IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 12, 2021, accepted October 21, 2021, date of publication October 26, 2021, date of current version November 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123234

A Lightweight Anonymous Authentication and Secure Communication Scheme for Fog Computing Services

CHI-YAO WENG[1], CHUN-TA LI[2], (Member, IEEE), CHIN-LING CHEN[3,4,5], CHENG-CHI LEE[6,7], (Member, IEEE), AND YONG-YUAN DENG[3]

[1]Department of Computer Science, National Pingtung University, Pingtung City 90003, Taiwan
[2]Department of Information Management, Tainan University of Technology, Tainan 71002, Taiwan
[3]Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan
[4]School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361005, China
[5]School of Information Engineering, Changchun Sci-Tech University, Changchun 130600, China
[6]Research and Development Center for Physical Education, Health, and Information Technology, Department of Library and Information Science, Fu Jen Catholic University, New Taipei City 24205, Taiwan
[7]Department of Computer Science and Information Engineering, Asia University, Taichung 41354, Taiwan

Corresponding authors: Chun-Ta Li (th0040@mail.tut.edu.tw), Chin-Ling Chen (clc@mail.cyut.edu.tw), and Cheng-Chi Lee (cclee@mail.fju.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract MOST 109-2410-H-165-001 and Contract MOST 110-2410-H-165-001-MY2.

ABSTRACT Fog-driven IoT architecture located between IoT devices and the centralized cloud infrastructure is introduced to extend computing, storage and network services to the edge of the Internet and therefore resources and services of the fog nodes are available and are closer to the end user and end device for providing mobility, low latency and location awareness. However, the paradigm of fog computing due to its inherited properties from cloud as inherits its security and privacy concerns such as spoofing, message replay, impersonation, man-in-the middle and physical capturing of IoT devices etc. To address these concerns in fog computing services, in this paper, a lightweight anonymous authentication and secure communication scheme is proposed and it only used secure one-way hash function and bitwise XOR operations when cloud, fog and user mutually authenticate each other. After the successful authentication, both fog-based participants can agree on a session key to encrypt the subsequent communication messages. The security can be ensured during authentication process by using the Burrows-Abadi-Needham (BAN) logic and the performance comparisons with existing schemes demonstrate that the proposed scheme is secure and highly efficient.

INDEX TERMS Anonymous authentication, fog computing, Internet of Things, session key agreement, security.

I. INTRODUCTION

In recent years, with the rapid development of ICT and intelligent device, the Internet of Things (IoT) has become a hot topic of many experts and scholars and ICT companies, because of its extension of the traditional network communication link between people, the realization of communication between people and objects, or communication between objects and objects [2], [18], [29], [35]. In addition, cloud computing enables network users to access or download intelligent services and applications provided by application service providers or cloud data centers from any place, anytime through personal communication devices with network connectivity, such as tablet computers, smartphones, and mobile devices [10], [21], [22]. However, the limited execution efficiency of the cloud computing environment has resulted in its inability to meet the requirements of many existing intelligent application services, such as the low latency, context awareness, and support for mobility of intelligent applications such as in-vehicle networks and medical augmented reality. In order to meet the above requirements, the concept of Fog Computing was first proposed by Cisco in 2012. It is an extension of traditional cloud computing. Its main purpose is to provide better computing power,

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

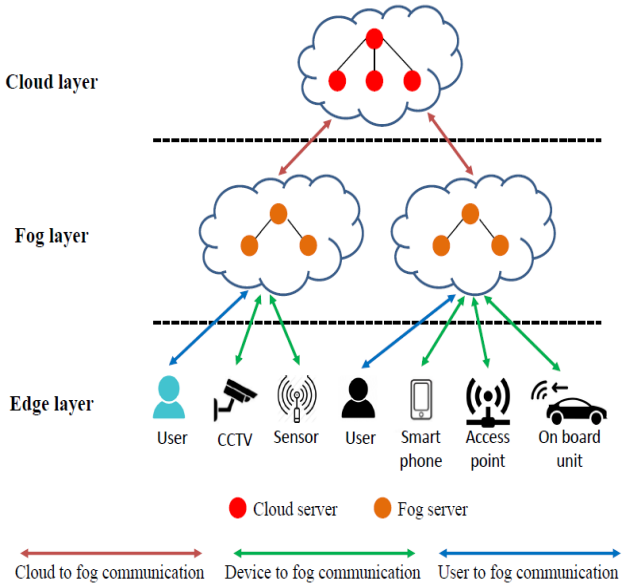

FIGURE 1. Three-layer architecture of fog computing.

storage space, and network services between terminal devices and cloud servers, as well as to reduce communication delay and management control complexity. Through the hierarchical architecture established by fog computing, computing requirements can be layered and processed in different regions, so that the information generated by local devices can be initially analyzed, and the back-end cloud can perform data analysis and calculation for heavy computationally demanding work such as coordination and global analysis. The advantage of this architecture is that it can solve the possible network congestion, speed up the data processing and transmission, and reduce the delay.

The fog computing architecture is generally composed of three different working layers, namely terminal layer, fog layer and cloud layer. The three-layer architecture of fog computing and its detailed description are shown in Fig. 1:

1) Edge layer: this layer is closest to end-users and end-devices and consists of various IoT or intelligent devices, such as sensors, mobile phones, intelligent vehicles, smart cards, and readers. What is special is that although these devices have the capability of computing, we usually only use these devices to carry out intelligent sensing of entity objects or events, and upload the collected sensing data to the upper layer for subsequent processing and storage.
2) Fog layer: this layer is located at the edge of the network and consists of a large number of fog nodes. These fog nodes usually contain routers, gateways, switchers, access points, base stations, and specific fog servers. These fog nodes can be widely distributed between terminal devices and the cloud, such as cafes, shopping centers, bus stops, streets, and parks. Fog nodes can be placed in a fixed position or moved on a mobile vehicle and are linked to terminal devices to provide intelligent services. In addition, they can calculate, transmit, and temporarily store the sensing data they receive, allowing real-time analysis and delay-sensitive applications to be performed within the fog layer. Finally, fog nodes are connected through IP core networks and cloud data centers, and through cooperation with cloud centers, they can obtain more powerful computing and storage capabilities.
3) Cloud layer: cloud layer is composed of multiple servers and storage devices with high performance to provide various intelligent application services, such as smart home, intelligent transportation, smart factory, and intelligent medical care. This layer has powerful computing and storage capabilities to support a wide range of computational analysis and storage of a large number of data. However, unlike the traditional cloud computing architecture, fog computing does not handle all computing and storage through the cloud. According to the demand load principle, some control strategies can be used to effectively manage and schedule the core cloud, so as to improve the utilization rate of cloud resources.

Compared with the traditional cloud computing mode, the main advantage of fog computing is that it is as close as possible to the network edge devices of the client to perform computing, communication, and storage. In this plan, its advantages are summarized and briefly described as follows:

1) Low latency and real-time interaction: the fog node is located at the edge of the network to quickly receive the data generated by the sensors and devices at the local end, and the data is processed and stored by the network edge devices in the local area network. In this way, fog computing can significantly reduce data transmission on the Internet and provide high speed and high-quality localization services, to achieve low latency and meet the needs of real-time interaction. It is especially suitable for delay-sensitive or time-sensitive application services.
2) Bandwidth saving: fog computing performs some computing work, such as data processing, redundancy removing, data filtering, and valuable information extraction at the local end, and only a small part of the data needs to be transferred to the cloud at the back end. For example, in the face recognition system based on fog computing, the fog node only needs to transmit the face identifier to the cloud, while the system based on traditional cloud computing needs to transmit the original face image to the cloud. Therefore, fog computing can effectively reduce network transmission and save bandwidth. In addition, in some application scenarios, decision making can be implemented locally on the fog node, rather than in the cloud on the back end. In this way, fog computing can effectively save bandwidth. With the advent of the era of big data, the advantages brought by this feature will become more and more important.

3) Supporting mobility: in some fog computing situations, various mobile devices, such as smartphones, smart cars, and smartwatches, can act at the terminal layer at will, while some terminal devices, such as traffic cameras, will remain static. Fog nodes in the fog layer can also be mobile or static computing resource platforms, which can be statically deployed in airports, coffee shops, or dynamically deployed on moving vehicles and trains.
4) Geographic distribution and distributed data analysis: compared with traditional centralized cloud computing, fog computing's services and applications are deployed in a geographically distributed manner, consisting of a large number of widely distributed nodes, enabling it to track and infer the location of end devices to support mobility. Unlike centralized data centers, where information is processed and stored far away from the end user, the fog computing environment of the distributed architecture will be as close to the client for data analysis and processing as possible. By the user location-based service model, it can provide users with more powerful real-time decision-making capability.
5) Heterogeneity: generally speaking, fog nodes are deployed in a variety of environments in various forms. They usually come from high-performance servers, edge servers, gateways, access points, base stations, etc. These hardware platforms have different levels of computing and storage capacity, and can run a variety of operating systems and load different software applications. Fog computing is a highly virtualized platform. Some virtual nodes, such as virtual computing nodes and virtual network nodes, can be regarded as fog nodes. Therefore, fog nodes are heterogeneous. In addition, the network infrastructure of fog computing is also heterogeneous, including high-speed connected data centers, many wireless access technologies, such as WLAN, WiFi, 3G/4G, and ZigBee, which are connected to edge devices.
6) Interoperability: because of the heterogeneity of fog computing, fog nodes and terminals are often from different suppliers and deployed in a variety of environments. Fog computing must be interoperable and work with different suppliers to provide a wide range of services in a seamless manner. For example, an intelligent transportation system based on fog computing needs to perform real-time data analysis and provide dynamic traffic information to intelligent vehicles, traffic signals, fog nodes, and fog applications. In order to realize complex cooperation and information sharing, a policy-based resource management scheme must be proposed to ensure that the resources requested by different users can be interoperable and cooperate safely in fog computing.
7) Data security and privacy protection: the host service provided by fog computing is close to the end user, so the data security and privacy protection of the fog computing environment must be ensured. First, data can be protected by encryption and isolation. The fog node provides mechanisms such as access control policies, encryption methods, integrity checks, and isolation measures to protect sensitive data. Secondly, in order to avoid the low efficiency of traditional devices when performing remote updates, fog computing does not need to update the firmware system, but only the algorithm and micro-application at the fog node end.
8) Low energy consumption: in the fog computing architecture, due to the geographical distribution of the fog nodes, it does not generate excessive heat energy and does not need to use an additional cooling system. In addition, short-range communication nodes combined with some energy management strategies can significantly reduce communication energy consumption and save energy consumption, so that fog computing can provide a more environmentally friendly computing situation.

