the fuzzy extractor are the probabilistic generation procedure (Gen) and the deterministic reproduction procedure (Rep).

IV. REVIEW OF ALI ET AL. AUTHENTICATION SCHEME

In 2021, Ali et al. [11] proposed a clogging-resistant secure authentication scheme for fog computing services. From the cryptanalysis, we have identified that the scheme is vulnerable to key revelation attacks and failed to provide user anonymity and session secrecy. This section presents the essential phases of the Ali et al. scheme.

A. PRE-DEPLOYMENT PHASE

This phase is for the registration of cloud servers, fog nodes, and smart devices with the Trusted Authority (TA) before deploying to the network.

1) CLOUD SERVERS REGISTRATION

For every cloud server CS_l, TA selects an identity ID_l and computes $d_l = h(K\|ID_l)$ as the private key of CS_l. Further, TA stores $\{ID_l, d_l\}$ in to the cloud server and deploys into the network.

2) FOG SERVERS REGISTRATION

TA selects the identity ID_j for every fog server FS_j and computes the private key $d_j = h(ID_j\|d_l)$ and the public key $P_j = d_j.G$ with respect to the corresponding cloud server CS_l. Further, TA stores $\{ID_j, d_j, P_j\}$ into the fog server's memory and sends $\{ID_j, P_j\}$ to the corresponding CS_l. Lastly TA publicizes the pair $\{ID_j, P_j\}$.

3) SMART DEVICES REGISTRATION

For every smart device D_k, TA picks the identity ID_k and computes $d_k = h(d_j\|ID_j\|ID_k)$ corresponding to the fog server FS_j. Finally, the parameters $\{ID_k, d_k\}$ are stored in D_k memory before the deployment. FS_j is informed about ID_k and stores ID_k in its memory.

B. KEY MANAGEMENT PHASE

Key management is performed with smart devices, fog servers, and cloud servers, through an insecure public channel and a secret key is established with D_k and FS_j.

1) KEY MANAGEMENT BETWEEN SMART DEVICES AND FOG SERVERS

D_k and FS_j are sharing key on an insecure public channel as follows:

- D_k picks a random nonce r_1 and timestamp TS_1, calculates $R_1 = r_1.G$, $R'_1 = r_1.P_j$ and $r'_1 = h(R_1\|TS_1\|d_k)$ and transmits the message containing $\{ID_k, R'_1, r'_1, TS_1\}$ to FS_j
- On receiving this message, FS_j the freshness of the message is checked based on the condition $TS_1 - TS^*_1 \leq \Delta T$, if true FS_j calculates $R_1 = R'_1.d_j^{-1}$, $d_k = h(d_j\|ID_k\|ID_j)$ and checks the condition $r'_1? = h(R_1\|TS_1\|d_k)$. If the condition is true it picks a random nonce r_2, presents timestamp TS_2, and calculates $R_2 = r_2.G, R'_2 = r_2.P_k, K_{jk} = h(R_1\|R_2\|TS_2)$ and $r'_2 = h(R_2\|TS_2\|K_{jk})$. FS_j now sends the message containing $\{ID_j, R'_2, r'_2, TS_2\}$ to D_k.
- On receiving the message from FS_j, D_k checks the freshness of the timestamp by examining the condition $|TS_2 - TS^*_2 \leq \delta T$. If true, D_k calculates $R_2 = R_2.d_k^{-1}$ and computes $K_{jk} = h(R_1\|R_2\|TS_2)$. Now D_k checks $r'_2? = h(R_2\|TS_2\|K_{jk})$. On success, D_k stores K_{jk} for secure communication in the future.

2) KEY MANAGEMENT BETWEEN FOG SERVERS AND CLOUD SERVERS

Key management establishes the secret key between FS_j and CS_l. It is performed over an insecure public channel. The steps are as follows:

- FS_j choose a nonce r_3, timestamp TS_2, and calculates $R_3 = r_3.G$, $R'_3 = r_1.P_l$ and $r'_3 = h(R_3\|TS_3\|d_j)$ and transmits the message $\{ID_j, R'_3, r'_3, TS_3\}$ to cloud server.
- CS_l receives $\{ID_j, R'_3, r'_3, TS_3\}$ and checks the freshness of message by checking the condition $TS_3 - TS^*_3 \leq \Delta T$, if true CS_l calculates $R_3 = R'_3.d_l^{-1}$, $d_l = h(K\|ID_l)$ and checks $r'_1? = h(R_3\|TS_3\|d_l)$ and on success picks a random nonce r_4, present timestamp TS_4 and calculates $R_4 = r_4.G, R'_4 = r_2.P_l, K_{lj} = h(R_3\|R_4\|TS_4)$ and $r'_4 = h(R_4\|TS_4\|K_{lj})$. CS_l now sends the message containing $\{ID_l, R'_4, r'_4, TS_4\}$ to FS_j.
- FS_j receives the message $\{ID_l, R'_4, r'_4, TS_4\}$ from CS_l and checks the message freshness by checking the condition $TS_4 - TS^*_4 \leq \Delta T$. If true, FS_j calculates $R_4 = R_4.d_j^{-1}$ and $K_{jl} = h(R_3\|R_4\|TS_4)$. Further FS_j verifies $r'_4 = h(R_4\|TS_4\|K_{jl})$. If the condition is satisfied FS_j stores K_{jl} for secure future communication.

C. USER REGISTRATION PHASE

This phase of Ali et al. scheme is as follows: The U_i is to access the smart device D_k.

- U_i picks a unique ID_i, a private key $d_i \in Z^*p$, calculates $P_i = d_i.G$ and sends $\{ID_i, P_i\}$ to TA through a secure channel
- TA receives $\{ID_i, P_i\}$ from U_i, and computes $TC_i = h(ID_i\|K)$. TA sends $\{TC_i, \{ID_k|k = 1, 2, - - -n_d\}, \{ID_j, P_j|j = 1, 2, ---, n_f\}\}$ to U_i through secure channel.
- U_i receives $\{TC_i, \{ID_k|k = 1, 2, - - -n_d\}, \{ID_j, P_j|j = 1, 2, - - -, n_f\}\}$ from TA, chooses password PW_i and imprints BIO_i. Further U_i calculates $Gen(BIO_i) = (\sigma_i, \tau_i)$, $d^*_i = d_i \oplus h(ID_i\|PW_i\|\sigma_i)$, $TC^*_i = TC_i \oplus h(ID_i\|\sigma_i)$, $RPB_i = h(ID_i\|TC_i\|PW_i\|\sigma_i)$, $ID^*_i = ID_i \oplus h(d_i\|\sigma_i)$. Finally, MD_i overwrites the information $\{ID_i, d_i, TC_i\}$. The final parameters in MD_i are $\{ID^*_i, TC^*_i, d^*_i, RPB_i, P_i, \{ID_k|k = 1, 2, - - -n_d\}, \{ID_j, P_j|j = 1, 2, - - -, n_f\}, \tau_i, Gen(.)$ $Rep(.), h(.)\}$ where n_d is the number of device identities, and n_f is the number of fog servers registered.

D. LOGIN AND AUTHENTICATION PHASE

Login and authentication are performed by the user U_i to log in through the mobile device MD_i and access D_k. The fog server FS_j mediates to authenticate parties and a mutual session key is established.

- In this phase, first U_i submits ID_i, PW_i, and imprints BIO'_i. Now, MD_i computes $\sigma'_i = Rep(BIO'_i, \tau_i)$, $TC_i = TC^*_i \oplus h(ID_i \| \sigma'_i)$, $d_i = d *_i \oplus h(ID_i \| PW_i \| \sigma'_i)$, $ID_i = ID *_i \oplus h(d_i \| \sigma'_i)$ and $RPB_i = h(ID_i \| TC_i \| PW_i \| \sigma'_i)$.
- MD_i checks the condition $RPB'_i ? = RPB_i$ if true U_i provides ID_j, ID_k, and MD_i fetches the P_j to the corresponding ID_j. Further, MD_i selects a nonce r_i, current timestamp TS_i, and computes $R_i = r_i.G$, $R'_i = r_i.P_j$ and $a_i = TS_i.d_i + r_i$ and $ID'_i = ID_i \oplus h(R_i \| TS_i)$, $E_i = h(R_i \| R'_i \| a_i \| TS_i)$ and $F_i = ID_k \oplus h(R'_i \| R_i \| TS_i)$. MD_i sends $Msg1 = \{ID'_i, R'_i, a_i, F_i, E_i, TS_i\}$ to FS_j via the public channel.
- FS_j receives $Msg1$ and checks the message freshness by checking the condition $TS_i - TS^*_i \leq \Delta T$. If it is true then MD_i computes $R_i = d_j^{-1}.R'_i$ $ID_i = ID'_i \oplus h(R_i \| TS_i)$, FS_j checks the condition $a_i.G = TS_i.P_i + R_i$ and $E_i ? = h(R_i \| R'_i \| a_i \| TS_i)$. If true FS_j picks a nonce r_f, current timestamp TS_f, and computes $K_{uf} = r_f.R_i = (r_i r_f).G$ $P_f = r_f.G$, $ID_k = F_i \oplus h(R'_i \| R_i \| TS_i)$, $d_k = h(d_j \| ID_k \| ID_j)$, $ID^*_i = ID_i \oplus h(d_k \| ID_k \| TS_f)$, $ID'_k = ID_k \oplus h(d_k \| TS_f)$, $G_j = h(d_k \| ID_k \| TS_f) \oplus h(K_{uf} \| h(R_i \| TS_i) \| ID_i)$ and $H_j = h(ID_i \| ID_k \| G_j \| P_f \| TS_f \| d_k)$. Further, FS_j sends $Msg2 = \{ID^*_i, ID'_k, P_f, H_j, G_j, TS_f\}$ to D_k through the open channel.
- D_k receives $Msg2$ from FS_j and the freshness of the message is verified using the condition $TS_f - TS^*_f \leq \Delta T$. If the condition is satisfied, D_k calculates $ID_k = ID'_k \oplus h(d_k \| TS_f)$, $ID_i = ID^*_i \oplus h(d_k \| ID_k \| TS_f)$, and verifies the condition $H_j = h(ID_i \| ID_k \| G_j \| P_f \| TS_f \| d_k)$. If this condition is not satisfied the system terminates the session. Otherwise, D_k selects r_k, timestamp TS_k and computes $I_j = G_j \oplus h(K_{uf} \| h(R_i \| TS_i) \| ID_i)$, $ID^*_k = ID_k \oplus h(ID_i \| TS_k \| I_j)$, $SKki = h(I_j \| r_k \| TS_k)$, $M_k = h(TC_k \| r_k) \oplus h(I_j)$ and $N_k = h(SKki \| P_f \| TS_k)$. At last, D_k sends $Msg3 = \{ID^*_k, M_k, N_k, P_f, TS_k\}$ to U_i through the open channel.
- U_i receives $Msg3$ from D_k and verifies the condition $TS_k - TS^*_k \leq \Delta T$. If true then U_i computes $ID_k = ID^*_k \oplus h(ID_i \| TS_k \| I_j)$, $K_{uf} = r_i.P_f$, $I_j = h(K_{uf} \| h(R_i \| TS_i) \| ID_i)$, $r_k = M_k \oplus h(I_j)$, $SK_{ik} = h(I_j \| r_k \| TS_k)$ and $N'_k = h(SKik \| P_f \| TS_k)$. Further U_i checks $N'_k ? = N_k$. If the condition is true U_i saves SK_{ik}. Else terminates the session.

V. CRYPTANALYSIS OF ALI ET AL. AUTHENTICATION SCHEME

It has been proved in this section that the scheme proposed by Ali et al. is susceptible to revelation attacks and does not ensure user anonymity and session secrecy.

A. KEY REVELATION ATTACK

In the Ali et al. authentication scheme, if the stored parameters of the server $\{ID_l, d_l\}$ are compromised to adversary $\mathcal{A}$, then $\mathcal{A}$ can also get $\{ID_j, P_j\}$ since it is a public parameter. To perform the attack $\mathcal{A}$ performs the following steps:

1) KEY REVELATION ATTACK FOR SMART DEVICE AND FOG SERVER

$\mathcal{A}$ intercepts the communicated parameters $\{ID_k, R'_1, r'_1, TS_1\}$, $\{ID_j, R'_2, r'_2, TS_2\}$ and computes $K_{jk} = h(R_1 \| R_2 \| TS_2)$ where $R_1 = R'_1.d_j^{-1}$. Here $d_j = h(ID_j \| d_l)$ where ID_j can be available from intercepted parameters. R_2 can be obtained from the equation $R_2 = R'_2.d_k^{-1}$ where D_k can be computed from $d_k = h(d_j \| ID_k \| ID_j)$. Here ID_j can be obtained from intercepted parameters.

2) KEY REVELATION ATTACK FOR FOG SERVER AND CLOUD SERVER

$\mathcal{A}$ intercepts the communicated parameters $\{ID_j, R'_2, r'_2, TS_2\}$, $\{ID_l, R'_4, r'_4, TS_4\}$ and computes $K_{jl} = h(R_3 \| R_4 \| TS_4)$ where $R_3 = R'_3.d_l^{-1}$. Here d_l can be obtained from stored parameters and R'_3 and TS_4 is an intercepted parameter from the public channel. R_4 can be obtained from the equation $R_4 = R'_4.d_j^{-1}$ where $d_j = h(ID_j \| d_l)$. where d_l can be obtained from stolen parameters ID_j from intercepted parameters.

From V.A.1 and V.A.2 it is clear that the keys K_{jk} and K_{jl} are not secure for device communication.