Many computing models have been proposed, such as cloud computing, edge computing, cluster computing, and jungle computing. Their computing tasks have their own advantages in specific scenarios. Edge computing is a computing method that extends cloud computing services to edge devices so that edge devices can perform computing and storage functions and make computing and storage occur at the source of things and data as much as possible. Edge nodes and devices can perform a large number of computing tasks, such as data processing, data staging, device management, decision making, and privacy protection, to reduce network latency and bandwidth congestion between terminal devices and the cloud. These edge nodes can be composed of smart sensors, smart phones, smart vehicles, or even edge servers. They can be linked to each other at the local end to form an edge network. In addition, edge devices can also provide edge intelligence services to nearby users through the connection with cloud data centers, so as to meet the key needs of the digital industry in real-time services, data optimization, application intelligence, security and privacy protection.

In this paper, fog computing, edge computing, cloud computing and other modes are sorted out and summarized as shown in Fig. 2. In terms of latency and mobility, cloud computing has a higher degree of latency than edge computing and fog computing, and the mobility is limited due to the centralized architecture. In terms of bandwidth cost, because the cloud computing model must transmit all the data collected from the sensor layer to the remote cloud server center through the network layer transmission technology, its bandwidth cost is higher than the other two models. In terms of deployment, cloud computing is mostly deployed in the core of the network system, while edge computing will limit the deployment of the edge computing platform to mobile network infrastructure, such as 5G. Fog computing can be deployed anywhere near the edge of the network, such as user-managed servers, access points, routers, and gateways.

In terms of network architecture, the cloud computing model is the centralized control architecture, while edge computing and fog computing can be regarded as the extension of the cloud to supplement its services, so as to realize the creation of n-level distributed network architecture. Edge computing can provide services and decisions autonomously without relying on a central infrastructure, and multiple edge infrastructures can exchange information and services with each other. In terms of computing and storage capacity, the main goal of both edge computing and fog computing is to make the network edge have similar functions to cloud computing, hoping to achieve computing and storage capacity near the end user, reduce service latency and save network bandwidth for delay-sensitive applications. Even though edge computing has the same goal as fog computing, they have some potential differences. For example, in edge computing, edge devices cannot implement multiple IoT applications because limited resources lead to resource contention and increased processing latency. By seamlessly integrating edge devices and cloud resources, fog computing can overcome the limitations of edge computing and avoid the contention of edge resources, and coordinate the geographically distributed network edge devices to balance the utilization rate of cloud resources.

The fog computing environment combines various IoTs sensing components, location services, wireless transmission reading, content services and other technologies, and has spawned many types of fog computing applications. The following is an introduction to the application scope of fog computing:

1) Application in the smart city [6]: The fog computing environment is especially suitable for smart city applications, such as urban disaster notification, through real-time data feedback and reply. In the development of a flood decision support system, the fog node is used to collect real-time data of urban water regimen and give early warning and alarm when there is doubt about the flood.
2) Application in medical care [7]: Fog computing can also be used in medical care. This paper proposes a fall monitoring system named FAST, which is aided by fog computing analysis. By measuring and analyzing the pulse between the edge device (connected to the user's smartphone) and the cloud server, the system can judge whether the user has fallen or other emergency situations at home, so as to provide real-time medical rescue services.
3) Application in intelligent transportation [14]: VANET (Vehicular Ad Hoc Network) ensures transportation efficiency, safety and convenience of driving by exchanging valuable information, and its applications include content sharing (such as advertising and entertainment) and information dissemination services (such as emergency operations such as natural disasters and terrorist attacks). New transportation applications, such as augmented reality and autonomous driving, require complex storage operations and data processing, and therefore require higher-level data storage, computing, and communication capabilities. A program called VFC (Vehicular Fog Computing) was proposed to meet the requirements of the above applications and some special requirements such as mobility, position awareness, and low latency.
4) Application in Fog in IoT and CoT (Cloud of Things) [1]: as different devices generate different types and frequencies of data, CoT combining IoT and cloud computing is proposed to simplify the ever-growing multimedia content and manage other data. In addition, CoT plays a key role in service discovery, resource provision, and ubiquitous access, especially for medical, emergency, and real-time response applications. In addition, when fog computing exists between the cloud and the Internet of Things, its work tasks can include resource management, data pretreatment, data filtering, and security assessment. Therefore, fog computing needs an effective and efficient IoT resource management framework. The application of fog computing IIoT (Industrial Internet of Things) can make the machines, sensors, actuators and gateways on the production site form a fog network to improve production efficiency [36].
5) Application in Smart Grid [31]: energy grid deploys smart meters in all locations of the distribution network to measure real-time status information in energy generation, energy transmission, energy consumption and pricing. A centralized server system called SCADA (Supervisory Control and Data Acquisition) collects and analyzes status information commands to respond to any demand change or emergency and stabilize the grid. After the introduction of fog computing, the smart grid can become a multi-level layered system, allowing the fog layer to interact with the SCADA system, and take charge of the micro-grid and communicate with neighboring fog layers and higher-level fog. The higher the layer, the greater the latency and the wider the geographical coverage.

While the integration of IoT-based smart services into fog computing can play a key role in delivering a wide range of smart application services to deployed smart devices in a more efficient manner, there are still potential security and privacy risks that need to be eliminated. First, the high frequency of data collection may cause great risks to location privacy, allowing attackers to track smart devices. Moreover, the identities of fog nodes and smart devices may also be impersonated by an attacker to transmit malicious data or illegally collect data [11], [15]–[17], [23]. In recent years, many researchers have proposed security and privacy issues in the fog computing environment [5], [9], [12], [25], [26], [28]. Alrawais *et al.* [3] proposed a secure key exchange method between the fog node and the cloud center. Koo and Hur [20] designed a data deduplication method with a privacy protection function, which can effectively manage the ownership of fog computing. Wang *et al.* [32] proposed

	Cloud Computing	Edge Computing	Fog Computing
Latency	High	Low	Low
Mobility	Limited	Supported	Supported
Location awareness	Partially Supported	Supported	Supported
Bandwidth costs	High	Low	Low
Deployment	Network core	Network edge	Near-edge
Hardware	Servers	Heterogeneous servers	Heterogeneous servers
Network architecture	Centralized	N-tier, Decentralized, Distributed	N-tier, Decentralized, Distributed
Computation and storage capabilities	Strong	Weak	Medium
Multiple IoT applications	Supported	Unsupported	Supported

FIGURE 2. Analysis and comparison of three computing modes.

an anonymous and secure aggregation method in the fog computing environment. Data from terminal nodes can be aggregated through the fog nodes, and then the aggregated data is forwarded to the public cloud server. In addition, some methods emphasize the protection of device privacy, but the computing capacity between the smart devices and the fog nodes in the fog computing environment cannot meet their requirements, so they are not applicable to real-time IoT applications. Guan *et al.* [13] and Lin *et al.* [24] proposed a data aggregation method based on blockchain technology. The paradigm of fog computing due to its inherited properties from cloud as inherits its security and privacy concerns such as spoofing, message replay, impersonation, man-in-the middle and physical capturing of IoT devices etc. To erase the various security pitfalls found in existing authentication schemes, existing schemes are not sustainable in fog computing environments, and it motivated us to design a new lightweight anonymous authentication and secure communication scheme that overcomes the drawbacks of existing authentication schemes and ensures both security and efficiency.

The remainder of the paper is organized as follows. Section 2 presents a new security architecture along with the threat model for fog computing services. Section 3 introduces our lightweight anonymous authentication scheme with privacy preserving for fog computing services. We present the security proof of the proposed scheme and evaluate the performance of the proposed authentication scheme with other related fog computing schemes in Section 4 and Section, respectively. Finally we conclude this paper in Section 6.

II. SYSTEM ARCHITECTURE IN FOG COMPUTING SERVICES

In this section, we will illustrate the proposed system architecture for fog computing paradigm, subsequently we define two adversary models to evaluate its security and usability.

A. SYSTEM MODEL

The system architecture used in fog computing services is shown in Fig. 3. In the given architecture, four roles participate in this system: the cloud server (CS), the fog server (FS), the edge user (U) and the edge device (D). When an U and FS (or D and FS) need to interact securely, they must be able to authenticate each others and may need the support of CS. Suppose CS wants to access the real-time data gathered from deployed edge devices, the given model is designed to minimize delay and burden on CS by exploiting the fog layer and the interactions between FS and CS become important since FS can easily gets local overview while the global coverage can be achieved at cloud layer. Therefore, a secure mutual authentication and key agreement mechanism among the deployed CS, FS, U and D is necessary because the communication happens through insecure channel and an adversary can be given an opportunity to threat with the privacy in fog computing services. After executing authentication process, cloud server, fog servers, edge users and edge devices can establish session keys for securing their interactions. There are three types of communication involve in this system: (1) edge user to fog server communication, (2) edge device to fog server communication, and (3) cloud server to fog server communication. The detailed steps of Fig. 3 are described as follows.

Step 1: This step permits CS to fulfill the registration of edge users, edge devices and fog servers before they are deployed in fog computing network.

Step 2: When an edge user wants to access FS and asks a services from FS, U must send a login request to FS. Further, when an edge device D wants to interact with FS and sends gathered data to FS, D must send a login request to FS.

Step 3: For secure interaction, in this step, both the legitimacy of U/D and FS can be verified by CS.

Step 4: If U/D and FS are legal, CS and FS can perform this step to achieve mutual authentication and establish a session key between them.

Step 5: After the successful execution of this step, both U/D, FS and CS can agree on a session key for securing their subsequent communications.

B. THREAT MODEL

According to the system model shown in Fig. 3, edge users and edge devices can communicate with their corresponding fog server, and the fog server forwards the data to its back-end cloud server. In this situation, all communications take place over the public channels and there is always a possibility of security pitfalls during the communication session in fog computing environment. In threat model, this paper will adopt the widely-used Dolev-Yao (DY) threat model and Canetti-Krawczyk (CK) adversary model. According to the definition of DY model, the communication channel between any two parties is open and insecure, and also the end-point parties are not trusted. An adversary can eavesdrop on the messages exchanged on the network, and can also delete or tamper the transmitted messages over public channel. According to the definition of CK model, the mobile device of an U may be lost or stolen, the secret parameters stored in that device can be also extracted by using power analysis attack. Further, an adversary may physical capture some edge device D and obtain the stored credentials in D with the help of complicated power analysis attack. After that, the compromised data will be used to undermine the security of fog computing services such as session key exposure, impersonation attack, replay attack, privacy exposure attack and man-in-the-middle attack etc. Note that CS and FS are trusted entities and they will not be compromised by adversaries.

III. THE PROPOSED SCHEME

In this section, we propose a new lightweight anonymous authentication scheme for fog computing services. The proposed authentication scheme consists of the following seven phases: system initialization, fog server registration, edge user registration, edge device registration, authentication and key agreement of edge user, authentication and key agreement of edge device and biometric update of edge user. The details of the proposed scheme are described in the following subsections. The notations used in the proposed scheme are summarized below in Table 1.

A. SYSTEM INITIALIZATION

The cloud server CS generates a master secret key MK and three long-term secret keys K_{cf}, K_{cu}, and K_{cd} and keeps them secret. CS further chooses a collision free one-way hash function $h(\cdot)$. We assume that CS is fully trusted and also maintains a database to record registered edge users, edge devices and fog servers.

B. FOG SERVER REGISTRATION

The fog server FS_i picks a unique real identity ID_i and registers itself with CS by sending identity ID_i via a secure channel. After receiving ID_i, CS generates a pseudonym TID_i and computes $B_i = h(ID_i||K_{cf})$ and $h(MK||K_{cf})$. CS then responses $\{TID_i, B_i, h(MK||K_{cf})\}$ to FS_i via a secure channel and maintains pseudonym and verifier of FS_i in a protected verifier table as depicted in Table 2. Finally, FS_i stores TID_i, B_i and $h(MK||K_{cf})$ in its memory. Registration process of FS_i is summarized in Fig. 4.

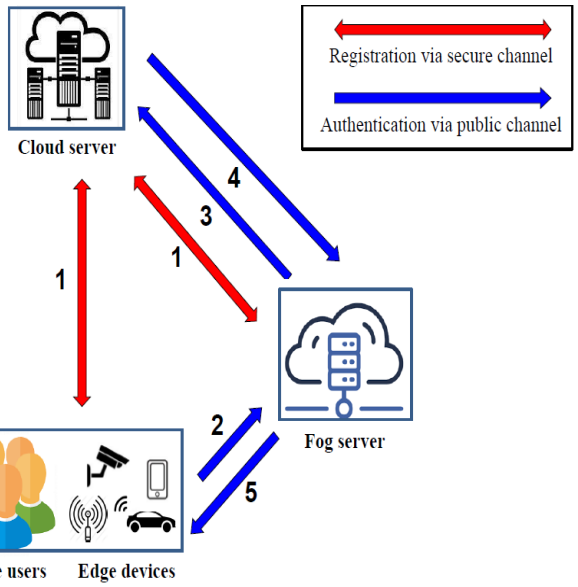

FIGURE 3. The system architecture of fog computing services.

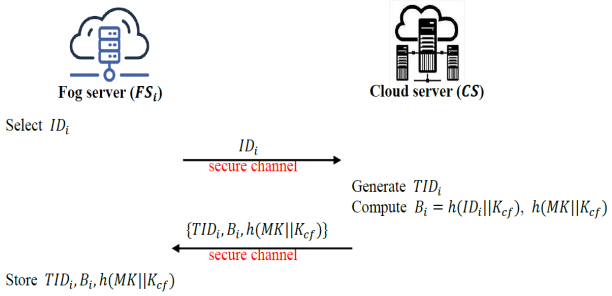

FIGURE 4. Registration process of FS_i.

C. EDGE USER REGISTRATION

The edge user EU_j picks a unique real identity ID_j and inputs his/her biometric BIO_j into his/her smart device. EU_j's smart device generates a 160-bit random secret number n_u and computes $A_j = h(ID_j||BIO_j||n_u)$. Then EU_j's smart device sends the registration request A_j along with the identity ID_i to CS through a secure channel. After receiving ID_j and A_j, CS generates a pseudonym TID_j and computes $B_j = h(ID_j||MK)$, $C_j = h(ID_j||A_j||h(K_{cu}))$ and $D_j = B_j \oplus h(MK||K_{cu}) \oplus A_j$. CS then responses $\{TID_j, C_j, D_j, h(\cdot), h(K_{cu})\}$ to EU_j through a secure channel and maintains pseudonym and verifier of EU_j in a protected verifier table as depicted in Table 3. Finally, EU_j's smart device stores TID_j, C_j, D_j, $h(\cdot)$, $h(K_{cu})$ and n_u in its memory. Registration process of EU_j is summarized in Fig. 5.

TABLE 1. Notations used in the paper.

Symbol	Definition
CS	The cloud server
FS_i	The ith fog server
EU_j	The jth edge user
ED_k	The kth edge device
ID_i, ID_j, ID_k	The identity of FS_i, EU_j and ED_k, respectively
BIO_j	The biometric of EU_j such as Face ID or Fingerprint ID
TID_i, TID_j, TID_k	The pseudo identity of FS_i, EU_j and ED_k, respectively
K_{cf}, K_{cu}, K_{cd}	The long-term secret keys chosen by CS
MK	The master secret key of CS
n_u, n_d	160-bit random secret number of EU_j and ED_k, respectively
r_c, r_f, r_u, r_d	128-bit random number of CS, FS_i, EU_j and ED_k, respectively
$h(\cdot)$	A collision free one-way hash function
$\|\|, \oplus$	Concatenation and bitwise XOR operations, respectively
SK_{ij}	Session key shared between CS, FS_i and EU_j
SK_{ik}	Session key shared between CS, FS_i and ED_k

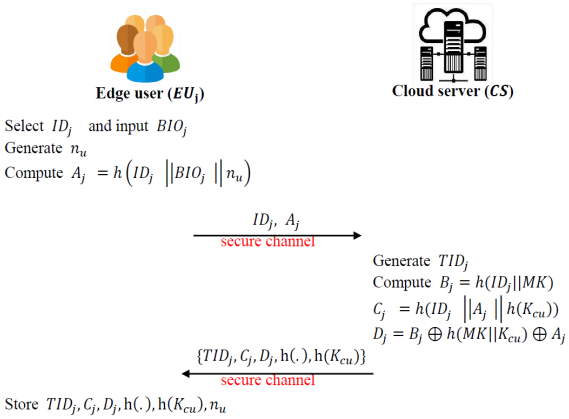

FIGURE 5. Registration process of EU_j.

TABLE 2. The verifier table of FS_i after finishing fog server registration.

Pseudonym of FS_i	Identity of FS_i	Verifier of FS_i
$\vdots$	$\vdots$	$\vdots$
TID_i	ID_i	$B_i = h(ID_i\|\|K_{cf})$
$\vdots$	$\vdots$	$\vdots$

D. EDGE DEVICE REGISTRATION

The edge device ED_k picks a unique real identity ID_k and registers itself with CS by sending identity ID_k via a secure channel. After receiving ID_k, CS generates a pseudonym TID_k and computes $B_k = h(ID_k||K_{cd})$ and $h(MK||K_{cd})$. CS then responses $\{TID_k, ID_i, B_k, h(MK||K_{cd})\}$ to ED_k via a secure channel and maintains pseudonym and verifier of ED_k in a protected verifier table as depicted in Table 4. Note that each ED_k will be deployed in the designated area and assigned a specific FS_i to it, where ID_i is the identity of designated FS_i of ED_k. Finally, ED_k stores TID_k, ID_i, B_k and $h(MK||K_{cd})$ in its memory. Registration process of ED_k is summarized in Fig. 6.

E. AUTHENTICATION AND KEY AGREEMENT OF EDGE USER

In this phase, we assume that an edge user EU_j wants to access the fog server FS_i and asks a service from system. In order to

TABLE 3. The verifier table of EU_j after finishing edge user registration.

Pseudonym of EU_j	Verifier of EU_j
$\vdots$	$\vdots$
TID_j	$B_j = h(ID_j\|\|MK)$
$\vdots$	$\vdots$

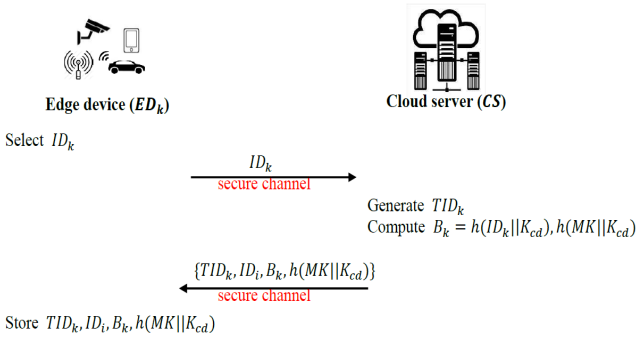

FIGURE 6. Registration process of ED_k.

preserve privacy of data transmitted through pubic channels, the cloud server CS can help EU_j and FS_i to authenticate each other and establish a session key SK_{ij} between them by performing following steps. The detailed steps of this phase are depicted in Fig. 7.

Step 1: EU_j first inputs ID_j' and BIO_j' into his/her smart device. Then, smart device retrieves n_u and $h(K_{cu})$ to compute $A_j' = h(ID_j'||BIO_j'||n_u)$ and $C_j' = h(ID_j'||A_j'||h(K_{cu}))$ and checks whether $C_j' = C_j$, where C_j is retrieved from its memory. If it is not true, the smart device rejects the request and terminates. Otherwise, it means EU_j is a legal user and the smart device randomly selects a 128-bit random number r_u and computes $E_j = D_j \oplus A_j' \oplus r_u$ and $F_j = h(h(K_{cu})||ID_i||r_u))$, where D_j is retrieved from its memory. Finally, the smart devices retrieves the pseudonym TID_j from its memory and sends the access request $M_{u1} = \{TID_j, E_j, F_j\}$ to FS_i through a public channel.

TABLE 4. The verifier table of ED_k after finishing edge device registration.