B. USER ANONYMITY

In Ali et al. scheme adversary can obtain the user identity ID_i as follows: Suppose $\mathcal{A}$ steals the device and succeeds in obtaining the stored parameters $\{ID_k | k = 1, 2, - - -n_d\}, \{ID_j, P_j | j = 1, 2, - - -, n_f\}, \tau_i, Gen(.)Rep(.), h(.)\}$. Also, $\mathcal{A}$ can intercept the communicated messages $\{ID'_i, R'_i, a_i, F_i, E_i, TS_i\}$, $\{ID^*_i, ID'_k, P_f, H_j, G_j, TS_f\}$, and $\{ID^*_k, M_k, N_k, P_f, TS_k\}$. Now consider the equation $ID_i = ID'_i \oplus h(R_i \| TS_i)$ where $R_i = d_j^{-1}.R'_i$. We know that $d_j = h(ID_j \| d_l)$. In section V-A, we illustrated how $\mathcal{A}$ could obtain ID_j and d_l. $\mathcal{A}$ can obtain ID'_i, and TS_i from the intercepted messages. Hence $\mathcal{A}$ computes the user identity ID_i through $ID_i = ID'_i \oplus h(R_i \| TS_i)$. Therefore Ali et al. scheme doesn't provide user anonymity.

C. INSECURE SESSION SECRET KEY/NO PERFECT FORWARD SECRECY

Assume that $\mathcal{A}$ stole the device and obtained the stored parameters $\{ID_k | k = 1, 2, - - -n_d\}, \{ID_j, P_j | j = 1, 2, - - -, n_f\}, \tau_i, Gen(.)Rep(.), h(.)\}$. Also, $\mathcal{A}$ intercepted communicated messages $\{ID'_i, R'_i, a_i, F_i, E_i, TS_i\}$, $\{ID^*_i, ID'_k, P_f, H_j, G_j, TS_f\}$, and $\{ID^*_k, M_k, N_k, P_f, TS_k\}$. Now consider the equation $SKki = h(I_j \| r_k \| TS_k)$, where $I_j = G_j \oplus h(K_{uf} \| h(R_i \| TS_i) \| ID_i)$ and $r_k = M_k \oplus h(I_j)$. Here $\mathcal{A}$ can get G_j from intercepted messages. d_k can be computed by the equation $d_k = h(d_j \| ID_k \| ID_j)$ where ID_k, and ID_j can

TABLE 2. Notations and descriptions.

Notations	Descriptions
TA	Trusted Authority
CS_l	Cloud Server
FS_j	Fog Node/Server
D_k	Smart Device
x	System master key
ID_c	Identity of cloud server
ID_f	Identity of fog node
ID_s	Identity of Smart device
ID_i	Identity of user
BIO_i	Biometric character of user
E_p	Elliptic Curve of order p
F_p	Finite field
$h_0(.), h_1(.), h_2(.)$	One-way hash functions
G	Point on Elliptic curve E_p
G_{pub}	Public key computed using master key
n_c, n_f, n_s	System generated private keys
$C_{pub}, F_{pub}, S_{pub}$	System computed private keys
RT_c, RT_f, RT_s, RT_i	Registration Timestamps
$f(x,y,z), g(x,y,z)$	Trivariate Polynomial
TS_1, TS_2, TS_3, TS_4	Timestamp used during device key exchange
T_1, T_2, T_3, T_4	Timestamp used during user authentication
r_1, r_2, r_3, r_4	Random numbers used during device key exchange
W_1, W_2, W_3	Random numbers used during user authentication
PW_i	Password of U_i
$SK_{sfc}, SK_{fcs}, SK_{cfs}$	Shared session keys
$\oplus$	Bitwise XOR operator
$\|$	Concatenation operator

be obtained from stolen device parameters and d_j can be obtained by $d_j = h(ID_j \| d_l)$. In section V-A, we explained how $\mathcal{A}$ could obtain ID_j and d_l. Now $\mathcal{A}$ has all the parameters required to compute $SKki$ i.e. $SKki = h(I_j \| r_k \| TS_k)$. Therefore Ali et al. scheme does not have perfect forward secrecy.

VI. PROPOSED AUTHENTICATION SCHEME

The authors propose an efficient and secure authentication scheme using key agreement and management for the cloud-fog-device framework to overcome the security issues identified in Ali et al. scheme. The proposed scheme contains six phases as follows: (1) the Pre-deployment phase, (2) the Secure Key Exchange phase (3) the User Registration phase (4) the Login and Authentication phase (5) the Secure Password change phase, and (6) Device addition phase. The notations used throughout the scheme and its description are presented in Table 2. The procedure to implement each step is explained below.

A. PHASE 1–PRE-DEPLOYMENT PHASE

The pre-deployment phase of a cloud server, fog node, and mobile/smart device registration involves generating parameters. Initially, a trusted authority (TA) selects the master key (x) and calculates the public key by $G_{pub} = x.G$. The TA uses a large prime number 'p', an elliptic curve E_p and two symmetric 't' degree trivariate polynomials ($f(x, y, z)$, $g(x, y, z)$) over the finite field F_p. In addition, the TA selects $h_0(.) \rightarrow Z{\cdot}h_1(.) \rightarrow Z{\cdot}h_2(.) \rightarrow Z^*$ as one-way hash functions, a point on the elliptic curve (G) of order 'n'. Finally, TA publishes the parameters $\{E_p, F_p, G, h_0(.), h_1(.), h_2(.), G_{pub}, f(x, y, z)g(x, y, z)\}$.

1) CLOUD SERVER REGISTRATION PHASE

TA generates ID_c, RT_c, n_c and computes $TID_c = h_0(ID_c \| x \| n_c)$ and $CID_c = h_0(TID_c \| RT_c \| n_c)$. Further, TA computes server public key $C_pub = n_c.G$ and sends $\{TID_c, CID_c, RT_c, n_c, f(x, y, z), G, h_0(.), h_1(.), h_2(.), G_{pub}\}$ to the cloud server through a secure channel. Finally, TA publicizes the server public key C_{pub}.

2) FOG NODE REGISTRATION PHASE

TA generates ID_f, RT_f, n_f and computes $TID_f = h_0(ID_f \| x \| n_f)$ and $CID_f = h_0(TID_f \| RT_f \| n_f)$. Further, TA computes node public key $F_pub = n_f.G$ and sends $\{TID_f,\ CID_f,\ RT_f,\ n_f,\ f(x, y, z),\ g(x, y, z),\ G,\ h_0(.),\ h_1(.),\ h_2(.), G_{pub}\}$ to the fog node through a secure channel. Finally, TA publicizes node public key F_{pub}.

3) SMART DEVICE REGISTRATION PHASE

TA generates ID_s, RT_s, n_s and computes $TID_s = h_0(ID_s \| x \| n_s)$ and $CID_s = h_0(TID_s \| RT_s \| n_s)$. Further, TA computes device public key $S_pub = n_s.G$ and sends $\{TID_s, CID_s, RT_s, n_s, g(x, y, z), G, h_0(.), h_1(.), h_2(.), G_{pub}\}$ to the device through a secure channel. Finally, TA publicizes the device's public key S_{pub}.

B. PHASE 2–SECURE KEY EXCHANGE PHASE

This phase performs the secure key exchange amongst smart devices, fog servers, and cloud servers over an insecure channel.

1) KEY EXCHANGE BETWEEN SMART DEVICES AND FOG NODES

The Key Exchange between smart devices and fog nodes is presented in Table 3. The detailed steps are as follows:

- The smart device generates r_1 and TS_1 and computes $G_1 = r_1.F_{pub}$, $G_2 = r_1.G$, $C_s = h_0(CID_s \| n_s) \oplus G_1$ $RID_s = h_0(G_1 \| TS_1) \oplus n_s$, $M_1 = h_0(RID_s \| n_s \| G_1 \| TS_1)$. Device sends $\{CID_s, RID, TS_1, M_1\}$
- Fog node FS_j receives the message $\{CID_s, RID, TS_1, M_1\}$ and verifies the freshness of the message. FS_j generates TS_1^* and verifies if $TS_1 - TS_1^* \leq \Delta T$. If the condition is false then the fog node drops the session, else FS_j computes $G_1' = G_2.n_f$, $C_s' = h_0(CID_s \| n_s \| G_1) \oplus G_1$, $n_s = RID_s \oplus h_0(G_1 \| TS_1)$ and $M_1' = h_0(RID_s \| n_s \| G_1 \| TS_1)$. Further, FS_j verifies the condition $M_1' = M_1$. If the condition is not satisfied, the system drops the session. Else fog server generates r_2 and TS_2 and computes $G_3 = r_2.S_{pub}$, $G_4 = r_2.G$, $g(CID_f, CID_s, 1)$, $g(CID_f, CID_s, r_2)$, $FID = r_2 \oplus h_0(g(CID_f, CID_s, 1) \| G_3 \| G_1' \| TS_2)$, $K_{fs} = h_0(g(CID_f, CID_s, r_2) \| G_3 \| G_1' \| r_2)$ and $M_2 = h_0(K_{fs} \| G_3 \| G_1')$. Finally, FS_j sends $\{M_2, CID_f, FID, G_4, TS_2\}$ to the smart device.
- Smart device D_k receives $\{M_2, CID_f, FID, G_4, TS_2\}$ from FS_j and verifies the freshness of the message. To do that D_k generates TS_2^* and checks the condition $TS_2 - TS_2^* \leq \Delta T$. If the condition is true then device computes $G_3' = G_4.n_s$, $g(CID_s, CID_f, 1)$, $r_2' = FID \oplus h_0(g(CID_f, CID_s, 1) \| G_3' \| G_1 \| TS_2)$, $g(CID_s, CID_f, r_2')$,

TABLE 3. Key Exchange between smart devices and fog servers.

smart device	fog server
generates r_1 and TS_1 Computes $G_1 = r_1.F_pub$, $G_2 = r_1.G$, $C_s = h_0(CID_s \Vert n_s) \oplus G_1$ $RID_s = h_0(G_1 \Vert TS_1) \oplus n_s$, $M_1 = h_0(RID_s \Vert n_s \Vert G_1 \Vert TS_1)$. Sends $\{CID_s, RID, TS_1, M_1\}$ $\xrightarrow{\{CID_s, RID, TS_1, M_1\}}$	
	Receives message, generates TS_1^* and checks $TS_1 - TS_1^* \leq \Delta T$ computes $G_1' = G_2.n_f$, $C_s' = h_0(CID_s \Vert n_s \Vert G_1) \oplus G_1$, $n_s = RID_s \oplus h_0(G_1 \Vert TS_1)$ and $M_1' = h_0(RID_s \Vert n_s \Vert G_1 \Vert TS_1)$. verifies the condition $M_1' = M_1$. Generates r_2 and TS_2 and Computes $G_3 = r_2.S_pub$, $G_4 = r_2.G$ $g(CID_f, CID_s, 1), g(CID_f, CID_s, r_2)$ $FID = r_2 \oplus h_0(g(CID_f, CID_s, 1) \Vert G_3 \Vert G_1' \Vert TS_2)$ $K_{fs} = h_0(g(CID_f, CID_s, r_2) \Vert G_3 \Vert G_1' \Vert r_2)$ $M_2 = h_0(K_{fs} \Vert G_3 \Vert G_1')$ sends $\{M_2, CID_f, FID, G_4, TS_2\}$ $\xleftarrow{\{M_2, CID_f, FID, G_4, TS_2\}}$
Receives message, generates TS_2^* and Checks the condition $TS_2 - TS_2^* \leq \Delta T$ Computes $G_3' = G_4.n_s, g(CID_s, CID_f, 1)$ $r_2' = FID \oplus h_0(g(CID_f, CID_s, 1) \Vert G_3' \Vert G_1 \Vert TS_2)$ $g(CID_s, CID_f, r_2')$ $K_{sf} = h_0(g(CID_f, CID_s, r_2') \Vert G_3' \Vert G_1 \Vert r_2')$ $M_2' = h_0(K_{sf} \Vert G_3' \Vert G_1)$ Compares $M_2' = M_2$ Stores the key K_{fs}	Stores the key K_{sf}

TABLE 4. Key Exchange between fog servers and cloud server.

fog server	Cloud server
generates r_3 and TS_3 computes $G_5 = r_3.C_pub$, $G_6 = r_3.G$, $C_f = h_0(CID_f \Vert n_f) \oplus G_5$ $RID_f = h_0(G_5 \Vert TS_3) \oplus n_f$, $M_3 = h_0(RID_f \Vert n_f \Vert G_5 \Vert TS_3)$. sends $\{CID_f, RID_f, TS_3, M_3\}$. $\xrightarrow{\{CID_f, RID_f, TS_3, M_3\}}$	
	Receives message, generates TS_3^* checks the condition $TS_3 - TS_3^* \leq \Delta T$. computes $G_5' = G_6.n_c, n_f = RID_f \oplus h_0(G_5' \Vert TS_3)$, $C_f' = h_0(CID_f \Vert n_f) \oplus G_5'$ and $M_3' = h_0(RID_f \Vert n_f \Vert G_5' \Vert TS_3)$. verifies the condition $M_3' = M_3$. Generates r_4 and TS_4 and Computes $G_7 = r_4.F_pub$, $G_8 = r_4.G$ $f(CID_c, CID_f, 1), f(CID_c, CID_f, r_4)$ $CSID_i = r_4 \oplus h_0(f(CID_c, CID_f, 1) \Vert G_6 \Vert G_4' \Vert TS_4)$, $K_{cf} = h_0(f(CID_c, CID_f, r_4) \Vert G_7 \Vert G_5' \Vert r_4)$ $M_4 = h_0(K_{cf} \Vert G_7 \Vert G_5')$ sends $\{M_4, CID_c, CSID_i, G_8, TS_4\}$ $\xleftarrow{\{M_4, CID_c, CSID_i, G_8, TS_4\}}$
Receives message, generates TS_4^* and Checks the condition $TS_4 - TS_4^* \leq \Delta T$ Computes $G_7' = G_8.n_f, f(CID_f, CID_c, 1)$ $r_4' = CSID \oplus h_0(f(CID_f, CID_c, 1) \Vert G_7' \Vert G_5 \Vert TS_4)$ $f(CID_f, CID_c, r_4')$ $K_{fc} = h_0(f(CID_f, CID_c, r_4') \Vert G_7' \Vert G_5 \Vert r_4')$ $M_4' = h_0(K_{fc} \Vert G_7' \Vert G_5)$ Compares $M_4' = M_4$ Stores the key K_{fc}	Stores the key K_{cf}

$K_{sf} = h_0(g(CID_f, CID_s, r_2') \| G_3' \| G_1 \| r_2')$ and $M_2' = h_0(K_{sf} \| G_3' \| G_1)$. Finally, the device compares $M_2' = M_2$. If the condition is false, the device terminates the session. Else, FS_j and D_k store the keys K_{fs} and K_{sf} on their respective sides.

2) KEY EXCHANGE BETWEEN FOG NODES AND CLOUD SERVERS

The Key Exchange between fog nodes and cloud servers is presented in Table 4. The steps are illustrated as follows:

TABLE 5. User registration phase.

smart device/user	Trusted Authority
Choose ID_i. Generates random nonce b_i computes $UID_i = h_1(ID_i \| b_i)$. sends $\{UID_i\}$ to TA. $\xrightarrow{\{UID_i\}}$	
	Receives message, generates e, and RT_i computes $m_i = h_1(x \| e).G$, $H_n = h_1(UID_i \| m_i \| RT_i)$, $V_i = h_1(x \| e) \oplus h_i(UID_i)$. Sends $\{V_i, RT_i\}$ to user $\xleftarrow{\{V_i, RT_i\}}$
Receives and selects PW_i and imprints BIO_i. computes $Gen(BIO_i) = (\sigma_i, \tau_i)$, $h(x \| e)' = V_i \oplus h_i(UID_i)$, $m_i' = h_1(x \| e)'.G$, $H_n = h_1(UID_i \| m_i' \| RT_i)$, $RPW = h_1(PW_i \| \sigma_i \| m_i')$, $B_i = h_1(H_n' \| RPW \| b_i)$, and $R_i = b_i \oplus h_1(ID_i \| PW_i \| \sigma_i)$ stores $\{B_i, R_i, Gen(.), Rep(.), \tau_i, h_1(.), h_2(.), V_i, RT_i\}$	

- The fog node generates r_3 and TS_3 and computes $G_5 = r_3.C_{pub}$, $G_6 = r_3.G$, $C_f = h_0(CID_f \| n_f) \oplus G_5$ $RID_f = h_0(G_5 \| TS_3) \oplus n_f$, $M_3 = h_0(RID_f \| n_f \| G_5 \| TS_3)$. The fog node sends $\{CID_f, RID_f, TS_3, M_3\}$ to the cloud server.
- Cloud server CS_l receives the message $\{CID_f, RID_f, TS_3, M_3\}$ and verifies the freshness of the received message. CS_l generates TS_3^* and checks the condition $TS_3 - TS_3^* \leq \Delta T$. If the condition is false then CS_l drops the session. Else CS_l computes $G_5' = G_6.n_c$, $n_f = RID_f \oplus h_0(G_5' \| TS_3)$, $C_f' = h_0(CID_f \| n_f) \oplus G_5'$ and $M_3' = h_0(RID_f \| n_f \| G_5' \| TS_3)$. Further CS_l verifies the condition $M_3' = M_3$. If the condition is not satisfied, the system drops the session. Else CS_l generates r_4 and TS_4 and computes $G_7 = r_4.F_{pub}$, $G_8 = r_4.G, f(CID_c, CID_f, 1), f(CID_c, CID_f, r_4)$, $CSID_i = r_4 \oplus h_0(f(CID_c, CID_f, 1) \| G_6 \| G_4' \| TS_4)$, $K_{cf} = h_0(f(CID_c, CID_f, r_4) \| G_7 \| G_5' \| r_4)$ and $M_4 = h_0(K_{cf} \| G_7 \| G_5')$. Finally CS_l sends $\{M_4, CID_c, CSID_i, G_8, TS_4\}$ to the fog node.
- Fog node receives $\{M_4, CID_c, CSID_i, G_8, TS_4\}$ from CS_l and validates the recency of the message. To do that FS_j generates TS_4^* and checks the condition $TS_4 - TS_4^* \leq \Delta T$. If the condition is true then device computes $G_7' = G_8.n_f, f(CID_f, CID_c, 1)$, $r_4' = CSID \oplus h_0(f(CID_f, CID_c, 1) \| G_7' \| G_5 \| TS_4)$, $f(CID_f, CID_c, r_4')$, $K_{fc} = h_0(f(CID_f, CID_c, r_4') \| G_7' \| G_5 \| r_4')$ and $M_4' = h_0(K_{fc} \| G_7' \| G_5)$. Finally device compares $M_4' = M_4$. If the condition is false, device terminates the session. Else, FS_j and CS_l stores the keys K_{cf} and K_{fc} on their respective sides.

C. PHASE 3–USER REGISTRATION PHASE

Table 5 presents the new user registration. If the user U_i registers for the first time, he/she follows the procedure mentioned below:

- The system performs the user registration phase through the secure channel. In the beginning, U_i chooses the identity ID_i. The further system generates random nonce b_i and computes $UID_i = h_1(ID_i \| b_i)$. U_i system sends a registration request $\{UID_i\}$ to TA.
- TA receives $\{UID_i\}$ from the U_i and generates a random nonce e, a registration timestamp RT_i. Further TA calculates the following parameters: $m_i = h_1(x \| e).G$, $H_n = h_1(UID_i \| m_i \| RT_i)$, and $V_i = h_1(x \| e) \oplus h_i(UID_i)$. Further, TA communicates the computed parameters $\{V_i, RT_i\}$ to the user.
- U_i receives $\{V_i, RT_i\}$ from TAz, selects the password PW_i and imprints the biometric BIO_i. Further, U_i computes $Gen(BIO_i) = (\sigma_i, \tau_i)$, $h(x \| e)' = V_i \oplus h_i(UID_i)$, $m_i' = h_1(x \| e)'.G$, $H_n = h_1(UID_i \| m_i' \| RT_i)$, $RPW = h_1(PW_i \| \sigma_i \| m_i')$, $B_i = h_1(H_n' \| RPW \| b_i)$, and $R_i = b_i \oplus h_1(ID_i \| PW_i \| \sigma_i)$ Further U_i stores $\{B_i, R_i, Gen(.), Rep(.), \tau_i, h_1(.), h_2(.), V_i, RT_i\}$ x into the smart card which is installed in the device D_k.

D. PHASE 4–LOGIN AND AUTHENTICATION PHASE

This phase performs between User U_i, smart device D_k, fog node FS_j and the cloud server CS_l. If U_i has to access a service from CS_l, the user logs in to the smart device, and D_k verifies the user and authenticates from FS_j. Further, the device and fog node will be authenticated by CS_l. Table 6 presents the login and authentication phase of the proposed scheme. Also, it is explained below as a series of steps.

- U_i input ID_i, PW_i, and imprints BIO_i. Since the smart card is installed in the device D_k, we consider U_i system as a combination of smart card and D_k.
- U_i system computes $\sigma{`}_i = Rep(BIO_i', \tau_i)$, $b_i' = R_i \oplus h_1(ID_i \| PW_i \| \sigma_i')$, $UID_i' = h_1(ID_i \| b_i')$, $h(x \| e)' = V_i \oplus h_i(UID_i')$, $m_i' = h_1(x \| e)'.G$, $H_n' = h_1(UID_i \| m_i' \| RT_i)$, $RPW' = h_1(PW_i \| \sigma_i' \| m_i')$, and $B_i' = h_1(H_n' \| RPW \| b_i')$. Further U_i verifies whether $B_i' = B_i$ or not. If the condition is true PW_i and BIO_i is true. Else U_i drops the session.
- After PW_i and BIO_i, if entered U_i credentials are correct user system generates the current timestamp T_1 and a

TABLE 6. Proposed scheme's login and authentication phase.

User and Smart Device	Fog Node	Cloud Server
Input ID_i, PW_i and imprints BIO_i computes $\sigma'_i = Rep(BIO'_i, \tau_i)$, $b'_i = R_i \oplus h_1(ID_i \Vert PW_i \Vert \sigma'_i)$, $UID'_i = h_1(ID_i \Vert b'_i)$, $h(x \Vert e)' = V_i \oplus h_i(UID'_i)$, $m'_i = h_1(x \Vert e)'.G$, $H'_n = h_1(UID_i \Vert m'_i \Vert RT_i)$, $RPW' = h_1(PW_i \Vert \sigma'_i \Vert m'_i)$, $B'_i = h_1(H'_n \Vert RPW \Vert b'_i)$. verifies whether $B'_i = B_i$ generates T_1 and w_1. computes $RV_1 = w_1.F_{pub}, RV_2 = w_1.G$, $C_{sm} = h_2(CID_s \Vert T_1 \Vert RV_1) \oplus h(x \Vert e)'$, $DUID_i = h_2(CID_s \Vert RV_1 \Vert T_1 \Vert m_i)$. Sends $\{CID_s, RV_2, C_{sm}, T_1\}$. $\xrightarrow{\{CID_s, RV_2, C_{sm}, T_1\}}$		
	Checks $T_1 - T_1^* \leq \Delta T$ Computes $RV'_1 = RV_2.n_f$ $h(x \Vert e)' = C_{sm} \oplus h_2(CID_s \Vert T_1 \Vert RV'_1)$ $m'_i = h_1(x \Vert e)'.G$ $DUID'_i = h_2(CID_s \Vert RV'_1 \Vert T_1 \Vert m'_i)$ Generates T_2, and w_2 computes $FV_1 = w_2.C_{pub}$, $FV_2 = w_2.G, c_f = h_2(CID_f \Vert T_2 \Vert FV_1) \oplus h(x \Vert e)'$, $F_c = h_2(h(x \Vert e)' \Vert C_f \Vert FV_1) \oplus RV'_1$ $FUID_i = h_2(DUID'_i \Vert m'_i \Vert FV_1 \Vert RV'_1 \Vert T_1 \Vert T_2)$. Sends $\{CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2\}$. $\xrightarrow{CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2}$	
		Receives and checks $T_2 - T_2^* \leq \Delta T$. Computes $FV_1 = FV_2.n_c$, $h(x \Vert e)' = c_f \oplus h_2(CID_f \Vert T_2 \Vert FV'_1)$, $m'_i = h_1(x \Vert e)'.G$, $RV'_1 = F_c \oplus h_2(h(x \Vert e)' \Vert C_f \Vert FV'_1)$, $DUID_i = h_2(CID_s \Vert RV'_1 \Vert T_1 \Vert m'_i)$, $FUID'_i = h_2(DUID'_i \Vert m'_i \Vert FV'_1 \Vert RV'_1 \Vert T_1 \Vert T_2)$. verifies $FUID'_i = FUID_i$ Generates T_3, and w_3 $CV_1 = w_3.F_{pub}, CV_2 = w_2.G$, $SK_{cfs} = h_2(m'_i \Vert RV'_1 \Vert FV'_1 \Vert CV_1)$ $CSUID_i = h_2(SK_{cfs} \Vert m'_i \Vert T_3 \Vert CV_1)$. Sends $\{CV_2, T_3\}$ $\xleftarrow{\{CV_2, T_3\}}$
	Receives and verifies $T_3 - T_3^* \leq \Delta T$ Computes $CV_1 = CV_2.n_f$, $SK_{fcs} = h_2(m'_i \Vert RV'_1 \Vert FV_1 \Vert CV'_1)$, $CSUID'_i = h_2(SK_{fcs} \Vert m'_i \Vert T_3 \Vert CV'_1)$, $FCSUID_i = h_2(CSUID'_i \Vert T_4 \Vert m'_i \Vert CV'_1)$, $F_{sn} = h_2(m'_i \Vert T_4) \oplus FV_1$ $F_{sm} = h_2(F_{sn} \Vert m'_i \Vert T_4) \oplus CV'_1$. Sends $\{F_{sm}, T_4, FCSUID_i, T_3, CV_2\}$ $\xleftarrow{\{F_{sm}, T_4, FCSUID_i, T_3, CV_2\}}$	
Receives and Checks $T_4 - T_4^* \leq \Delta T$ Computes $FV'_1 = F_{sn} \oplus h_2(m'_i \Vert T_4)$, $CV'_1 = F_{sm} \oplus h_2(F_{sn} \Vert m'_i \Vert T_4)$, $SK_{sfc} = h_2(m'_i \Vert RV_1 \Vert FV'_1 \Vert CV'_1)$, $CSUID'_i = h_2(SK_{sfc} \Vert m'_i \Vert T_3 \Vert CV'_1)$ $FCSUID'_i = h_2(CSUID'_i \Vert T_4 \Vert m'_i \Vert CV'_1)$.		
User Session key $SK_{sfc} = h_2(m'_i \Vert RV_1 \Vert FV'_1 \Vert CV'_1)$	Fog Server session key $SK_{fcs} = h_2(m'_i \Vert RV'_1 \Vert FV_1 \Vert CV'_1)$	Cloud Server session key $SK_{cfs} = h_2(m'_i \Vert RV'_1 \Vert FV'_1 \Vert CV_1)$

random nonce w_1. Further, the system computes $RV_1 = w_1.F_{pub}$, $RV_2 = w_1.G$, $C_{sm} = h_2(CID_s \| T_1 \| RV_1) \oplus h(x \| e)'$, and $DUID_i = h_2(CID_s \| RV_1 \| T_1 \| m_i)$. U_i sends $\{CID_s, RV_2, C_{sm}, T_1\}$ to the fog server FS_j.