Pseudonym of ED_k	Identity of designated FS_i	Verifier of ED_k
TID_k	ID_i	$B_k = h(ID_k \| \| K_{cd})$

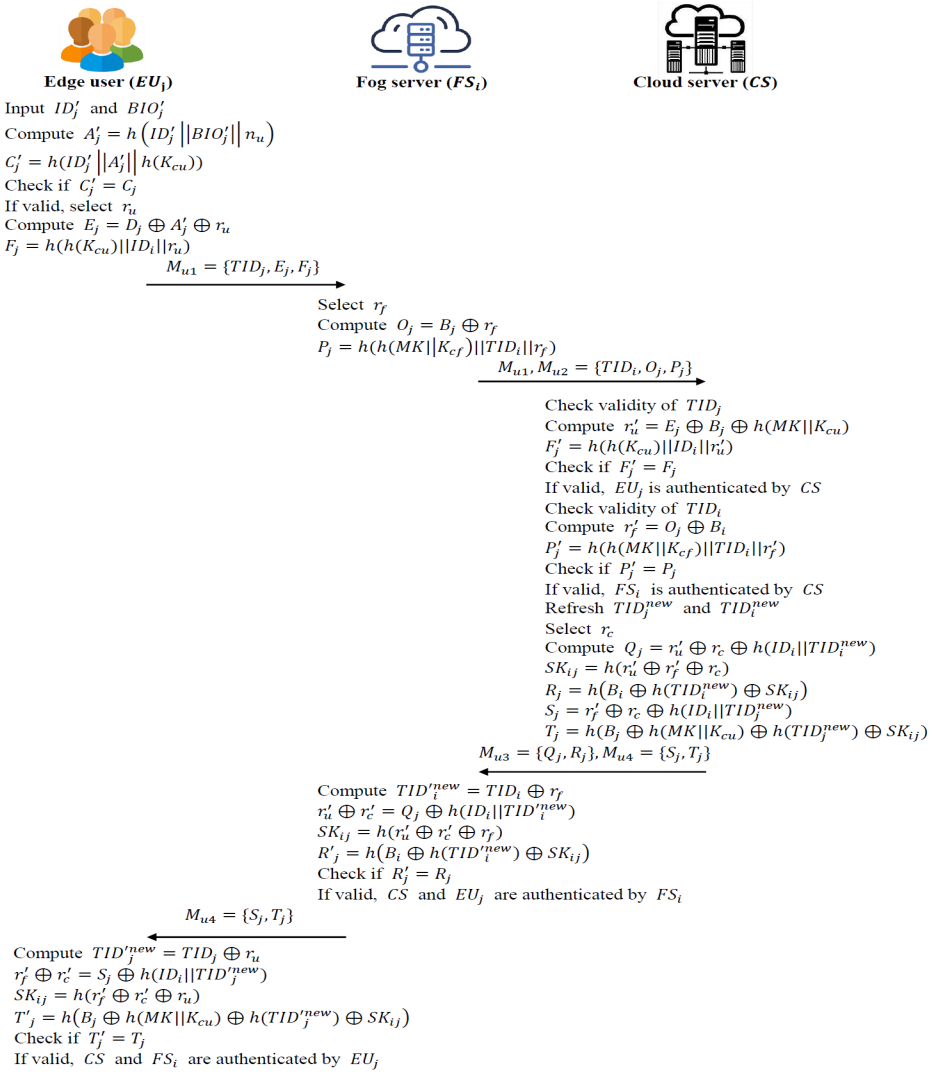

FIGURE 7. Authentication and key agreement phase of EU_j in fog computing services.

Step 2: Upon receiving M_{u1}, FS_i has no information about EU_j, hence it randomly selects a 128-bit random number r_f and retrieves TID_i, B_i and $h(MK||K_{cf})$ to compute $O_j = B_i \oplus r_f$ and $P_j = h(h(MK||K_{cf})||TID_i||r_f)$. Finally, FS_i forwards the authentication request M_{u1} and $M_{u2} = \{TID_i, O_j, P_j\}$ to CS through a public channel.

Step 3: Upon receiving M_{u1} and M_{u2}, CS inspects M_{u1} and searches the verifier table of EU_j in its database to find entry that match TID_j. If there is no matching entry, CS rejects the request and terminates session. Otherwise, CS retrieves B_j and $h(MK||K_{cu})$ to compute $r'_u = E_j \oplus B_j \oplus h(MK||K_{cu})$ and uses the derived r'_u to compute $F'_j = h(h(K_{cu})||ID_i||r'_u))$. Then CS checks whether $F'_j = F_j$. If it is not true, CS terminates the session. Otherwise, the legitimacy of EU_j is authenticated by CS and CS goes to next step.

Step 4: In this step, CS further inspects M_{u2} and searches the verifier table of FS_i in its database to find entry that match TID_i. If there is no matching entry, CS rejects the request and terminates session. Otherwise, CS retrieves B_i and $h(MK||K_{cf})$ to compute $r'_f = O_j \oplus B_i$ and uses the derived r'_f to compute

$P'_j = h(h(MK||K_{cf})||TID_i||r'_f)$. Then CS checks whether $P'_j = P_j$. If it is not true, CS terminates the session. Otherwise, the legitimacy of FS_i is authenticated by CS and CS goes to next step.

Step 5: After verifying the validity of EU_j and FS_i, CS refreshes a new pseudonym TID_j^{new} for EU_j by computing $TID_j^{new} = TID_j \oplus r'_u$ and replaces TID_j with TID_j^{new} in the verifier table of EU_j. In addition, CS refreshes a new pseudonym TID_i^{new} for FS_i by computing $TID_i^{new} = TID_i \oplus r'_f$ and replaces TID_i with TID_i^{new} in the verifier table of FS_i. CS further selects a 128-bit random number r_c and uses the derived r'_u, r'_f, TID_i^{new} and TID_j^{new} to compute $Q_j = r'_u \oplus r_c \oplus h(ID_i||TID_i^{new})$, $SK_{ij} = h(r'_u \oplus r'_f \oplus r_c)$, $R_j = h(B_i \oplus h(TID_i^{new}) \oplus SK_{ij})$, $S_j = r'_f \oplus r_c \oplus h(ID_i||TID_j^{new})$ and $T_j = h(B_j \oplus h(MK||K_{cu}) \oplus h(TID_j^{new}) \oplus SK_{ij})$, where SK_{ij} is the session key established with FS_i, EU_j and CS. Finally, CS sends $M_{u3} = \{Q_j, R_j\}$ and $M_{u4} = \{S_j, T_j\}$ to FS_i.

Step 6: Upon receiving M_{u3} and M_{u4}, FS_i first inspects M_{u3} and uses original TID_i and r_f to compute $TID_i'^{new} = TID_i \oplus r_f$. Then FS_i uses its identity ID_i and $TID_i'^{new}$ to compute $r'_u \oplus r'_c = Q_j \oplus h(ID_i||TID_i'^{new})$, $SK_{ij} = h(r'_u \oplus r'_c \oplus r_f)$ and $R'_j = h(B_i \oplus h(TID_i'^{new}) \oplus SK_{ij})$. If $R'_j = R_j$, FS_i believes that CS and EU_j are legal parties and stores the shared session key SK_{ij} for future secure communication. Otherwise, FS_i terminates the session. Finally, FS_i forwards M_{u4} to EU_j.

Step 7: Upon receiving M_{u4}, EU_j first inspects M_{u4} and uses original TID_j and r_u to compute $TID_j'^{new} = TID_j \oplus r_u$. Then EU_j uses FS_i's identity ID_i and $TID_j'^{new}$ to compute $r'_f \oplus r'_c = S_j \oplus h(ID_i||TID_j'^{new})$, $SK_{ij} = h(r'_f \oplus r'_c \oplus r_u)$ and $T'_j = h(B_j \oplus h(MK||K_{cu}) \oplus h(TID_j'^{new}) \oplus SK_{ij})$. If $T'_j = T_j$, EU_j believes that CS and FS_i are legal parties and stores the shared session key SK_{ij} for future secure communication. Otherwise, EU_j terminates the session.

F. AUTHENTICATION AND KEY AGREEMENT OF EDGE DEVICE

In this phase, we assume that an edge device ED_k is deployed in designated environment and is ready to send the gathered data to its corresponding fog server FS_i. In order to ensure the integrity of the sensitive data gathered from ED_k, the cloud server CS can help ED_k and FS_i to authenticate each other and establish a session key SK_{ik} between them by performing following steps. The detailed steps of this phase are depicted in Fig. 8.

Step 1: ED_k first randomly selects a 128-bit random number r_d and retrieves ID_i, TID_k, B_k and $h(MK||K_{cd})$ from its memory to compute $E_k = B_k \oplus r_d$ and $F_k = h(h(MK||K_{cd})||ID_i||r_d))$. Then ED_k sends the access request $M_{d1} = \{TID_k, E_k, F_k\}$ to FS_i through a public channel.

Step 2: Upon receiving M_{d1}, FS_i has no information about ED_k, hence it randomly selects a 128-bit random number r_f and retrieves TID_i, B_i and $h(MK||K_{cf})$ to compute $O_k = B_i \oplus r_f$ and $P_k = h(h(MK||K_{cf})||TID_i||r_f)$. Finally, FS_i forwards the authentication request M_{d1} and $M_{d2} = \{TID_i, O_k, P_k\}$ to CS through a public channel.

Step 3: Upon receiving M_{d1} and M_{d2}, CS inspects M_{d1} and searches the verifier table of ED_k in its database to find entry that match TID_k. If there is no matching entry, CS rejects the request and terminates session. Otherwise, CS retrieves B_k and $h(MK||K_{cd})$ to compute $r'_d = E_k \oplus B_k$ and uses the derived r'_u to compute $F'_k = h(h(MK||K_{cd})||TID_k||r'_d))$. Then CS checks whether $F'_j = F_j$. If it is not true, CS terminates the session. Otherwise, the legitimacy of ED_k is authenticated by CS and CS goes to next step.

Step 4: In this step, CS further inspects M_{d2} and searches the verifier table of FS_i in its database to find entry that match TID_i. If there is no matching entry, CS rejects the request and terminates session. Otherwise, CS retrieves B_i and $h(MK||K_{cf})$ to compute $r'_f = O_k \oplus B_i$ and uses the derived r'_f to compute $P'_k = h(h(MK||K_{cf})||TID_i||r'_f)$. Then CS checks whether $P'_k = P_k$. If it is not true, CS terminates the session. Otherwise, the legitimacy of FS_i is authenticated by CS and CS goes to next step.

Step 5: After verifying the validity of ED_k and FS_i, CS refreshes a new pseudonym TID_k^{new} for ED_k by computing $TID_k^{new} = TID_k \oplus r'_d$ and replaces TID_k with TID_k^{new} in the verifier table of ED_k. In addition, CS refreshes a new pseudonym TID_i^{new} for FS_i by computing $TID_i^{new} = TID_i \oplus r'_f$ and replaces TID_i with TID_i^{new} in the verifier table of FS_i. CS further selects a 128-bit random number r_c and uses the derived r'_d, r'_f, TID_i^{new} and TID_k^{new} to compute $Q_k = r'_d \oplus r_c \oplus h(ID_i||TID_i^{new})$, $SK_{ik} = h(r'_d \oplus r'_f \oplus r_c)$, $R_k = h(B_i \oplus h(TID_i^{new}) \oplus SK_{ik})$, $S_k = r'_f \oplus r_c \oplus h(ID_i||TID_k^{new})$ and $T_k = h(B_k \oplus h(MK||K_{cd}) \oplus h(TID_k^{new}) \oplus SK_{ik})$, where SK_{ik} is the session key established with FS_i, ED_k and CS. Finally, CS sends $M_{d3} = \{Q_k, R_k\}$ and $M_{d4} = \{S_k, T_k\}$ to FS_i.