- FS_j receives $\{CID_s, RV_2, C_{sm}, T_1\}$ from U_i, and checks the message's validity. FS_j uses the current time T_1^* and checks for $T_1 - T_1^* \leq \Delta T$. If the condition holds, FS_j will receive a fresh message, else FS_j rejects the massage and drops the session.
- Once FS_j receives the fresh request message, it computes the following. $RV'_1 = RV_2.n_f$, $h(x \| e)' = C_{sm} \oplus h_2(CID_s \| T_1 \| RV'_1)$, $m'_i = h_1(x \| e)'.G$, $DUID'_i = h_2(CID_s \| RV'_1 \| T_1 \| m'_i)$
- Further, FS_j generates a timestamp T_2, a random nonce w_2, and computes $FV_1 = w_2.C_{pub}$, $FV_2 = w_2.G$, $c_f = h_2(CID_f \| T_2 \| FV_1) \oplus h(x \| e)'$, $F_c = h_2(h(x \| e)' \| C_f \| FV_1) \oplus RV'_1$ and $FUID_i = h_2(DUID'_i \| m'_i \| FV_1 \| RV'_1 \| T_1 \| T_2)$. Finally, fog node sends the authentication request message $\{CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2\}$ to the cloud server CS_l.
- The authentication request message $\{CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2\}$ is received by the cloud server. It verifies the validity of the timestamp to ensure the freshness of the message. CS_l uses the current time T_2^* and checks for $T_2 - T_2^* \leq \Delta T$. If the condition is satisfied, CS_l receives a fresh message, else CS_l rejects the message and the session is dropped.
- Once CS_l receives the fresh request message, computes the following: $FV_1 = FV_2.n_c$, $h(x \| e)' = c_f \oplus h_2(CID_f \| T_2 \| FV'_1)$, $m'_i = h_1(x \| e)'.G$, $RV'_1 = F_c \oplus h_2(h(x \| e)' \| C_f \| FV'_1)$, $DUID_i = h_2(CID_s \| RV'_1 \| T_1 \| m'_i)$, and $FUID'_i = h_2(DUID'_i \| m'_i \| FV'_1 \| RV'_1 \| T_1 \| T_2)$. Further, CS_l verifies whether $FUID'_i = FUID_i$ or not. If the condition is true CS_l starts the mutual authentication process.
- In mutual authentication, CS_l generates a timestamp T_3, a random nonce w_3, and computes $CV_1 = w_3.F_{pub}$, $CV_2 = w_2.G$, $SK_{cfs} = h_2(m'_i \| RV'_1 \| FV'_1 \| CV_1)$ and $CSUID_i = h_2(SK_{cfs} \| m'_i \| T_3 \| CV_1)$. Further, CS_l sends $\{CV_2, T_3\}$ to the fog node for mutual authentication.
- On the other side, FS_j receives $\{CV_2, T_3\}$ from CS_l, and checks the freshness of the message using the condition $T_3 - T_3^* \leq \Delta T$. If the condition is true then FS_j computes $CV_1 = CV_2.n_f$, $SK_{fcs} = h_2(m'_i \| RV'_1 \| FV_1 \| CV'_1)$, $CSUID'_i = h_2(SK_{fcs} \| m'_i \| T_3 \| CV'_1)$, $FCSUID_i = h_2(CSUID'_i \| T_4 \| m'_i \| CV'_1)$, $F_{sn} = h_2(m'_i \| T_4) \oplus FV_1$ and

$F_{sm} = h_2(F_{sn}\|m'_i\|T_4) \oplus CV'_1$. Finally fog server sends $\{F_{sm}, F_{sn}, T_4, FCSUID_i, T_3, CV_2\}$ to U_i.

- U_i receives $\{F_{sm}, F_{sn}, T_4, FCSUID_i, T_3, CV_2\}$, verifies the condition $T_4 - T_4^* \leq \Delta T$ and computes $FV'_1 = F_{sn} \oplus h_2(m'_i\|T_4)$, $CV'_1 = F_{sm} \oplus h_2(F_{sn}\|m'_i\|T_4)$, $SK_{sfc} = h_2(m'_i\|RV_1\|FV'_1\|CV'_1)$, $CSUID'_i = h_2(SK_{sfc}\|m'_i\|T_3\|CV'_1)$ and $FCSUID'_i = h_2(CSUID'_i\|T_4\|m'_i\|CV'_1)$. Finally, U_i verifies the condition $FCSUID'_i = FCSUID_i$. If the condition is true the system completes mutual authentication successfully. Otherwise, if the freshness verification fails, U_i the message is rejected and the session is terminated.
- Once mutual authentication is successfully completed, the shared session keys are used for further communication between the parties. Specifically, the session key for the smart device user is $SK_{sfc} = h_2(m'_i\|RV_1\|FV'_1\|CV'_1)$, and for fog server, is $SK_{fcs} = h_2(m'_i\|RV'_1\|FV_1\|CV'_1)$ and for cloud server is $SK_{cfs} = h_2(m'_i\|RV'_1\|FV'_1\|CV_1)$.

E. PHASE 5–SECURE PASSWORD CHANGE PHASE

The user can change the password PW_i to PW^{new}. as per the procedure mentioned below:

- U_i input ID_i, PW_i, and imprints BIO_i. Since the smart card is installed in the device D_k, we consider U_i system as a combination of smart card and D_k.
- U_i system computes $\sigma`_i = Rep(BIO'_i, \tau_i)$, $b'_i = R_i \oplus h_1(ID_i\|PW_i\|\sigma'_i)$, $UID'_i = h_1(ID_i\|b'_i)$, $h(x\|e)' = V_i \oplus h_i(UID'_i)$, $m'_i = h_1(x\|e)'.G$, $H'_n = h_1(UID_i\|m'_i\|RT_i)$, $RPW' = h_1(PW_i\|\sigma'_i\|m'_i)$, and $B'_i = h_1(H'_n\|RPW\|b'_i)$. Further, U_i verifies whether $B'_i = B_i$ or not. If the condition is true PW_i and BIO_i are true.
- After verification of the password, U_i selects the new password PW_i^{new} and computes $RPW^{new} = h_1(PW_i^{new}\|\sigma'_i\|m'_i)$, $B_i^{new} = h_1(H'_n\|RPW^{new}\|b'_i)$ and $R_i^{new} = b'_i \oplus h_1(ID_i\|PW_i^{new}\|\sigma_i)$
- Further, U_i replaces $\{B_i^{new}, R_i^{new}\}$ with old $\{B_i, R_i\}$ and store into the smart card installed in the device D_k.

F. PHASE 6 - DEVICE ADDITION PHASE

Performing the device addition phase is necessary to avoid service interruption because of device failure or if an adversary captures the device. To add a new device, the following steps are followed:

TA generates ID_s^{new}, RT_s^{new}, n_s^{new} and computes $TID_s^{new} = h_0(ID_s^{new}\|x\|n_s^{new})$ and $CID_s^{new} = h_0(TID_s^{new}\|RT_s^{new}\|n_s^{new})$ Further, TA computes device public key $S_{pub}^{new} = n_s^{new}.G$ and sends $\{TID_s^{new}, CID_s^{new}, RT_s^{new}, n_s^{new}, g(x, y, z), G, h_0(.), h_1(.), h_2(.), G_{pub}\}$ to the device. Finally, TA publicizes the device's public key S_{pub}^{new}.

VII. CRYPTANALYSIS OF PROPOSED AUTHENTICATION SCHEME

This section discusses a detailed cryptanalysis of the proposed scheme, firstly an informal security analysis is performed considering nine propositions. A formal security analysis was performed using the ROR model and the computational problem. The security proof is discussed followed by a formal security verification using the Scyther simulation.

A. INFORMAL SECURITY ANALYSIS

An informal security analysis was performed to check for the security requirements identified for the proposed authentication scheme.

1) PROPOSITION 1–THE SCHEME PREVENTS EPHEMERAL SECRET KEY LEAKAGE

Proof: The threat model for the authentication scheme requires that the secret key is not revealed to the legitimate user or the adversary. In our scheme, the secret key is not communicated in plain text. When TA receives a request for registration, it generates a random nonce e and calculates $m_i = h_1(x\|e).G$. Additionally, the random nonce e is not stored in any of the entities, namely U_i, D_k, FS_j, or CS_l. Therefore, the proposed scheme guarantees the security of the secret key.

2) PROPOSITION 2–THE SCHEME PROVIDES PROTECTION FROM REPLAY ATTACKS

Proof: To be secure from replay attacks, the cloud server must verify the freshness of the login request message before verification. This is achieved by using a timestamp to verify the validity of the message in our scheme. Assume that adversary $\mathcal{A}$ intercepted previously successful login request message $\{CID_s, RV_2, C_{sm}, T_1\}$ and resend the same. Upon receiving the message, the fog node FS_j first verifies its freshness. FS_j obtains the current time T_1^* and checks whether the condition $T_1 - T_1^* \leq \Delta T$ holds. Since the FS_j current timestamp T_1^* is different from the previous, the condition fails, and FS_j rejects the message and drops the session. This ensures that our scheme is secure from replay attacks.

3) PROPOSITION 3–THE SCHEME ENSURES SECURITY FOR MAN-IN-THE-MIDDLE (MITM) ATTACKS.

Proof: To perform a man-in-the-middle attack, adversary $\mathcal{A}$ must intercept the communicated login request message and modify it. Assume that $\mathcal{A}$ intercepted the login request message $\{CID_s, RV_2, C_{sm}, T_1\}$ and tries to modify using the equations $RV_1 = w_1.F_{pub}$, $RV_2 = w_1.G$, $C_{sm} = h_2(CID_s\|T_1\|RV_1) \oplus h(x\|e)'$, and $DUID_i = h_2(CID_s\|RV_1\|T_1\|m_i)$. $\mathcal{A}$ can generate random nonce w'_1 and current timestamp T_1 and calculate RV'_1 and RV'_2. However, to compute C_{sm} and $DUID_i$, the adversary does not know the long-term secret key, and hence $\mathcal{A}$ cannot recreate the login request message. Therefore the scheme is secured from Man-in-the-middle attacks.

4) PROPOSITION 4–THE SCHEME IS SECURE FROM OFFLINE GUESSING ATTACKS

Proof: Assume that $\mathcal{A}$ steals the device and succeeds in extracting stored parameters $\{B_i, R_i, Gen(.), Rep(.), \tau_i, h_1(.),$

$h_2(.), V_i, RT_i\}$. To perform the attack, $\mathcal{A}$ must guess both ID_i, PW_i, and the biometric character σ_i. The adversary can verify guessed PW_i through the $RPW = h_1(PW_i\|\sigma_i\|m_i')$ and $R_i = b_i \oplus h_1(ID_i\|PW_i\|\sigma_i)$ equations. However, it is difficult for $\mathcal{A}$ to verify PW_i, since the adversary must have knowledge of ID_i, PW_i, σ_i, and m_i. Hence the scheme is secure from offline password-guessing attacks.

5) PROPOSITION 5–THE SCHEME IS SECURE FROM PRIVILEGED-INSIDER ATTACK

Proof: If a privileged insider user is an adversary, then $\mathcal{A}$ can easily access the Trusted authority TA and get the registration information. Assume that insider $\mathcal{A}$ steals the device and succeeds in extracting stored parameters $\{B_i, R_i, Gen(.), Rep(.), \tau_i, h_1(.), h_2(.), V_i, RT_i\}$. $\mathcal{A}$ can easily get long-term secret key x, random nonce e, and computes $m_i = h_1(x\|e).G$. However, it is difficult for $\mathcal{A}$ to guess ID_i, PW_i, and the biometric character σ_i together. Therefore the scheme is secure from privileged-insider attacks.

6) PROPOSITION 6–THE SCHEME IS SECURE FROM LOST OR STOLEN MOBILE DEVICE ATTACKS

Proof: Let us consider $\mathcal{A}$ steals device and succeeded in extracting stored parameters $\{B_i, R_i, Gen(.), Rep(.), \tau_i, h_1(.), h_2(.), V_i, RT_i\}$. Also, assume that $\mathcal{A}$ intercepted communication messages $\{CID_s, RV_2, C_{sm}, T_1\}$, $\{CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2\}$, $\{CV_2, T_3\}$, and $\{F_{sm}, T_4, FCSUID_i, T_3, CV_2\}$ exchanged during login and authentication. Still, the session is secure because $\mathcal{A}$ doesn't have knowledge of ID_i, PW_i, and the biometric character σ_i. Also, the long-term session key x is unknown to $\mathcal{A}$. We have already proved the secrecy of the secret key. Therefore, it is not possible for $\mathcal{A}$ to guess the credentials and recreate the communication messages. Hence the scheme is also secure even if the mobile device is compromised.