Step 6: Upon receiving M_{d3} and M_{d4}, FS_i first inspects M_{d3} and uses original TID_i and r_f to compute $TID_i'^{new} = TID_i \oplus r_f$. Then FS_i uses its identity ID_i and $TID_i'^{new}$ to compute $r'_d \oplus r'_c = Q_k \oplus h(ID_i||TID_i'^{new})$, $SK_{ik} = h(r'_d \oplus r'_c \oplus r_f)$ and $R'_k = h(B_i \oplus h(TID_i'^{new}) \oplus SK_{ik})$. If $R'_k = R_k$, FS_i believes that CS and ED_k are legal parties and stores the shared session key SK_{ik} for future secure communication. Otherwise, FS_i terminates the session. Finally, FS_i forwards M_{d4} to ED_k.

Step 7: Upon receiving M_{d4}, ED_k first inspects M_{d4} and uses original TID_k and r_d to compute

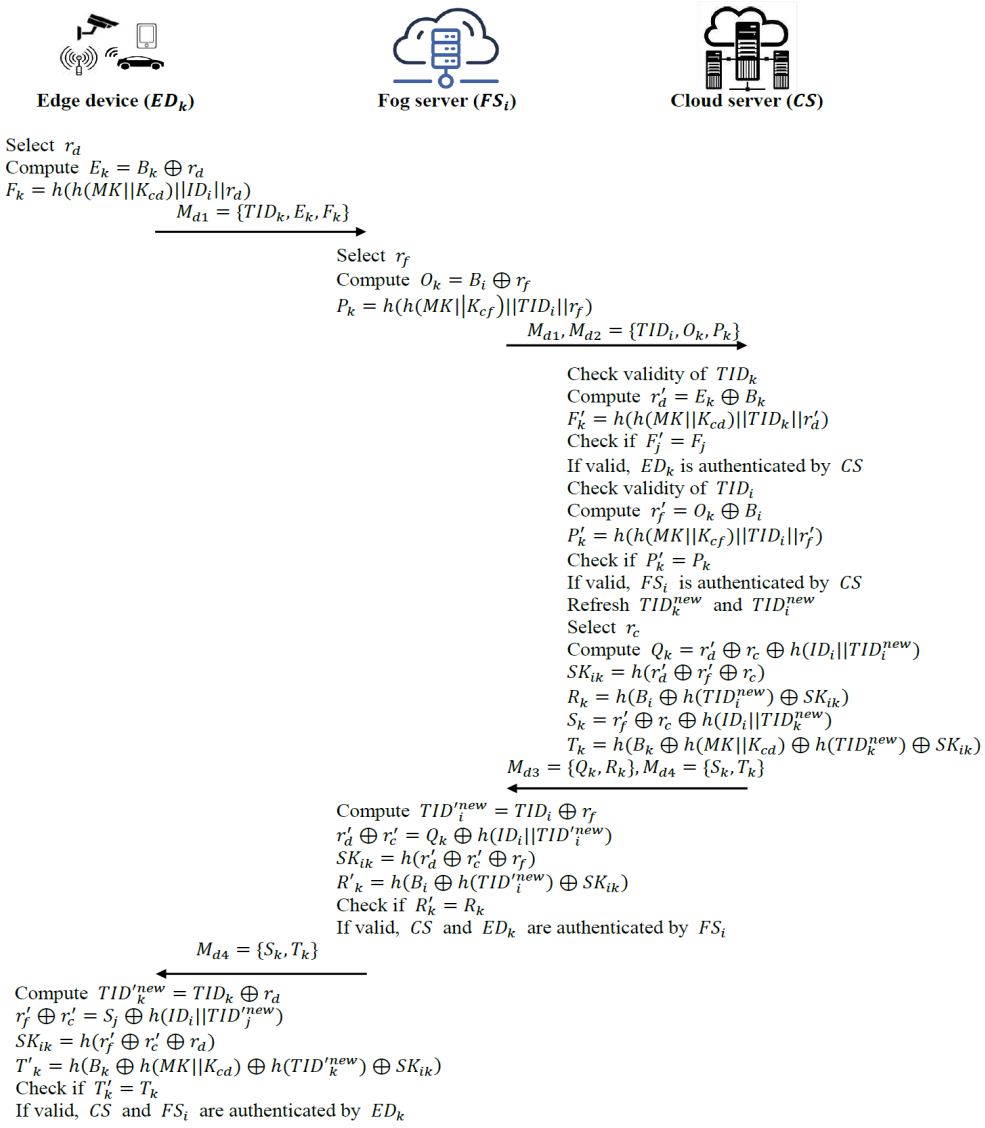

FIGURE 8. Authentication and key agreement phase of ED_k in fog computing services.

$TID_k^{new} = TID_k \oplus r_d$. Then ED_k uses FS_i's identity ID_i and TID_k^{new} to compute $r'_f \oplus r'_c = S_k \oplus h(ID_i||TID_k^{new})$, $SK_{ik} = h(r'_f \oplus r'_c \oplus r_d)$ and $T'_k = h(B_k \oplus h(MK||K_{cd}) \oplus h(TID_k^{new}) \oplus SK_{ik})$. If $T'_k = T_k$, ED_k believes that CS and FS_i are legal parties and stores the shared session key SK_{ik} for future secure communication. Otherwise, ED_k terminates the session.

G. BIOMETRIC UPDATE OF EDGE USER

In the proposed scheme, an edge user EU_j can freely update his/her biometric BIO_j with a new biometric BIO_j^{new} without interaction with cloud server CS. EU_j first inputs the identity ID'_j and original BIO'_j into his/her smart device. Then, smart device retrieves n_u and $h(K_{cu})$ to compute $A'_j = h(ID'_j||BIO'_j||n_u)$ and $C'_j = h(ID'_j||A'_j||h(K_{cu}))$ and checks whether $C'_j = C_j$, where C_j is retrieved from its memory. If it is not true, the smart device denies the update request and terminates. Otherwise, the smart device asks EU_j to input his/her new biometric BIO_j^{new} and computes $A_j^{new} = h(ID'_j||BIO_j^{new}||n_u)$, $C_j^{new} = h(ID'_j||A_j^{new}||h(K_{cu}))$ and $D_j^{new} = D_j \oplus A_j \oplus A_j^{new}$. Finally, the smart devices replaces original (C_j, D_j) with new (C_j^{new}, D_j^{new}) in its memory and ends this phase.

IV. SECURITY PROOF OF THE PROPOSED SCHEME

A. BAN LOGIC PROOF

In this section, we use the BAN logic [4] to analyze the security of the session key between node A and node B. Some notations used in BAN logic analysis are described as follows:

- $A \mid \equiv X$: A believes X or A would be entitled to believe X.
- $A \lhd X$: A sees X. Someone has sent a message containing X to A, who can read and repeat X.

- $A \mid \Rightarrow X$: A has jurisdiction over X. A is an authority on X and should be trusted on this matter.
- $A \mid \sim X$: A once said X. A at some time sent a message including X.
- $< X >_Y$: This represents X combined with Y.
- $\sharp(X)$: The formula X is fresh, that is, X has not been sent in a message at any time before the current run of the protocol.
- $A \overset{K}{\longleftrightarrow} B$: A and B may use the shared key K to communicate.
- $A \overset{S}{\Longleftrightarrow} B$: The formula S is a secret known only to A and B and possibly to principals trusted by them.

In the authentication and key agreement of the edge user phase of the proposed scheme, the main goal of the scheme is to analyze the session key establishment between the edge user EU and the fog server FS, with the help of the cloud server CS.

G1: $EU \mid \equiv EU \overset{SK_{ij}}{\longleftrightarrow} FS$

G2: $EU \mid \equiv FS \mid \equiv EU \overset{SK_{ij}}{\longleftrightarrow} FS$

G3: $FS \mid \equiv EU \overset{SK_{ij}}{\longleftrightarrow} FS$

G4: $FS \mid \equiv EU \mid \equiv EU \overset{SK_{ij}}{\longleftrightarrow} FS$

G5: $EU \mid \equiv TID_i$

G6: $EU \mid \equiv FS \mid \equiv TID_i$

G7: $FS \mid \equiv TID_j$

G8: $FS \mid \equiv EU \mid \equiv TID_j$

According to the authentication and key agreement of the edge user phase, BAN logic is used to produce an idealized form as follows:

M1: $(< TID_j, B_j, r_u >_{K_{cu}}, < H(SK_{ij}, TID_j, r_u) >_{r_u \oplus r_c})$

M2: $(< TID_i, B_i, r_f >_{K_{cf}}, < H(SK_{ij}, TID_i, r_f) >_{r_f \oplus r_c})$

To analyze the proposed scheme, the following assumptions are made:

A1: $EU \mid \equiv \sharp(r_u)$

A2: $FS \mid \equiv \sharp(r_u)$

A3: $EU \mid \equiv \sharp(r_f)$

A4: $FS \mid \equiv \sharp(r_f)$

A5: $EU \mid \equiv FS \mid \Rightarrow EU \overset{SK_{ij}}{\longleftrightarrow} FS$

A6: $FS \mid \equiv EU \mid \Rightarrow EU \overset{SK_{ij}}{\longleftrightarrow} FS$

A7: $EU \mid \equiv FS \mid \Rightarrow TID_i$

A8: $FS \mid \equiv EU \mid \Rightarrow TID_j$

According to these assumptions and rules of BAN logic, the main proof of the authentication and key agreement of the edge user phase is as follows:

The fog server FS authenticates the edge user EU, with the help of the cloud server CS. According to **M1** and the *seeing rule*, we could obtain:

S1: $FS \triangleleft (< TID_j, B_j, r_u >_{K_{cu}}, < H(SK_{ij}, TID_j, r_u) >_{r_u \oplus r_c})$

According to **A2** and the *freshness rule*, we could obtain:

S2: $FS \mid \equiv \sharp(< TID_j, B_j, r_u >_{K_{cu}}, < H(SK_{ij}, TID_j, r_u) >_{r_u \oplus r_c})$

According to **S1**, **A4** and the *message meaning rule*, we could obtain:

S3: $FS \mid \equiv EU \mid \sim (< TID_j, B_j, r_u >_{K_{cu}}, < H(SK_{ij}, TID_j, r_u) >_{r_u \oplus r_c})$

According to **S2**, **S3**, and the *nonce verification rule*, we could obtain:

S4: $FS \mid \equiv EU \mid \equiv (< TID_j, B_j, r_u >_{K_{cu}}, < H(SK_{ij}, TID_j, r_u) >_{r_u \oplus r_c})$

According to **S4** and the *belief rule*, we could obtain:

S5: $FS \mid \equiv EU \mid \equiv EU \overset{SK_{ij}}{\longleftrightarrow} FS$

According to **S5**, **A6** and the *jurisdiction rule*, we could obtain:

S6: $FS \mid \equiv EU \overset{SK_{ij}}{\longleftrightarrow} FS$

According to **S6** and the *belief rule*, we could obtain:

S7: $FS \mid \equiv EU \mid \equiv TID_j$

According to **S7**, **A8** and the *jurisdiction rule*, we could obtain:

S8: $FS \mid \equiv TID_j$

The edge user EU authenticates the fog server FS, with the help of the cloud server CS. According to **M2** and the *seeing rule*, we could obtain:

S9: $EU \triangleleft (< TID_i, B_i, r_f >_{K_{cf}}, < H(SK_{ij}, TID_i, r_f) >_{r_f \oplus r_c})$

According to **A1** and the *freshness rule*, we could obtain:

S10: $EU \mid \equiv \sharp(< TID_i, B_i, r_f >_{K_{cf}}, < H(SK_{ij}, TID_i, r_f) >_{r_f \oplus r_c})$

According to **S9**, **A3** and the *message meaning rule*, we could obtain:

S11: $EU \mid \equiv FS \mid \sim (< TID_i, B_i, r_f >_{K_{cf}}, < H(SK_{ij}, TID_i, r_f) >_{r_f \oplus r_c})$

According to **S10**, **S11**, and the *nonce verification rule*, we could obtain:

S12: $EU \mid \equiv FS \mid \equiv (< TID_i, B_i, r_f >_{K_{cf}}, < H(SK_{ij}, TID_i, r_f) >_{r_f \oplus r_c})$

According to **S12** and the *belief rule*, we could obtain:

S13: $EU \mid \equiv FS \mid \equiv EU \overset{SK_{ij}}{\longleftrightarrow} FS$

According to **S13**, **A5** and the *jurisdiction rule*, we could obtain:

S14: $EU \mid \equiv EU \overset{SK_{ij}}{\longleftrightarrow} FS$

According to **S14** and the *belief rule*, we could obtain:

S15: $EU \mid \equiv FS \mid \equiv TID_i$

According to **S15**, **A7** and the *jurisdiction rule*, we could obtain:

S16: $EU \mid \equiv TID_i$

According to **S6**, **S8**, **S14** and **S16**, it can be proved the edge user EU and the fog server FS authenticate each other

with the help of the cloud server CS. Moreover, it can also be proved that the proposed scheme can establish a session key SK_{ij} between EU and FS with the help of CS. The authentication and key agreement of the edge user phase of the proposed scheme thus guarantee the security of the session key between EU and FS.

In the authentication and key agreement of the edge device phase of the proposed scheme, the main goal of the scheme is to analyze the session key establishment between the edge device ED and the fog server FS, with the help of the cloud server CS.

G9: $ED \mid \equiv ED \overset{SK_{ik}}{\longleftrightarrow} FS$

G10: $ED \mid \equiv FS \mid \equiv ED \overset{SK_{ik}}{\longleftrightarrow} FS$

G11: $FS \mid \equiv ED \overset{SK_{ik}}{\longleftrightarrow} FS$

G12: $FS \mid \equiv ED \mid \equiv ED \overset{SK_{ik}}{\longleftrightarrow} FS$

G13: $ED \mid \equiv TID_i$

G14: $ED \mid \equiv FS \mid \equiv TID_i$

G15: $FS \mid \equiv TID_k$

G16: $FS \mid \equiv ED \mid \equiv TID_k$

According to the authentication and key agreement of the edge device phase, BAN logic is used to produce an idealized form as follows:

M3: $(< TID_k, B_k, r_d >_{K_{cd}}, < H(SK_{ik}, TID_k, r_d) >_{r_d \oplus r_c})$

M4: $(< TID_i, B_i, r_f >_{K_{cf}}, < H(SK_{ik}, TID_i, r_f) >_{r_f \oplus r_c})$

To analyze the proposed scheme, the following assumptions are made:

A9: $ED \mid \equiv \sharp(r_d)$

A10: $FS \mid \equiv \sharp(r_d)$

A11: $ED \mid \equiv \sharp(r_f)$

A12: $FS \mid \equiv \sharp(r_f)$

A13: $ED \mid \equiv FS \mid \Rightarrow ED \overset{SK_{ik}}{\longleftrightarrow} FS$

A14: $FS \mid \equiv ED \mid \Rightarrow ED \overset{SK_{ik}}{\longleftrightarrow} FS$

A15: $ED \mid \equiv FS \mid \Rightarrow TID_i$

A16: $FS \mid \equiv ED \mid \Rightarrow TID_k$

According to these assumptions and rules of BAN logic, the main proof of the authentication and key agreement of the edge device phase is as follows:

The fog server FS authenticates the edge device ED, with the help of the cloud server CS. According to **M3** and the *seeing rule*, we could obtain:

S17: $FS \triangleleft (< TID_k, B_k, r_d >_{K_{cd}}, < H(SK_{ik}, TID_k, r_d) >_{r_d \oplus r_c})$

According to **A10** and the *freshness rule*, we could obtain:

S18: $FS \mid \equiv \sharp(< TID_k, B_k, r_d >_{K_{cd}}, < H(SK_{ik}, TID_k, r_d) >_{r_d \oplus r_c})$

According to **S17**, **A10** and the *message meaning rule*, we could obtain:

S19: $FS \mid \equiv ED \mid \sim (< TID_k, B_k, r_d >_{K_{cd}}, < H(SK_{ik}, TID_k, r_d) >_{r_d \oplus r_c})$

According to **S18**, **S19**, and the *nonce verification rule*, we could obtain:

S20: $FS \mid \equiv ED \mid \equiv (< TID_k, B_k, r_d >_{K_{cd}}, < H(SK_{ik}, TID_k, r_d) >_{r_d \oplus r_c})$

According to **S20** and the *belief rule*, we could obtain:

S21: $FS \mid \equiv ED \mid \equiv ED \overset{SK_{ik}}{\longleftrightarrow} FS$

According to **S21**, **A14** and the *jurisdiction rule*, we could obtain:

S22: $FS \mid \equiv ED \overset{SK_{ik}}{\longleftrightarrow} FS$

According to **S22** and the *belief rule*, we could obtain:

S23: $FS \mid \equiv ED \mid \equiv TID_k$

According to **S23**, **A16** and the *jurisdiction rule*, we could obtain:

S24: $FS \mid \equiv TID_k$

The edge device ED authenticates the fog server FS with the help of the cloud server CS. According to **M4** and the *seeing rule*, we could obtain:

S25: $ED \triangleleft (< TID_i, B_i, r_f >_{K_{cf}}, < H(SK_{ik}, TID_i, r_f) >_{r_f \oplus r_c})$

According to **A9** and the *freshness rule*, we could obtain:

S26: $ED \mid \equiv \sharp(< TID_i, B_i, r_f >_{K_{cf}}, < H(SK_{ik}, TID_i, r_f) >_{r_f \oplus r_c})$

According to **S25**, **A11** and the *message meaning rule*, we could obtain:

S27: $ED \mid \equiv FS \mid \sim (< TID_i, B_i, r_f >_{K_{cf}}, < H(SK_{ik}, TID_i, r_f) >_{r_f \oplus r_c})$

According to **S26**, **S27**, and the *nonce verification rule*, we could obtain:

S28: $ED \mid \equiv FS \mid \equiv (< TID_i, B_i, r_f >_{K_{cf}}, < H(SK_{ik}, TID_i, r_f) >_{r_f \oplus r_c})$

According to **S28** and the *belief rule*, we could obtain:

S29: $ED \mid \equiv FS \mid \equiv ED \overset{SK_{ik}}{\longleftrightarrow} FS$

According to **S29**, **A13** and the *jurisdiction rule*, we could obtain:

S30: $ED \mid \equiv ED \overset{SK_{ik}}{\longleftrightarrow} FS$

According to **S30** and the *belief rule*, we could obtain:

S31: $ED \mid \equiv FS \mid \equiv TID_i$

According to **S31**, **A15** and the *jurisdiction rule*, we could obtain:

S32: $ED \mid \equiv TID_i$

According to **S22**, **S24**, **S30** and **S32**, it can be proved that, in the proposed scheme, the edge device ED and the fog server FS authenticate each other with the help of the cloud server CS. Moreover, it can also be proved that the proposed scheme can establish a session key SK_{ik} between ED and FS with the help of CS. The authentication and key agreement of the edge device phase of the proposed scheme thus guarantee the security of the session key between ED and FS.

Scenario: A malicious attacker uses an illegal fog server to get the message from a legal edge user or a legal edge device.

Analysis: The attacker will not succeed because the illegal fog server has not been registered to the cloud server and thus cannot establish a session key with a legal edge user or a legal edge device. We assume the following situation that a legal edge user generates the message $F_j = h(h(K_{cu})||ID_i||r_u)$, then sends the message with legal TID_j to an illegal fog server. The illegal fog server has no information to calculate the message F_j. Thus, the illegal fog server generates the message M_{u2} and sends these messages to the cloud server. The cloud server computes $F'_j = h(h(K_{cu})||ID_i||r'_u)$ and checks whether $F'_j = F_j$. After that, the cloud server checks the correctness of the message M_{u2}. Since the illegal fog server has not been registered to the cloud server, the attacker cannot send the correct M_{u2}, the verification will fail and the cloud server will not give any response. In the same scenario, a legal edge device generates the message $F_k = h(h(MK||K_{cd})||TID_k||r_d)$, then sends the message with legal TID_k to an illegal fog server. The illegal fog server has no information to calculate the message F_k. Thus, the illegal fog server generates the message M_{d2} and sends these messages to the cloud server. The cloud server computes $F'_k = h(h(MK||K_{cd})||TID_k||r'_d)$ and checks whether $F'_k = F_k$. After that, the cloud server checks the correctness of the message M_{d2}. Since the illegal fog server has not been registered to the cloud server, the attacker cannot send the correct M_{d2}, the verification will fail and the cloud server will not give any response. Therefore, the attack will fail when the malicious attacker uses an illegal fog server to get the message from a legal edge user or a legal edge device.