7) PROPOSITION 7–THE SCHEME PRESERVES BOTH ANONYMITY AND UNTRACEABILITY

The user's ID_i is not transmitted in plain text, thereby ensuring anonymity and untraceability. During the user's login and authentication phase, a dynamic identity $C_{sm} = h_2(CID_s\|T_1\|RV_1) \oplus h(x\|e)'$ is generated using a timestamp. C_{sm} is then sent to the fog node. The C_{sm} generation occurs in each login and authentication session, resulting in a different identity with each attempt. Therefore anonymity and untraceability are preserved in the proposed scheme.

8) PROPOSITION 8–THE SCHEME IS SECURE FROM PARALLEL SESSION ATTACK

Suppose the attacker intercepts login request messages $\{CID_s, RV_2, C_{sm}, T_1\}$, $\{CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2\}$ and mutual authentication messages $\{CV_2, T_3\}$, $\{F_{sm}, T_4, FCSUID_i, T_3, CV_2\}$ to generate a session using old data. However, the proposed scheme uses a dynamic identity $DUID_i = h_2(CID_s\|RV_1\|T_1\|m_i)$ that changes every session and requires parameters CID_s, RV_1, m_i that are not publicly available to the attacker. This makes it difficult for the attacker to perform a parallel session attack on the scheme.

9) PROPOSITION 9–THE SCHEME IS SECURE FROM REFLECTION ATTACK

Suppose that the attacker has intercepted login request messages and mutual authentication messages, $\{CID_s, RV_2, C_{sm}, T_1\}$, $\{CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2\}$, $\{CV_2, T_3\}$, and $\{F_{sm}, T_4, FCSUID_i, T_3, CV_2\}$. If the attacker attempts to perform a reflection attack by substituting T_4 in place of T_1 in the message $\{CID_s, RV_2, C_{sm}, T_1\}$, then modify it to $\{CID_s, RV_2, C_{sm}, T_4\}$ and sends it to the fog node, the message will be verified as valid since the condition $T_4 - T_4^* \leq \Delta T$ holds.

However, the proposed scheme is designed to resist such attacks because the login request parameter C_{sm} requires the current timestamp to be included in the computation, i.e., $C_{sm} = h_2(CID_s\|T_1\|RV_1) \oplus h(x\|e)'$. Since the proposed scheme also resists impersonation attacks, the attacker can't compute C_{sm}' using the intercepted timestamp T_4. As a result, the proposed scheme is effective in resisting reflection attacks.

10) PROPOSITION 10–THE SCHEME IS SECURE AGAINAST DESYNCHRONIZATION ATTACK

The proposed scheme updates the dynamic identity $C_{sm} = h_2(CID_s\|T_1\|RV_1) \oplus h(x\|e)'$ only after mutual authentication is completed. Moreover, it is only possible for $\mathcal{A}$ to alter intercepted messages with access to the random nonces w_1, w_2, and w_3, which are generated only after credential verification. As a result, the scheme is secure against desynchronization attacks.

B. FORMAL SECURITY USING THE ROR MODEL

This section performs a formal security analysis to demonstrate the security of the proposed scheme against the adversary outlined in [46], as proposed by [47]. In this model, adversary $\mathcal{A}$ has full control over the communication channel and can eavesdrop, intercept, and modify the communication messages. Additionally, $\mathcal{A}$ has knowledge of all the public parameters but has no direct access to the secret parameter. Nevertheless, $\mathcal{A}$ can generate queries to extract information.

1) PARTICIPANTS

In our scheme, authentication involves four distinct entities called participants. These participants include the User (U_i), the Smart Device (D_k), the Fog Node (FS_j), and the Cloud Server (CS_l). Each participant can have multiple instances of the scheme in parallel, denoted as U^i, D^i, FS^i, and CS^i, where 'i' denotes the i^th participant instance [48].

- $Execute(U^i, D^i, FS^i, CS^i)$: The eavesdropping attack is a query that enables $\mathcal{A}$ to simulate the login and authentication and retrieve the communication transcript in the i^{th} instance of participants.

- $Send(U^i/D^i/FS^i/CS^i, M)$: Adversary $\mathcal{A}$ employs this query to launch active attacks. Using this query, $\mathcal{A}$ can intercept the message M exchanged among the instances U_i, D_k, FS_j, and CS_l. Additionally, $\mathcal{A}$ attempts to modify the intercepted message. In other words, the query produces a message M sent by the participant U_i, D_k, FS_j, or CS_l.
- $Reveal(U^i/FS^i/CS^i)$: Using this query, an adversary can retrieve the ephemeral secret key information of the instance $U_i/FS_j/CS_l$.
- $Corrupt(U^i/D^i/FS^i/CS^i)$: Using the oracle, adversary $\mathcal{A}$ can obtain the session key even after the long-term secret key is compromised.
- $Test(U^i/D^i/FS^i/CS^i)$: This query can be constructed only once, and it models the semantic security of the session. When this query is made, $\mathcal{A}$ returns the session key held by $U_i/FS_j/CS_l$, or it returns a random string with the same length. The outcome depends on the result of a coin toss. If the toss results in b = 1, the adversary receives the original session key; otherwise, $\mathcal{A}$ is provided with a random string of equal length to the actual session key.

Before demonstrating the security of the proposed scheme, it is necessary to provide certain definitions.

- Partnering: For a secure communication channel, two entities must share a common session key. In the proposed scheme, the entities U_i, D_k, FS_j, and CS_l are considered partners if and only if they share the same session key. Specifically, the session key SK_{sfc} shared between U_i and FS_j, the session key SK_{fcs} shared between FS_j and CS_l, and the session key SK_{cfs} shared between CS_l and FS_j must all be equal in order for the entities to be considered partners.
- Freshness: Freshness in this context refers to the newly created session key. A session key is considered fresh if it satisfies the below-mentioned conditions when it is constructed by the oracle:
  1) The session keys must not be null, and no *Reveal* query must be constructed between U_i, D_k, FS_j, and CS_l for freshness to be ensured.
  2) After constructing the *Currupt* query $Send(U^i/D^i/FS^i/CS^i, M)$ query should be asked
- Semantic Security: One task for $\mathcal{A}$ is to identify the actual session key of a participant and a random key of the same size. This requires $\mathcal{A}$ to execute multiple queries, such as *Execute*, *Send*, *Reveal*, and *Corrupt*, and also conduct *Test* queries for U_i, FS_j, and CS_l. To illustrate, take the example of U_i. When U_i flips a coin and guesses b, $\mathcal{A}$ will receive the genuine session key SK_{sfc} if $b = 1$. If b equals 0, $\mathcal{A}$ obtains a random string of the same length as the actual session key.
  We define the winning probability of adversary $\mathcal{A}$ as $Pr[Succ]$. The advantage of adversary $\mathcal{A}$ if it breaches the semantic security of the proposed scheme is denoted by $Adv_P^{Auth}(A) = |2Pr[Succ] - 1|$. The authentication scheme is considered secure if $Adv_P^{Auth}(A)$ is negligible for any probabilistic polynomial time adversary $\mathcal{A}$.

C. COMPUTATIONAL PROBLEM

The security analysis of the proposed scheme is grounded on solving the following computational problems:

- Elliptic curve computational Diffie-Hellman problem (ECDH): Given P, xP, and yP on an elliptic curve E_p with $a, b \in Z_q^*$, it is computationally difficult to determine xyP without knowing either x or y in polynomial time.
- Elliptic curve discrete logarithm problem (ECDLP): When $G \in E_p(x, y)$ of order n and $G = kP \in E_p(x, y)$, it is computationally hard to find k within polynomial time.
- Reversing One-way Hash function: For any given input x, a one-way hash function is easy to compute; however, it is computationally difficult to reverse and find x from $H(x)$. It is also challenging to find a different input x' that produces the same output hash value as x or $H(x)$.

D. SECURITY PROOF

Theorem 7.1: Considering a probabilistic polynomial time, adversary $\mathcal{A}$ has the intention of breaching the semantic security of the proposed scheme. The scheme employs E_p, an elliptic curve over a finite field F_p where p is a large prime number. The adversary uses a uniformly distributed finite set of passwords, denoted by $\mathcal{D}$, and σ_i bits in the biometric secret key. The adversary's ability to solve ECDH in E_p and its advantage against the proposed scheme is denoted by $Adv^{Auth}P(\mathcal{A})$. To accomplish this, adversary $\mathcal{A}$ performs send queries, hash oracles, and executes queries, represented as q_{send}, q_{hsh}, q_{exe}, within time t. The advantage for $\mathcal{A}$ is as follows:

$$Adv_P^{Auth}(\mathcal{A}) \leq \frac{q_{send}}{2^n.|\mathcal{D}|} + \frac{q_{hsh}^2}{|HASH|} + 2Adv_{EC}^{ECDH}(\mathcal{A})$$

Proof: The proof consists of a sequence of experiments denoted as Exp_i where $i = 0, 1, 2, 3, 4$ which are based on the queries generated by adversary $\mathcal{A}$. Let $Succ_n$ represent the event when adversary $\mathcal{A}$ guesses the bit b after making the *Test* query.

$Experiment_0$: In this experiment, the adversary $\mathcal{A}$ constructs attacks within the framework of the ROR model. According to the definition, we have

$$Adv_{AKE}^{SG} \leq 2Pr[Succ_0] - 1 \tag{1}$$

$Experiment_1$: $\mathcal{A}$ attempts to perform an eavesdropping attack by constructing an *Execute* query and a *Test* query to determine the session key SK communicated between U_i, D_k, FS_j, and CS_l. The goal is to distinguish between the actual key and a random number. The session key $SK_{sfc/fcs/cfs}$ in the proposed scheme can be calculated using the equation $SK_{sfc} = h_2(m_i' \| RV_1 \| FV_1' \| CV_1')$. Consider, $\mathcal{A}$ intercepts all the messages communicated in the Login and authentication phase CID_s, RV_2, C_{sm}, T_1, $CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2, CV_2, T_3$, and $F_{sm}, T_4, FCSUID_i, T_3, CV_2$ between U_i, D_k, FS_j, and CS_l. However, the adversary cannot calculate m_i', RV_1, FV_1', CV_1' through these intercepted messages. Therefore, the winning probability of $\mathcal{A}$ through an eavesdropping attack is not changed. As a result, there is no change in $Experiment_0$ and $Experiment_1$. This implies that

$$Pr[Succ_0] = Pr[Succ_1] \tag{2}$$

Experiment$_2$: This experiment was formed by adding *send*, and *hash* queries $\mathcal{HO}$ on the $Experiment_1$. This experiment simulates the active attacks. Assume that $\mathcal{A}$ intercepts the communicated messages $\{CID_s, RV_2, C_{sm}, T_1\}$, $\{CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2\}$, $\{CV_2, T_3\}$, and $\{F_{sm}, T_4, FCSUID_i, T_3, CV_2\}$ during login and authentication phase and forge a message. If $\mathcal{A}$ tries to modify any of these messages, $\mathcal{A}$ must have knowledge of $C_{sm}, FUID_i, F_{sm}$, and $FCSUID_i$ which is secure because of the collision-resistant one-way hash function h(·). Also, using random numbers, timestamps, and dynamic identity helps to protect the parameters from $\mathcal{A}$ to construct the queries. Therefore it is clear that $Experiment_1$ and $Experiment_2$ are equal if $\mathcal{A}$ fails to frame *send* and *hash* queries. According to the result of the birthday paradox, we have:

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{(q_{hash})^2}{2.|Hash|} \tag{3}$$

Experiment$_3$ This experiment aims to model a lost/stolen device attack by incorporating the *Corrupt* query into $Experiment_2$. In this scenario, $\mathcal{A}$ creates a *Corrupt* query on the device to extract all stored information $B_i, R_i, Gen(.), Rep(.), \tau_i, h_1(.), h_2(.), V_i, RT_i$. $\mathcal{A}$ then performs an offline guessing attack using the equations $B_i = h_1(H_n' \| RPW \| b_i)$ and $R_i = b_i \oplus h_1(ID_i \| PW_i \| \sigma_i)$, using the information obtained from the mobile device. The proposed scheme utilizes a fuzzy extractor method for biometric verification, and the probability of $\mathcal{A}$ guessing the biometric key $\sigma_i \in 0, 1^n$ is approximately $1/2^n$. If the system restricts the number of incorrect password inputs, the following result can be derived:

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{q_{send}}{2^n|\mathcal{D}|} \tag{4}$$

Experiment$_4$ To simulate the session key security using the *Corrupt* query, the final experiment involves $\mathcal{A}$ attempting to obtain the session key $SK_{sfc/fcs/cfs}$ through the equation $SK_{sfc} = h_2(m_i' \| RV_1 \| FV_1' \| CV_1')$. To compute the session key, $\mathcal{A}$ must also compute $RV_1 = w_1.F_{pub}$, $FV_1 = w_2.C_{pub}$, and $CV_1 = w_3.F_{pub}$ from the intercepted message parameter. However, since $\mathcal{A}$ needs more information to compute $SK_{sfc/fcs/cfs}$ without solving the ECDH, Experiments 3 and 4 are indistinguishable as long as the ECDH assumption is true. Therefore, we obtain

$$|Pr[Succ_4] - Pr[Succ_3]| = Adv_{EC}^{ECDH}(\mathcal{A}) \tag{5}$$

TABLE 7. Notations and execution time of cryptographic operations.

Notation	Description	Estimated Execution time
T_h	One way Hash operation	0.5 ms
T_{eca}	ECC point addition	10.875 ms
T_{ecm}	ECC point multiplication	63.075 ms
T_{ge}	Bilinear pairing	49.6 ms
T_{fe}	Fuzzy Extractor $Gen(.) or Rep(.)$ function	63.075 ms

At this point, in order to complete the experiment using the *Test* query, $\mathcal{A}$ must make a guess for b. It is evident that

$$Pr[Succ_4] = 1/2 \tag{6}$$

Based on equation (1) - (6) we can get

$$Adv_P^{Auth}(\mathcal{A}) \leq \frac{q_{send}}{2^n.|\mathcal{D}|} + \frac{q_{hsh}^2}{|HASH|} + 2Adv_{EC}^{ECDH}(\mathcal{A})$$

E. FORMAL SECURITY VERIFICATION USING SCYTHER SIMULATION

Scyther is a specialized tool created for the formal analysis of security protocols, aimed at identifying flaws and evaluating security requirements [49]. By automatically examining the behavior of the protocol in relation to potential attacks, Scyther produces an output that includes claims specifying essential security requirements such as Alive, Nisynch, weak agree, and secret. These requirements ensure the proper execution of intended events, accurate message synchronization, safeguarding of sensitive information, and resilience against impersonation attacks.

In terms of input language, Scyther utilizes the Security Protocol Descriptive Language (SPDL) to describe the specifications of security protocols. The specification involves defining a set of roles, such as User, FogNode, and CloudServer. Figure 2(a) illustrates the initial setup, while Figure 2(b), Figure 3(a), and Figure 3(b) represent the roles of User, FogNode, and CloudServer, respectively.

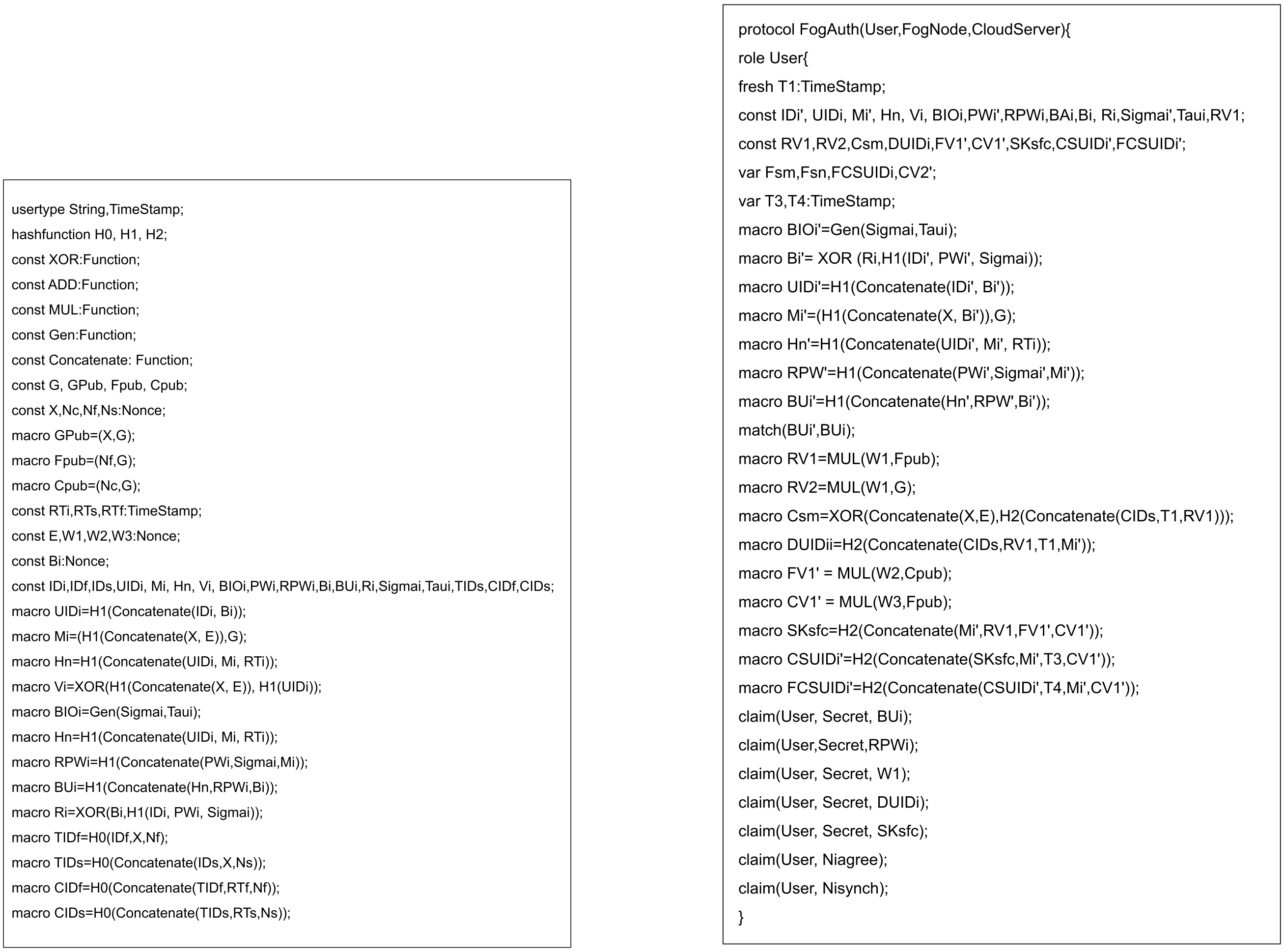
The proposed scheme utilizes the concept of "Claim" to define specific security requirements. Claims like *Niagree* and *Nisynch* ensure that both the sender and receiver have successfully exchanged all the required messages. Claims made with *secret* indicate that the parameters mentioned within those claims remain unknown to any potential adversaries. The simulation outcome is depicted in Figure 4, demonstrating that the proposed scheme effectively satisfies all the specified security requirements, without succumbing to any attacks.

VIII. EFFICIENCY ANALYSIS

In this section the computation and communication costs are tabulated, and the results are compared to other schemes to analyse the performance of the proposed scheme. Also, the functional analysis is tabulated in and a detailed discussion of the proposed scheme is done.

A. COMPUTATION AND COMMUNICATION COSTS ANALYSIS

The authors analyze the efficiency of the scheme by focusing on the computation and communication costs and compare it



```
usertype String,TimeStamp;
hashfunction H0, H1, H2;
const XOR:Function;
const ADD:Function;
const MUL:Function;
const Gen:Function;
const Concatenate: Function;
const G, GPub, Fpub, Cpub;
const X,Nc,Nf,Ns:Nonce;
macro GPub=(X,G);
macro Fpub=(Nf,G);
macro Cpub=(Nc,G);
const RTi,RTs,RTf:TimeStamp;
const E,W1,W2,W3:Nonce;
const Bi:Nonce;
const IDi,IDf,IDs,UIDi, Mi, Hn, Vi, BIOi,PWi,RPWi,Bi,BUi,Ri,Sigmai,Taui,TIDs,CIDf,CIDs;
macro UIDi=H1(Concatenate(IDi, Bi));
macro Mi=(H1(Concatenate(X, E)),G);
macro Hn=H1(Concatenate(UIDi, Mi, RTi));
macro Vi=XOR(H1(Concatenate(X, E)), H1(UIDi));
macro BIOi=Gen(Sigmai,Taui);
macro Hn=H1(Concatenate(UIDi, Mi, RTi));
macro RPWi=H1(Concatenate(PWi,Sigmai,Mi));
macro BUi=H1(Concatenate(Hn,RPWi,Bi));
macro Ri=XOR(Bi,H1(IDi, PWi, Sigmai));
macro TIDf=H0(IDf,X,Nf);
macro TIDs=H0(Concatenate(IDs,X,Ns));
macro CIDf=H0(Concatenate(TIDf,RTf,Nf));
macro CIDs=H0(Concatenate(TIDs,RTs,Ns));
```

(a) Initial Setup

```
protocol FogAuth(User,FogNode,CloudServer){
role User{
fresh T1:TimeStamp;
const IDi', UIDi, Mi', Hn, Vi, BIOi,PWi',RPWi,BAi,Bi, Ri,Sigmai',Taui,RV1;
const RV1,RV2,Csm,DUIDi,FV1',CV1',SKsfc,CSUIDi',FCSUIDi';
var Fsm,Fsn,FCSUIDi,CV2';
var T3,T4:TimeStamp;
macro BIOi'=Gen(Sigmai,Taui);
macro Bi'= XOR (Ri,H1(IDi', PWi', Sigmai));
macro UIDi'=H1(Concatenate(IDi', Bi'));
macro Mi'=(H1(Concatenate(X, Bi')),G);
macro Hn'=H1(Concatenate(UIDi', Mi', RTi));
macro RPW'=H1(Concatenate(PWi',Sigmai',Mi'));
macro BUi'=H1(Concatenate(Hn',RPW',Bi'));
match(BUi',BUi);
macro RV1=MUL(W1,Fpub);
macro RV2=MUL(W1,G);
macro Csm=XOR(Concatenate(X,E),H2(Concatenate(CIDs,T1,RV1)));
macro DUIDii=H2(Concatenate(CIDs,RV1,T1,Mi'));
macro FV1' = MUL(W2,Cpub);
macro CV1' = MUL(W3,Fpub);
macro SKsfc=H2(Concatenate(Mi',RV1,FV1',CV1'));
macro CSUIDi'=H2(Concatenate(SKsfc,Mi',T3,CV1'));
macro FCSUIDi'=H2(Concatenate(CSUIDi',T4,Mi',CV1'));
claim(User, Secret, BUi);
claim(User,Secret,RPWi);
claim(User, Secret, W1);
claim(User, Secret, DUIDi);
claim(User, Secret, SKsfc);
claim(User, Niagree);
claim(User, Nisynch);
}
```

(b) Role User

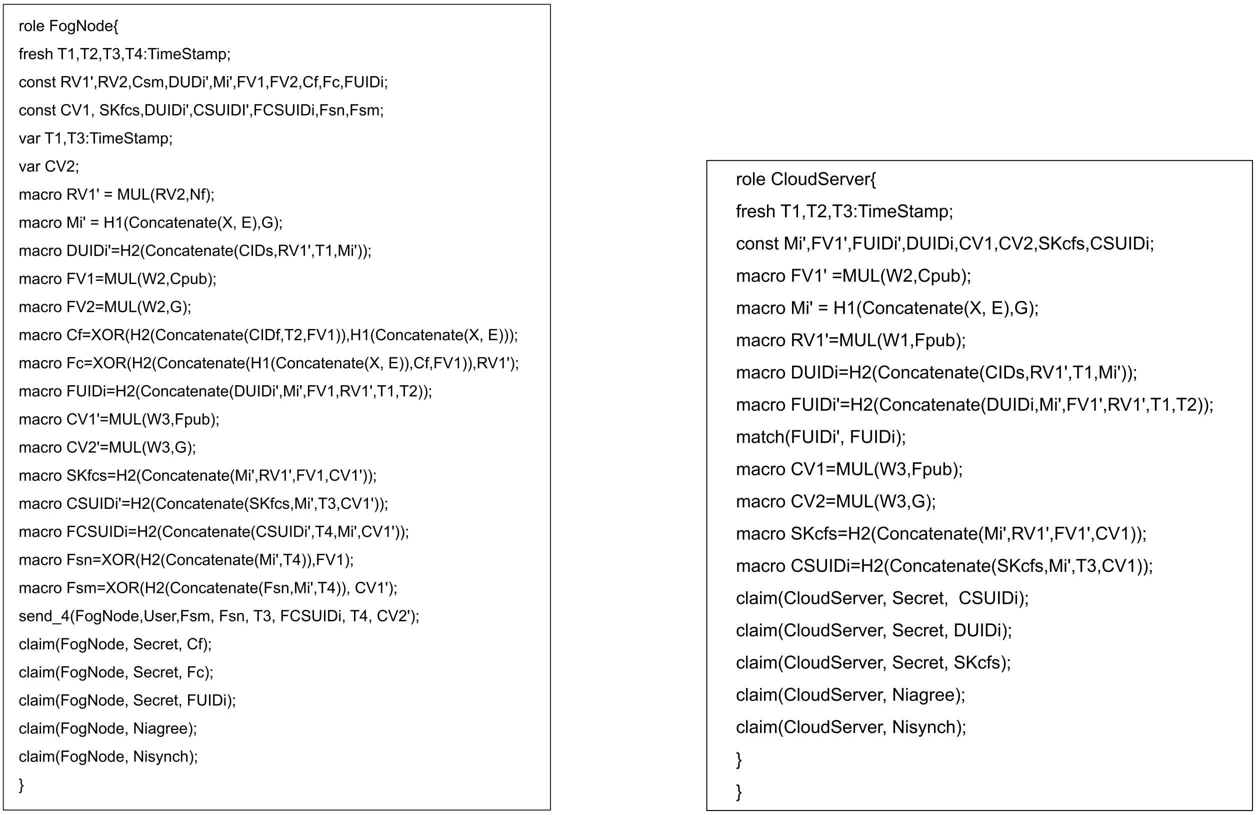
FIGURE 2. Scyther simulation setup.

TABLE 8. Computation cost analysis.