B. RESISTANCE TO IMPERSONATION ATTACK

If an attacker pretends to be a legal edge user or edge device and tries to communicate with the fog server and cloud server, this is an impersonation attack. In our proposed scheme, the cloud server will verify the legitimacy of the edge user or edge device, so the impersonation attack will not be achieved.

Scenario: A malicious attacker pretends to be a legal edge user or edge device and tries to communicate with the fog server and cloud server. The purpose of the attacker is to establish a session key with the fog server.

Analysis: The attacker pretends to be a legal edge user and generates the message $F_j = h(h(K_{cu})||ID_i||r_u)$, then sends the message with legal TID_j to a legal fog server. The legal fog server has no information to check the correctness of the message F_j. Thus, the legal fog server generates the message M_{u2} and sends these messages to the cloud server. The cloud server computes $F'_j = h(h(K_{cu})||ID_i||r'_u)$ and checks whether $F'_j = F_j$. Since the attacker does not know the correct r_u, the correct message F_j cannot be generated. In the same scenario, the attacker pretends to be a legal edge device and generates the message $F_k = h(h(MK||K_{cd})||TID_k||r_d)$, then sends the message with legal TID_k to a legal fog server. The legal fog server has no information to check the correctness of the message F_k. Thus, the legal fog server generates the message $F'_k = h(h(MK||K_{cd})||TID_k||r'_d)$ and sends these messages to the cloud server. The cloud server computes and checks whether $F'_k = F_k$. Since the attacker does not know the correct r_d, the correct message F_k cannot be generated. Thus, the attacker cannot establish a session key with the fog server, and the impersonation attack will not be achieved in the proposed scheme.

C. RESISTANCE TO MAN-IN-THE-MIDDLE ATTACK

When role A and role B want to communicate with each other, the attacker will try to intercept the transmission content of both parties, which is a man-in-the-middle attack. In our proposed scheme, the communication content of both parties is encrypted by the session key. If the attacker cannot know the session key, he/she will not be able to obtain the communication content. Therefore, the proposed scheme prevents man-in-the-middle attacks.

Scenario: The attacker tries to intercept and obtain the plain text of the communication between the edge user and the fog server, or the plain text of the communication between the edge device and the fog server.

Analysis:When the attacker tries to intercept and obtain the plain text of the communication between the edge user and the fog server, he/she will fail due to the transmitted message is encrypted by the session key SK_{ij}. The attacker cannot know the random number r_f of the legal fog server, he/she cannot calculate the correct session key $SK_{ij} = h(r'_u \oplus r'_c \oplus r_f)$ through $r'_u \oplus r'_c = Q_i \oplus h(ID_i||TID_i^{'new})$. The attacker also cannot know the random number r_u of the legal edge user, he/she cannot calculate the correct session key $SK_{ij} = h(r'_f \oplus r'_c \oplus r_u)$ through $r'_f \oplus r'_c = S_j \oplus h(ID_i||TID_j^{'new})$. In the same scenario, when the attacker tries to intercept and obtain the plain text of the communication between the edge device and the fog server, he/she will fail due to the transmitted message is encrypted by the session key SK_{ik}. The attacker cannot know the random number r_f of the legal fog server, he/she cannot calculate the correct session key $SK_{ik} = h(r'_d \oplus r'_c \oplus r_f)$ through $r'_d \oplus r'_c = Q_k \oplus h(ID_i||TID_i^{'new})$. The attacker also cannot know the random number r_d of the legal edge device, he/she cannot calculate

the correct session key $SK_{ik} = h(r'_f \oplus r'_c \oplus r_d)$ through $r'_f \oplus r'_c = S_k \oplus h(ID_i||TID_k^{new})$. Therefore, the attacker cannot achieve the purpose to obtain the plain text of the communication between the edge user and the fog server, or the plain text of the communication between the edge device and the fog server. Therefore, the proposed scheme prevents man-in-the-middle attacks.

D. RESISTANCE TO REPLAY ATTACK

When role A sends a message to role B, the attacker intercepts the message and sends the same message to role B again later. Similarly, when role B sends a message to role A, the attacker intercepts the message and sends the same message to role A again later. In our proposed method, pseudo-identity and random number will be changed in every communication round, thus resisting replay attack.

Scenario:When the fog server sends a message to the cloud server, the attacker intercepts the message and sends the same message to the cloud server again later. Similarly, when the cloud server sends a message to the fog server, the attacker intercepts the message and sends the same message to the fog server again later.

Analysis: When the fog server sends a message to the cloud server, the attacker intercepts the message and sends the same message to the cloud server again later. The message $P_j = h(h(MK||K_{cf})||TID_i||r_f)$ or $P_k = h(h(MK||K_{cf})||TID_i||r_f)$ sent by the fog server to the cloud server contains TID_i, when the same content was previously sent, the TID_i in the cloud server has been updated to $TID_i^{new} = TID_i \oplus r'_f$, the cloud server will directly discard this message, and the attacker will not be able to get any response. In the same scenario, when the cloud server sends a message to the fog server, the attacker intercepts the message and sends the same message to the fog server again later. The message $Q_j = r'_u \oplus r_c \oplus h(ID_i||TID_i^{new})$ or $Q_k = r'_d \oplus r_c \oplus h(ID_i||TID_i^{new})$ sent by the cloud server to the fog server contains TID_i^{new}, when the same content was previously sent, the TID_i in the fog server has been updated to $TID_i^{new} = TID_i \oplus r_f$, the fog server will directly discard this message, and the attacker will not be able to get any response. Therefore, the attacker cannot achieve the purpose by replay the same message from the fog server to the cloud server, or by replay the message from the cloud server and the fog server. The proposed scheme can resist replay attacks.

E. RESISTANCE TO PRIVACY EXPOSURE ATTACK

Another form of privacy attack involves attempting to obtain a person's physical location by tracing any personal device. If a terminal device continues to send the same parameters, then this device will be tracked by the attacker, causing privacy exposure. In our proposed architecture, the edge user uses a pseudonym TID_j and the edge device uses a pseudonym TID_k. The pseudonym TID_j and TID_k is changed for every communication round to avoid location tracking. Thus, location privacy is protected and avoided privacy exposure attacks.

TABLE 5. Execution time of the various cryptographic operations.

Symbol	Description	Execution time
T_{bp}	Bilinear pairing	17.4 ms
T_{fe}	Fuzzy extraction operation [8]	17.1 ms
T_{ecm}	Elliptic curve point multiplication	13.5 ms
T_h	One-way hash function	0.42 ms

F. RESISTANCE TO LOST/STOLEN SMART DEVICE ATTACK

The smart device lost/stolen is an inherent limitation of authentication protocol and we found that the best solution is to prohibit the guesstimate chance of the off-line password guessing attack. The sensitive parameters stored in edge user's smart device are $\{TID_j, C_j, D_j, h(\cdot), h(K_{cu}), n_u\}$ in our proposed scheme and we assume the attacker can extract all of them by using the power analysis attack. Therefore, knowing all the sensitive parameters, the attacker may try to derive user's identity ID_j and biometric key BIO_j in off-line manner. To derive the secret value of EU_i, which is $B_j \oplus h(MK||K_{cu}) = D_j \oplus h(ID_j||BIO_j||n_u)$, the attacker needs to know identity ID_j and biometric key BIO_j of EU_j together. However, it is computationally infeasible for attacker to derive correct $B_j \oplus h(MK||K_{cu})$ without the knowledge of ID_j and BIO_j and the proposed scheme is secure against lost/stolen smart device attacks.

G. RESISTANCE TO EDGE DEVICE PHYSICAL CAPTURE ATTACK

When physical capture attack on edge device is launched, the attacker may try to break into the system by using a compromised edge device. First of all, the attacker can extract the sensitive parameters $\{TID_k, ID_i, B_k, h(MK||K_{cd})\}$ stored in the captured edge device ED_k's memory. Since the master secret key MK of CS and the long-term secret key K_{cd} is embedded in secure one-way hash function, the attacker cannot derive the correct master secret key MK and long-term secret key K_{cd}. In addition, the session key established between ED_k, FS_i and CS is $SK_{ik} = h(r_d \oplus r_f \oplus r_c)$. Since all random numbers selected by them are distinct for all the edge devices in the system, use of random numbers make all the session key SK_{ik} are also distinct. As a result, compromise of ED_k does not lead to compromise the session keys between other non-compromised edge devices and the same fog server FS_i.

H. RESISTANCE TO KNOWN SESSION KEY ATTACK

Assume that an attacker knows the session key for a particular session. The attacker may use the old compromised session key to obtain sensitive parameters and keys for subsequent communication sessions. As we know, the session keys SK_{ij} and SK_{ik} are hash values of participants' random numbers and it is computational difficulty of one-way hash function.

TABLE 6. Performance comparisons between our proposed scheme and other related existing schemes.

	Jia et al. [19] (2019)	Wazid et al. [33] (2019)	Nikravan et al. [27] (2020)	Wu et al. [34] (2021)	Ours
P1	$2T_{bp}$+$7T_{ecm}$+$18T_h$	$1T_{fe}$+$5T_{ecm}$+$35T_h$	$4T_{bp}$+$10T_{ecm}$+$25T_h$	$3T_{bp}$+$10T_{ecm}$+$21T_h$	$22T_h$
P2	154.26 ms	99.3 ms	215.1 ms	196.02 ms	9.24 ms

P1: Computation cost during authentication and key agreement phase
P2: Rough estimation

The attacker cannot derive the new session keys from the old compromised session key without the knowledge of current random numbers. Therefore, the proposed scheme is resilient against known session key attacks.

I. PROVISION OF FORWARD AND BACKWARD SECRECY

Even if the session keys SK_{ij} and SK_{ik} between the sender and the receiver are compromised at any point by an attacker, the system still satisfies forward and backward secrecy. The attacker may use the session keys SK_{ij} and SK_{ik} for future communication or to obtain previous messages. However, in the proposed scheme, the session keys SK_{ij} and SK_{ik} are established by random numbers, and may only be used in the current round. The attacker cannot use the same session keys SK_{ij} and SK_{ik} for future communication or to obtain previous messages. Thus, the proposed scheme achieves forward and backward secrecy.

V. PERFORMANCE EVALUATION

In this section, we benchmark the performance of the proposed scheme with the related existing schemes [19], [27], [33], [34] to demonstrate that our authentication scheme for fog computing paradigm is more efficient than the compared scheme and hence can be workable for various IoT-driven applications and services. For convenience to evaluate the computation operations, we define some symbols (T_{bp}, T_{fe}, T_{ecm}, T_h) and give the execution time of these cryptographic operations in Table 5. The execution time of a bitwise XOR operation is negligible and we omit this operation for performance evaluation. From Table 6, it is clear that our proposed scheme needs less computation time during authentication and key agreement phase as compared to related existing schemes and is feasible for resource-limited devices in fog computing environments.

VI. CONCLUSION

In recent years, fog-driven IoT applications become popular among researchers due to their vital features such as heterogeneity, low latency, real time interactions, data locality, location awareness, geographical distribution and support for mobility etc. We first discussed the critical issues of anonymous authentication and secure communication in fog computing environments. We then introduced a more lightweight and secure authentication scheme for ensuring privacy preserving and key agreement in fog computing services to erase the various security pitfalls found in existing authentication schemes. The security proof and performance evaluation demonstrate that the proposed authentication scheme indeed has more security features with better performance when compared with other recent existing schemes, which is more suitable for the practical service of network system based on the fog computing environment.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable suggestions and comments.

REFERENCES

[1] M. Aazam and E.-N. Huh, "Fog computing: The cloud-IoTIoE middleware paradigm," *IEEE Potentials*, vol. 35, no. 3, pp. 40–44, May 2016.
[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
[3] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar./Apr. 2017.
[4] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.
[5] S. Banerjee, A. K. Das, S. Chattopadhyay, S. S. Jamal, J. P. C. Rodrigues, and Y. Park, "Lightweight failover authentication mechanism for IoT-based fog computing environment," *Electronics*, vol. 10, no. 12, pp. 1–25, 2021.
[6] R. Brzoza-Woch, M. Konieczny, B. Kwolek, P. Nawrocki, T. Szydło, and K. Zieliński, "Holistic approach to urgent computing for flood decision support," *Proc. Comput. Sci.*, vol. 51, pp. 2387–2396, Jan. 2015.
[7] Y. Cao, S. Chen, P. Hou, and D. Brown, "FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation," in *Proc. IEEE Int. Conf. Netw., Archit. Storage (NAS)*, Aug. 2015, pp. 2–11. [Online]. Available: https://ieeexplore.ieee.org/document/7255196
[8] S. Challa, A. K. Das, V. Odelu, N. Kumar, S. Kumari, M. K. Khan, and A. V. Vasilakos, "An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks," *Comput. Elect. Eng.*, vol. 69, pp. 534–554, Jul. 2018.
[9] 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, doi: 10.1080/17517575.2020.1712746.
[10] C.-L. Chen, Y.-Y. Deng, C.-T. Li, S. Zhu, Y.-J. Chiu, and P.-Z. Chen, "An IoT-based traceable drug anti-counterfeiting management system," *IEEE Access*, vol. 8, pp. 224532–224548, 2020.
[11] C.-L. Chen, Y.-Y. Deng, W.-J. Tsaur, C.-T. Li, C.-C. Lee, and C.-M. Wu, "A traceable online insurance claims system based on blockchain and smart contract technology," *Sustainability*, vol. 13, no. 16, p. 9386, Aug. 2021.
[12] P. Gope, "LAAP: Lightweight anonymous authentication protocol for D2D-aided fog computing paradigm," *Comput. Secur.*, vol. 86, pp. 223–237, Sep. 2019.
[13] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, and Y. Ma, "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 82–88, Jul. 2018.
[14] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.
[15] S. H. Islam and G. P. Biswas, "Design of improved password authentication and update scheme based on elliptic curve cryptography," *Math. Comput. Model.*, vol. 57, nos. 11–12, pp. 2703–2717, 2013.
[16] S. H. Islam, "A provably secure ID-based mutual authentication and key agreement scheme for mobile multi-server environment without ESL attack," *Wireless Personal Commun.*, vol. 79, no. 3, pp. 1975–1991, 2014.

[17] S. H. Islam, "Design and analysis of an improved smartcard-based remote user password authentication scheme," *Int. J. Commun. Syst.*, vol. 29, no. 11, pp. 1708–1719, Jul. 2016.
[18] S. H. Islam, P. Vijayakumar, M. Z. A. Bhuiyan, R. Amin, M. R. Varun, and B. Balusamy, "A provably secure three-factor session initiation protocol for multimedia big data communications," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3408–3418, Oct. 2018.
[19] 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, pp. 4737–4750, May 2019.
[20] D. Koo and J. Hur, "Privacy-preserving deduplication of encrypted data with dynamic ownership management in fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 739–752, Jan. 2018.
[21] C.-T. Li, T.-Y. Wu, C.-L. Chen, C.-C. Lee, and C.-M. Chen, "An efficient user authentication and user anonymity scheme with provably security for IoT-based medical care system," *Sensors*, vol. 17, no. 7, pp. 1482, 2017.
[22] C.-T. Li, D.-H. Shih, and C.-C. Wang, "Cloud-assisted mutual authentication and privacy preservation protocol for telecare medical information systems," *Comput. Methods Programs Biomed.*, vol. 157, pp. 191–203, Apr. 2018.
[23] C.-T. Li, D.-H. Shih, C.-C. Wang, C.-L. Chen, and C.-C. Lee, "A blockchain based data aggregation and group authentication scheme for electronic medical system," *IEEE Access*, vol. 8, pp. 173904–173917, 2020.
[24] Q. Lin, H. Yan, Z. Huang, W. Chen, J. Shen, and Y. Tang, "An ID-based linearly homomorphic signature scheme and its application in blockchain," *IEEE Access*, vol. 6, pp. 20632–20640, 2018.
[25] H. Ma, G. H. Tian, and L. C. Zhang, "Anti-leakage client-side deduplication with wwnership management in fog computing," *Int. J. Netw. Secur.*, vol. 22, no. 1, pp. 24–35, 2020.
[26] S. Mahmood, M. Gohar, J.-G. Choi, S.-J. Koh, H. Alquhayz, and M. Khan, "Digital certificate verification scheme for smart grid using fog computing (FONICA)," *Sustainability*, vol. 13, no. 5, p. 2549, Feb. 2021.
[27] M. Nikravan and A. Reza, "A multi-factor user authentication and key agreement protocol based on bilinear pairing for the Internet of Things," *Wireless Pers. Commun.*, vol. 111, no. 1, pp. 463–494, Mar. 2020.
[28] H. Noura, O. Salman, A. Chehab, and R. Couturier, "Preserving data security in distributed fog computing," *Ad Hoc Netw.*, vol. 94, Nov. 2019, Art. no. 101937.
[29] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, 2013.
[30] S. L. Keoh, S. S. Kumar, and H. Tschofenig, "Securing the Internet of Things: A standardization perspective," *IEEE Internet Things J.*, vol. 1, no. 3, pp. 265–275, Jun. 2014.
[31] P. Varshney and Y. Simmhan, "Demystifying fog computing: Characterizing architectures, applications and abstractions," 2017, *arXiv:1702.06331*.
[32] H. Wang, Z. Wang, and J. Domingo-Ferrer, "Anonymous and secure aggregation scheme in fog-based public cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 712–719, Jan. 2018.
[33] 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.
[34] T.-Y. Wu, T. Wang, Y.-Q. Lee, W. Zheng, S. Kumari, and S. Kumar, "Improved authenticated key agreement scheme for fog-driven IoT healthcare system," *Secur. Commun. Netw.*, vol. 2021, pp. 1–16, Jan. 2021.
[35] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for Internet of Things," *J. Netw. Comput. Appl.*, vol. 42, pp. 120–134, Jun. 2014.
[36] T. Zhu, S. Dhelim, Z. Zhou, S. Yang, and H. Ning, "An architecture for aggregating information from distributed data nodes for industrial Internet of Things," *Comput. Electr. Eng.*, vol. 58, pp. 337–349, Feb. 2017.

CHI-YAO WENG received the Ph.D. degree in computer science from the National Tsing Hua University, Hsinchu, Taiwan, in 2011. From 2011 to 2015, he was a Postdoctoral Researcher with the National Sun Yat-sen University and the National Tsing Hua University. From August 2015 to January 2019, he was an Assistant Professor, and is currently an Associate Professor with the Department of Computer Science, National Pingtung University, Pingtung, Taiwan. His research interests include information security, information hiding, image privacy, and multimedia security.

CHUN-TA LI (Member, IEEE) received the Ph.D. degree in computer science and engineering from the National Chung Hsing University, Taiwan, in 2008. He is currently a full-time Professor with the Department of Information Management, Tainan University of Technology. His research interests include information security, wireless sensor networks, mobile computing, and security protocols for the IoTs and *ad hoc* networks. He had published more than 100 international journal articles and international conference papers on the above research fields. He received the 2011 IJICIC Most Cited Paper Award from *International Journal of Innovative Computing, Information and Control*. He also served as a reviewer for many SCI-index journals.

CHIN-LING CHEN received the Ph.D. degree from the National Chung Hsing University, Taiwan, in 2005. From 1979 to 2005, he was a Senior Engineer with Chunghwa Telecom Company Ltd. He is currently a Distinguished Professor. He has published over 120 articles in SCI/SSCI international journals. His research interests include cryptography, network security, and electronic commerce. He also served as a reviewer for many SCI-index journals.

CHENG-CHI LEE (Member, IEEE) received the Ph.D. degree in computer science from the National Chung Hsing University (NCHU), Taiwan, in 2007. He is currently a Distinguished Professor with the Department of Library and Information Science, Fu Jen Catholic University. His current research interests include data security, cryptography, network security, mobile communications and computing, and wireless communications. He had published over 200 scientific articles on the above research fields in international journals and conferences. He is a member of the Chinese Cryptology and Information Security Association (CCISA), the Library Association of The Republic of China, and the ROC Phi Tau Phi Scholastic Honor Society. He is an Editorial Board Member of *Mathematics*, *Electronics*, *Future Internet*, *International Journal of Network Security*, *Journal of Computer Science*, *Cryptography*, *International Journal of Internet Technology and Secured Transactions*, *Journal of Library and Information Studies*, and *Journal of InfoLib and Archives*, and a Guest Editor of *Sensors*. He also served as a reviewer for many SCI-index journals, other journals, and other conferences.

YONG-YUAN DENG received the Ph.D. degree from the Institute of Information Management, Chaoyang University of Technology, Taichung, Taiwan, in 2016. Since 2017, he has been a Postdoctoral Researcher with the Institute of Information Engineering and Computer Science, Chaoyang University of Technology. His research interests include cryptography, sensor networks, mobile commerce, and radio frequency identification systems.