Schemes	U_i/D_i	FS_j	CS_l	**Total**	**Expected Time**
Wazid et al. [21]	$1T_{fe} + 2T_{ecm} + 25T_h$	$3T_{ecm} + 10T_h$	$--$	$1T_{fe} + 5T_{ecm} + 35T_h$	395.95 ms
Ma et al. [38]	$3T_{ecm} + 4T_h$	$4T_{ecm} + 4T_h$	$10T_{ecm} + 10T_h$	$17T_{ecm} + 18T_h$	1081.275 ms
Ali et al. [11]	$1T_{fe} + 3T_{ecm} + 18T_h$	$4T_{ecm} + 1T_{eca} + 8T_h$	$--$	$1T_{fe} + 7T_{ecm} + 1T_{eca} + 26T_h$	528.475 ms
Amin et al.[39]	$6T_{ecm} + 2T_{eca} + 17T_h$	–	$2T_{ecm} + 4T_h$	$8T_{ecm} + 2T_{eca} + 21T_h$	536.85 ms
Chattarjee et al. [29]	$4T_{ecm} + 11T_h$	$1T_{ecm} + 4T_h$	$--$	$5T_{ecm} + 13T_h$	321.875 ms
Chen et al. [24]	$1T_{fe} + 2T_{ecm} + 6T_h$	$3T_{ecm} + 12T_h$	$4T_{ecm} + 4T_h$	$1T_{fe} + 9T_{ecm} + 22T_h$	641.75 ms
Li et al. [40]	$8T_h$	$7T_h$	$11T_h$	$26T_h$	13 ms
Jia et al. [23]	$1T_{ge} + 2T_{ecm} + 5T_h$	$1T_{ge} + 2T_{ecm} + 4T_h$	$1T_{ge} + 3T_{ecm} + 9T_h$	$3T_{ge} + 7T_{ecm} + 18T_h$	599.325 ms
Proposed Scheme	$1T_{fe} + 2T_{ecm} + 1T_{eca} + 13T_h$	$2T_{ecm} + 3T_{eca} + 10T_h$	$2T_{ecm} + 2T_{eca} + 6T_h$	$1T_{fe} + 6T_{ecm} + 6T_{eca} + 29T_h$	521.275 ms

with related authentication schemes such as [11], [21], [23], [24], [29], [38], [39], and [40].

Initially, we present the computation cost and the estimated execution time of the scheme. The computational parameters necessary to compute the cost are presented in Table 7. In our analysis, we take into account the total computation cost and expected execution time of the scheme. To estimate the expected execution time, we rely on existing evaluation results of various cryptographic operations, as presented by [11], [50], [51], and [52]. The estimated execution time for each cryptographic function is presented in Table 7. To calculate the computation cost of the proposed scheme, we consider one complete round of login and authentication between U_i, D_k, FS_j, and CS_l. In computing this cost,



```
role FogNode{
fresh T1,T2,T3,T4:TimeStamp;
const RV1',RV2,Csm,DUDi',Mi',FV1,FV2,Cf,Fc,FUIDi;
const CV1, SKfcs,DUIDi',CSUIDI',FCSUIDi,Fsn,Fsm;
var T1,T3:TimeStamp;
var CV2;
macro RV1' = MUL(RV2,Nf);
macro Mi' = H1(Concatenate(X, E),G);
macro DUIDi'=H2(Concatenate(CIDs,RV1',T1,Mi'));
macro FV1=MUL(W2,Cpub);
macro FV2=MUL(W2,G);
macro Cf=XOR(H2(Concatenate(CIDf,T2,FV1)),H1(Concatenate(X, E)));
macro Fc=XOR(H2(Concatenate(H1(Concatenate(X, E)),Cf,FV1)),RV1');
macro FUIDi=H2(Concatenate(DUIDi',Mi',FV1,RV1',T1,T2));
macro CV1'=MUL(W3,Fpub);
macro CV2'=MUL(W3,G);
macro SKfcs=H2(Concatenate(Mi',RV1',FV1,CV1'));
macro CSUIDi'=H2(Concatenate(SKfcs,Mi',T3,CV1'));
macro FCSUIDi=H2(Concatenate(CSUIDi',T4,Mi',CV1'));
macro Fsn=XOR(H2(Concatenate(Mi',T4)),FV1);
macro Fsm=XOR(H2(Concatenate(Fsn,Mi',T4)), CV1');
send_4(FogNode,User,Fsm, Fsn, T3, FCSUIDi, T4, CV2');
claim(FogNode, Secret, Cf);
claim(FogNode, Secret, Fc);
claim(FogNode, Secret, FUIDi);
claim(FogNode, Niagree);
claim(FogNode, Nisynch);
}
```

(a) Role Fog node

```
role CloudServer{
fresh T1,T2,T3:TimeStamp;
const Mi',FV1',FUIDi',DUIDi,CV1,CV2,SKcfs,CSUIDi;
macro FV1' =MUL(W2,Cpub);
macro Mi' = H1(Concatenate(X, E),G);
macro RV1'=MUL(W1,Fpub);
macro DUIDi=H2(Concatenate(CIDs,RV1',T1,Mi'));
macro FUIDi'=H2(Concatenate(DUIDi,Mi',FV1',RV1',T1,T2));
match(FUIDi', FUIDi);
macro CV1=MUL(W3,Fpub);
macro CV2=MUL(W3,G);
macro SKcfs=H2(Concatenate(Mi',RV1',FV1',CV1));
macro CSUIDi=H2(Concatenate(SKcfs,Mi',T3,CV1));
claim(CloudServer, Secret,  CSUIDi);
claim(CloudServer, Secret, DUIDi);
claim(CloudServer, Secret, SKcfs);
claim(CloudServer, Niagree);
claim(CloudServer, Nisynch);
}
}
```

(b) Role Cloud server

FIGURE 3. Scyther simulation setup.

we have excluded XOR and Concatenation operations, as their execution time is negligible.

The total computation cost of U_i and D_k is $1T_{fe} + 2T_{ecm} + 1T_{eca} + 13T_h$ and a fuzzy extractor is used for the authenticating biometric character. The scheme also used $2T_{ecm}$ and $1T_{eca}$ to secure the random nonce and computation parameters for dynamic ID. $13T_h$ was used to perform the hash operation for securing the login request parameters. To perform authentication through FS_j, at least $2T_{ecm}$ is required to retrieve parameter RV' received from U_i and compute m_i', which is one of the variables used for computing the login request parameter. Similarly, $3T_{eca}$ is required to compute other request parameters for authentication and session key generation. Apart from that, $10T_h$ is required for parameter security. Therefore, the total cost of FS_j is $2T_{ecm} + 3T_{eca} + 10T_h$. On the CS_l side, $2T_{ecm}$ is used to retrieve FV_1 and m_i' required for authentication, and $2T_{eca}$ is used for computing mutual authentication parameters. Also, $6T_h$ is used for session key computation and securing mutual authentication parameters. Hence the total cost CS_l is $2T_{ecm} + 2T_{eca} + 6T_h$. Therefore, the overall computation cost of the proposed scheme is $1T_{fe} + 6T_{ecm} + 6T_{eca} + 29T_h$. The overall estimated execution time of our scheme is $(1 * 63.075\text{ms}) + (6 * 63.075\text{ms}) + (6 * 10.875\text{ms}) + (29 * 0.5\text{ms}) = 521.275$ ms.

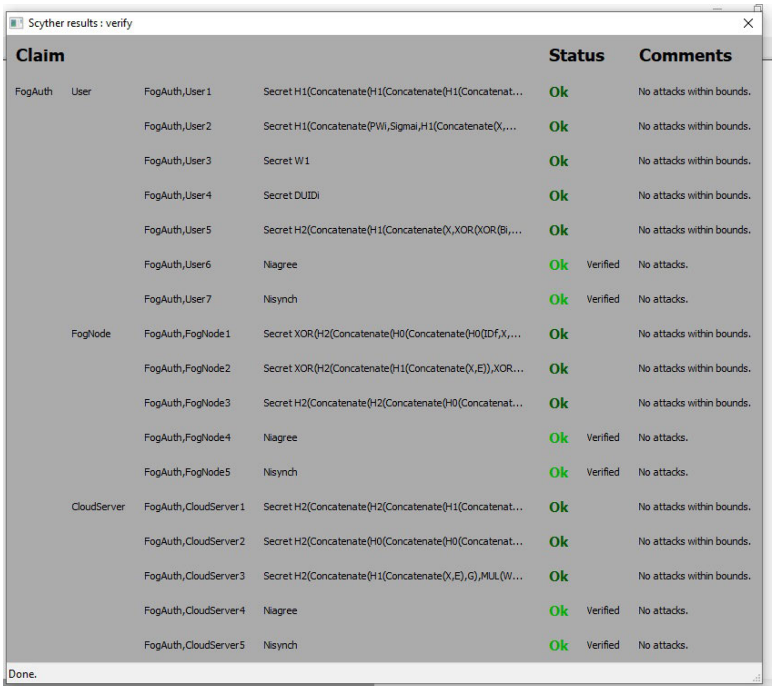
TABLE 9. Communication cost analysis.

Schemes	Number of Messages	Communication Cost
Wazid et al. [21]	3	2816 bits
Ma et al. [38]	4	4800 bits
Ali et al. [11]	3	2816 bits
Amin et al.[39]	4	2144 bits
Chattarjee et al. [29]	3	2016 bits
Chen et al. [24]	4	4768 bits
Li et al. [40]	4	2208 bits
Jia et al. [23]	4	3520 bits
Proposed Scheme	4	3464 bits

The computation cost of the proposed scheme was compared with relevant schemes and is presented in Table 8. The results of Table 8 shows that the proposed scheme has a significantly lower overall computation cost compared the schemes [11], [24], [38], [39], and [23] Further, [40] scheme's computation cost is slightly better than ours because the scheme uses only hash functions for authentication. However, [53] identified that using only hash functions without any public key cryptographic techniques in the authentication scheme will lead to a loss of user anonymity. Hence, we used public-key techniques. The computation cost of the proposed scheme is higher than [21] and [29] authentication schemes. But still, the proposed scheme is justifiable because the scheme authenticates every entity

TABLE 10. Functional analysis.

Schemes	$\mathbf{F}n1$	$\mathbf{F}n2$	$\mathbf{F}n3$	$\mathbf{F}n4$	$\mathbf{F}n5$	$\mathbf{F}n6$	$\mathbf{F}n7$	$\mathbf{F}n8$
Wazid et al. [21]	✓	✓	✓	✓	×	✓	×	×
Ma et al. [38]	✓	✓	✓	✓	×	×	×	✓
Ali et al. [11]	×	✓	✓	✓	×	✓	×	×
Amin et al.[39]	✓	✓	✓	✓	×	✓	×	×
Chattarjee et al. [29]	✓	×	✓	✓	✓	✓	✓	×
Chen et al. [24]	✓	✓	✓	✓	×	✓	✓	✓
Li et al. [40]	✓	✓	✓	✓	✓	✓	×	✓
Jia et al. [23]	×	✓	✓	✓	✓	✓	✓	✓
Proposed Scheme	✓	✓	✓	✓	✓	✓	✓	✓



```
Scyther results : verify

Claim                                                                                          Status            Comments
FogAuth  User         FogAuth,User1          Secret H1(Concatenate(H1(Concatenate(H1(Concatenat...   Ok                No attacks within bounds.
                      FogAuth,User2          Secret H1(Concatenate(PWi,Sigmai,H1(Concatenate(X,...   Ok                No attacks within bounds.
                      FogAuth,User3          Secret W1                                               Ok                No attacks within bounds.
                      FogAuth,User4          Secret DUIDi                                            Ok                No attacks within bounds.
                      FogAuth,User5          Secret H2(Concatenate(H1(Concatenate(X,XOR(XOR(Bi,...   Ok                No attacks within bounds.
                      FogAuth,User6          Niagree                                                 Ok    Verified    No attacks.
                      FogAuth,User7          Nisynch                                                 Ok    Verified    No attacks.
         FogNode      FogAuth,FogNode1       Secret XOR(H2(Concatenate(H0(Concatenate(H0(IDf,X,...   Ok                No attacks within bounds.
                      FogAuth,FogNode2       Secret XOR(H2(Concatenate(H1(Concatenate(X,E)),XOR...   Ok                No attacks within bounds.
                      FogAuth,FogNode3       Secret H2(Concatenate(H2(Concatenate(H0(Concatenat...   Ok                No attacks within bounds.
                      FogAuth,FogNode4       Niagree                                                 Ok    Verified    No attacks.
                      FogAuth,FogNode5       Nisynch                                                 Ok    Verified    No attacks.
         CloudServer  FogAuth,CloudServer1   Secret H2(Concatenate(H2(Concatenate(H1(Concatenat...   Ok                No attacks within bounds.
                      FogAuth,CloudServer2   Secret H2(Concatenate(H0(Concatenate(H0(Concatenat...   Ok                No attacks within bounds.
                      FogAuth,CloudServer3   Secret H2(Concatenate(H1(Concatenate(X,E),G),MUL(W...   Ok                No attacks within bounds.
                      FogAuth,CloudServer4   Niagree                                                 Ok    Verified    No attacks.
                      FogAuth,CloudServer5   Nisynch                                                 Ok    Verified    No attacks.
Done.
```

FIGURE 4. Scyther simulation - Role Cloud server.

involved in the communication, that is, U_i, D_k, FS_j and CS_l. In [21] and [29] authentication schemes the assumption is that the cloud server CS_l is a secure entity. In section I the authors have identified the need for authenticating the cloud server.

The communication cost analysis of our scheme is presented in Table 9. For consistency in comparison, we assume the following: The length of the identity is 128 bits, the size of the timestamp is 32 bits, the size of an elliptic curve point is 320 bits, the size of the hash function is 160 bits, and the size of the random number is 128 bits. During login and authentication phase, the request messages $\{CID_s, RV_2, C_{sm}, T_1\}$ and $\{CID_s, CID_f, C_{sm}, C_f, F_c, FUID_i, RV_2, FV_2, T_1, T_2\}$ requires $(128 + 360 + 160 + 32) = 680$ bits and $(128 + 128 + 128 + 160 + 160 + 160 + 360 + 360 + 32 + 32) = 1648$ bits. The mutual authentication messages $\{CV_2, T_3\}$, and $\{F_{sm}, T_4, FCSUID_i, T_3, CV_2\}$ requires $(360 + 32) = 392$ bits and $160 + 32 + 160 + 32 + 360 = 744$ bits. The total communication cost of the scheme during the login and authentication phase is 3464 bits. Compared to the other related schemes, the communication cost of the proposed scheme is less than [23], [24], and [38] scheme. The communication cost of the scheme is more when compared to [11], [21], and [29]. Still, it is acceptable because the proposed scheme achieves multi-level authentication, wherein every entity involved in the communication is authenticated. Compared to [39] and [40], the proposed scheme communication cost is higher, but these schemes are vulnerable to multiple attacks which are addressed in our scheme.

B. FUNCTIONAL ANALYSIS

The security functionalities of our scheme were compared with other relevant schemes and presented in Table 10. To perform the functional analysis, we have considered the following parameters: Fn 1 - resilient against secret key leakage, Fn 2 - secure against replay attack, Fn 3 - Secure against a Man-in-the-middle attack, Fn 4 - secure against offline guessing attacks, Fn 5 - secure against privileged-insider attacks, Fn 6 - Secure against lost/stolen mobile device attacks, Fn 7 - preserves anonymity and untraceability, Fn 8 - Achieves multi-party authentication. Table 9 clearly proves that our scheme achieves all the security requirements mentioned in section VII-A. Other authentication schemes are unable to address the security issues of privileged insider attacks and preserve user anonymity. Most schemes do not address the multi-party authentication required for the cloud-fog-device framework.

IX. CONCLUSION AND FUTURE SCOPE

This article showcases an efficient and secure authentication scheme for fog-cloud-device architecture using key agreement and management. The authors reviewed Ali et al. scheme, which is an improved scheme from SAKA-FC. It was proved that the scheme could be breached through key revelation attacks and that it does not provide user anonymity or perfect forward secrecy. Cloud servers need to be authenticated too, and our scheme authenticates all the entities involved in the communication securely. Through rigorous cryptanalysis, the proposed scheme was thoroughly verified and the security was proven by performing a formal security analysis with the ROR model. The informal security analysis and the scyther simulation proved that the proposed scheme is secure against multiple active and passive attacks. The performance analysis proved that our scheme's computation and communication costs are much better than other relevant schemes. Further, the functional analysis proves that the proposed scheme exhibits all the functionalities required for a robust authentication scheme in the cloud-fog-device framework.

In the future, our focus would be further reducing the communication cost, improving the throughput, and reducing latency in the cloud computing environment. Most authentication schemes perform authentication using trusted third parties, which needs to be eliminated. The adoption of blockchain technology-based consensus algorithms in the authentication scheme could be a solution to the problem. Artificial intelligence-based techniques and blockchain technologies could revolutionize the authentication schemes for cloud-fog-device architecture.

REFERENCES

[1] Y. Guo, Z. Zhang, and Y. Guo, "SecFHome: Secure remote authentication in fog-enabled smart home environment," *Comput. Netw.*, vol. 207, Apr. 2022, Art. no. 108818.

[2] S. O. Ogundoyin and I. A. Kamil, "Secure and privacy-preserving D2D communication in fog computing services," *Comput. Netw.*, vol. 210, Jun. 2022, Art. no. 108942.

[3] S. Feng, P. Setoodeh, and S. Haykin, "Smart home: Cognitive interactive people-centric Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 34–39, Feb. 2017.

[4] S. A. Ahmed, "A performance study of hyperledger fabric in a smart home and IoT environment," M.S. thesis, Dept. Inform., Fac. Math. Natural Sci., Univ. Oslo, Oslo, Norway, 2019.

[5] M. Rahimi, M. Songhorabadi, and M. H. Kashani, "Fog-based smart homes: A systematic review," *J. Netw. Comput. Appl.*, vol. 153, Mar. 2020, Art. no. 102531.

[6] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[7] B. Alohali, M. Merabti, and K. Kifayat, "A secure scheme for a smart house based on cloud of things (CoT)," in *Proc. 6th Comput. Sci. Electron. Eng. Conf. (CEEC)*, Sep. 2014, pp. 115–120.

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

[9] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[10] Y. Guo and Y. Guo, "FogHA: An efficient handover authentication for mobile devices in fog computing," *Comput. Secur.*, vol. 108, Sep. 2021, Art. no. 102358.

[11] Z. Ali, S. A. Chaudhry, K. Mahmood, S. Garg, Z. Lv, and Y. B. Zikria, "A clogging resistant secure authentication scheme for fog computing services," *Comput. Netw.*, vol. 185, Feb. 2021, Art. no. 107731.

[12] Y. Lin, X. Wang, Q. Gan, and M. Yao, "A secure cross-domain authentication scheme with perfect forward security and complete anonymity in fog computing," *J. Inf. Secur. Appl.*, vol. 63, Dec. 2021, Art. no. 103022.

[13] M. Amadeo, A. Giordano, C. Mastroianni, and A. Molinaro, "On the integration of information centric networking and fog computing for smart home services," in *The Internet Things for Smart Urban Ecosystems*. Springer, 2019, pp. 75–93.

[14] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. MCC workshop Mobile cloud Comput.*, Aug. 2012, pp. 13–16.

[15] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 10, pp. 2991–3005, Jul. 2016.

[16] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Proc. 10th Int. Conf. Wireless Algorithms Syst. Appl.* Cham, Switzerland: Springer, 2015, pp. 685–695.

[17] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for Internet of Things applications: Challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 601–628, 1st Quart., 2018.

[18] Y. Imine, D. E. Kouicem, A. Bouabdallah, and L. Ahmed, "MASFOG: An efficient mutual authentication scheme for fog computing architecture," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 608–613.

[19] B. Huang, X. Cheng, Y. Cao, and L. Zhang, "Lightweight hardware based secure authentication scheme for fog computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 433–439.

[20] F. M. Salem, "A secure privacy-preserving mutual authentication scheme for publish-subscribe fog computing," in *Proc. 14th Int. Comput. Eng. Conf. (ICENCO)*, Dec. 2018, pp. 213–218.

[21] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Gener. Comput. Syst.*, vol. 91, pp. 475–492, Feb. 2019.

[22] F. Dewanta and M. Mambo, "A mutual authentication scheme for secure fog computing service handover in vehicular network environment," *IEEE Access*, vol. 7, pp. 103095–103114, 2019.

[23] X. Jia, D. He, N. Kumar, and K.-K.-R. Choo, "Authenticated key agreement scheme for fog-driven IoT healthcare system," *Wireless Netw.*, vol. 25, no. 8, pp. 4737–4750, Nov. 2019.

[24] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, and M.-E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Inf. Syst.*, vol. 15, no. 9, pp. 1200–1215, Oct. 2021.

[25] L. Wang, H. An, and Z. Chang, "Security enhancement on a lightweight authentication scheme with anonymity fog computing architecture," *IEEE Access*, vol. 8, pp. 97267–97278, 2020.

[26] R. Kalaria, A. S. M. Kayes, W. Rahayu, and E. Pardede, "A secure mutual authentication approach to fog computing environment," *Comput. Secur.*, vol. 111, Dec. 2021, Art. no. 102483.

[27] Y. Guo, Z. Zhang, and Y. Guo, "Fog-centric authenticated key agreement scheme without trusted parties," *IEEE Syst. J.*, vol. 15, no. 4, pp. 5057–5066, Dec. 2021.

[28] M. Hamada, S. A. Salem, and F. M. Salem, "LAMAS: Lightweight anonymous mutual authentication scheme for securing fog computing environments," *Ain Shams Eng. J.*, vol. 13, no. 6, Nov. 2022, Art. no. 101752.

[29] U. Chatterjee, S. Ray, M. K. Khan, M. Dasgupta, and C.-M. Chen, "An ECC-based lightweight remote user authentication and key management scheme for IoT communication in context of fog computing," *Computing*, vol. 104, no. 6, pp. 1359–1395, Jun. 2022.

[30] M. Wazid, A. K. Das, R. Hussain, N. Kumar, and S. Roy, "BUAKA-CS: Blockchain-enabled user authentication and key agreement scheme for crowdsourcing system," *J. Syst. Archit.*, vol. 123, Jun. 2022, Art. no. 102370.

[31] A. Vangala, S. Roy, and A. K. Das, "Blockchain-based lightweight authentication protocol for IoT-enabled smart agriculture," in *Proc. Int. Conf. Cyber-Phys. Social Intell. (ICCSI)*, Nov. 2022, pp. 110–115.

[32] N. C. Gowda, S. S. Manvi, and P. Lorenz, "BSKM-FC: Blockchain-based secured key management in a fog computing environment," *Future Gener. Comput. Syst.*, vol. 142, pp. 276–291, May 2023.

[33] N. C. Gowda, S. S. Manvi, A. B. Malakreddy, and R. Buyya, "TAKM-FC: Two-way authentication with efficient key management in fog computing environments," *J. Supercomput.*, vol. 80, no. 5, pp. 6855–6890, Mar. 2024.

[34] M. A. Akram, A. N. Mian, and S. Kumari, "Fog-based low latency and lightweight authentication protocol for vehicular communication," *Peer-Peer Netw. Appl.*, vol. 16, no. 2, pp. 629–643, 2023.

[35] P. S. Mahesh and K. Muthumanickam, "Secure and novel authentication model for protecting data centers in fog environment," *Wireless Netw.*, vol. 29, no. 4, pp. 1671–1683, 2023.

[36] Y. Huo, B. Kang, X. Zuo, S. Niu, and A. Li, "Analysis and improvement of authentication schemes for industrial wireless sensor networks with fog computing," *Frontiers Comput. Intell. Syst.*, vol. 4, no. 3, pp. 20–27, 2023.

[37] S. S. Sahoo, S. Mohanty, and B. Majhi, "An efficient three-factor user authentication scheme for industrial wireless sensor network with fog computing," *Int. J. Commun. Syst.*, vol. 35, no. 3, 2022, Art. no. e5028.

[38] M. Ma, D. He, H. Wang, N. Kumar, and K. R. Choo, "An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8065–8075, Oct. 2019.

[39] R. Amin, S. Kunal, A. Saha, D. Das, and A. Alamri, "CFSec: Password based secure communication protocol in cloud-fog environment," *J. Parallel Distrib. Comput.*, vol. 140, pp. 52–62, May 2020.

[40] Z. Li, Q. Miao, S. A. Chaudhry, and C.-M. Chen, "A provably secure and lightweight mutual authentication protocol in fog-enabled social Internet of Vehicles," *Int. J. Distrib. Sensor Netw.*, vol. 18, no. 6, 2022, Art. no. 15501329221104332.

[41] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 2, pp. 198–208, Mar. 1983.

[42] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly secure key distribution for dynamic conferences," *Inf. Comput.*, vol. 146, no. 1, pp. 1–23, 1998.

[43] Y. Zhou and Y. Fang, "A two-layer key establishment scheme for wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 9, pp. 1009–1020, Sep. 2007.

[44] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Interlaken, Switzerland. Cham, Switzerland: Springer, 2004, pp. 523–540.

[45] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[46] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1900–1910, May 2018.

[47] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 2001, pp. 453–474.

[48] J.-L. Tsai and N.-W. Lo, "Secure anonymous key distribution scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 906–914, Mar. 2016.

[49] C. J. F. Cremers, "Scyther: Semantics and verification of security protocols," Dept. Math. Comput. Sci., Eindhoven Univ. Technol., Eindhoven, The Netherlands, Tech. Rep., 2006.

[50] W. Li, Q. Wen, Q. Su, and Z. Jin, "An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network," *Comput. Commun.*, vol. 35, no. 2, pp. 188–195, 2012.

[51] C.-C. Lee, C.-T. Chen, P.-H. Wu, and T.-Y. Chen, "Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices," *IET Comput. Digit. Techn.*, vol. 7, no. 1, pp. 48–55, 2013.

[52] D. He, N. Kumar, J.-H. Lee, and R. S. Sherratt, "Enhanced three-factor security protocol for consumer USB mass storage devices," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 30–37, Feb. 2014.

[53] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions," *Comput. Netw.*, vol. 73, pp. 41–57, Nov. 2014.

MANJUNATH HEGDE received the master's degree in computer science from Mangalore University, Karnataka, India, in 2014, and the Ph.D. degree in mathematical and computational sciences from the National Institute of Technology Karnataka, India, in 2019. He is currently an Assistant Professor with the Department of Data Science and Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. His research interests include network security, information security, secure authentication, cryptography, and blockchain technology.

ROHINI R. RAO received the Ph.D. degree in electronic health records with respect to interoperability and privacy. She is currently a Faculty Member of the Department of Data Science and Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. She is currently a Researcher, working on cost-effective IT solutions and technology-based interventions for public health. Her current research interests include information security, data privacy, blockchain technologies, and health care analytics.

RADHAKRISHNA BHAT (Member, IEEE) received the Bachelor of Engineering degree from Government SKSJTI, Bengaluru, India, in 2011, and the integrated Ph.D. (M.Tech. and Ph.D.) degrees from Visveswaraya Technological University (VTU), Belagavi, India, in 2020. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE), Manipal, India. He is also an Active Researcher who has published more than 15 scientific research articles in reputed journals and conferences. He has ten years of industry and teaching experience at the undergraduate (UG) and postgraduate (PG) levels. He has also supervised a number of projects at different levels at the university. His research interests include information security, high-performance computing, blockchain technology, and machine learning.