The Journal of Supercomputing (2024) 80:6855–6890
https://doi.org/10.1007/s11227-023-05712-3

TAKM-FC: Two-way Authentication with efficient Key Management in Fog Computing Environments

Naveen Chandra Gowda[1,2] · Sunilkumar S. Manvi[2] · A. Bharathi Malakreddy[3] · Rajkumar Buyya[4]

Accepted: 6 October 2023 / Published online: 30 October 2023

Abstract

A mechanism of fog computing environment is employed in order to enhance the cloud computing services toward the edge devices in a range of locations with low latency. A fog computing environment is effective when compared to cloud computing for providing communication between various edge devices such as smart devices and mobile devices used by users in the same location. Even though fog servicing extends the best services of cloud computing, it also suffers from a set of security threats like authentication, key management, data privacy and trust management. Authentication with effective key management between edge devices is the most pressing security issue in fog computing. This paper proposes an effective two-way authentication between edge devices with key management in fog computing environments (TAKM-FC). The edge nodes are the user's mobile devices and set of smart devices controlled by the fog server. To improve the proposed authentication system, we have made use of techniques like fuzzy extractor and one-way hash with cryptographic primitives. The proposed TAKM-FC scheme is validated mathematically based on the ROR model and then verified using the ProVerif tool. The TAKM-FC scheme has been evaluated using iFogSim to measure the performance parameters like throughput, end-to-end delay, packet loss, energy consumption and network usage. The overhead analysis of the proposed scheme is carried out and shows that the computation cost, communication cost and storage cost are improved by 11–21%, 8–19% and 6–13%, respectively, compared to existing schemes.

Keywords Cloud computing · Fog computing · Authentication · Key management · iFogSim

Extended author information available on the last page of the article

1 Introduction

Cloud computing offers well-organized resource allocation and access based on a request from a set of clients/users [1]. Even though cloud computing makes our lives easier and more efficient, it is plagued by several challenges that include increasing data transfers, insufficiency in data capacity, low adaptation to mobility rate, excessive latency, security and privacy. The most important of them is cloud computing security and protection, which should be approached with caution in practice [2]. Cisco designed the fog computing environment to address these challenges, and then expanded its use in 2012 [3]. Distributed edge devices are used in the fog computing environment to offer storage, communication, and computing capabilities.

IoT data processing has three layers of computing that include the edge layer, the fog layer, and the cloud layer. The edge layer includes a substantial number of IoT (Internet of Things) or other smart devices and it is the foundation of a fog computing system. The Fog layer at an intermediate level contains various fog servers with computational and storage devices, whereas the Cloud layer has various service-providing devices and cloud servers [4]. Fog computing can be used in conjunction with cloud computing, but it is not a replacement for it. Fog computing services are used in various applications such as smart cities, smart vehicles, smart grids, automotive systems, healthcare and other IoT services mechanisms [4].

Fog computing, on the other hand, is an innovative and expanded service of cloud computing services, thus it suffers from similar problems such as security concerns, hardware issues, and storage safeguards. Among such issues in the fog computing environment, the most significant one is security-related problems [5] such as authentication, privacy, integrity, trust maintenance, and other security concerns are most important to consider. To accomplish safe authentication, effective key management with low computation and transmission costs is one of the most careful security concerns. To provide security, various mechanisms have been suggested for key exchange and authentication in fog computing [6, 7]. However, in dispersed fog and cloud computing settings, these must be enhanced to combat all types of security threats.

Consider the fog server will be acting as a middle party between the edge devices such as mobile devices and a set of smart devices. The edge devices communicate with each other through the fog server using an insecure channel as it is public. A scheme must be designed so that only the authorized user must access avoiding any illegal access to the smart device. This is considered to be the authentication problem. At the same time the required keys must not be generated only based on the store credentials in order to avoid the respective attacks. This is considered to be key management issue. So it requires a secured mechanism that can secure the connections so that authorized user can only be given access to smart device. This paper proposes an effective two-way mutual and multi-level authentication between the user's mobile device and smart device with efficient key management controlled by fog server.

1.1 Motivation and problem statement

Data flow between the edge devices via fog server is insecure as the channel is public and easily accessed by malicious users in the vicinity. This raises an issue of authentication between edge devices and also with the fog server. The authentication can be considered to be a device authentication and message authentication which ensures the legitimacy of the device and the exchanged message. Device authentication is all about the validation of identity and other credentials of the device, whereas message authentication is to maintain the integrity of communication. An authentication scheme must satisfy certain requirements such as: (i) computation, communication and storage overhead must be low. (ii) the authentication and key management must be strong and scalable. (iii) there must be provision for re-authentication and revocation.

The most popular authentication schemes proposed for fog and cloud computing using symmetric and asymmetric cryptographic use lengthy certificates and tripartite algorithms for key exchange and authentication. Also, the session key generation used to be carried out by fog servers instead of edge devices which lead to heavy network overhead such as computation, communication, and storage overheads. They also face too many security attacks and are unsuitable for multi-level authentication in a decentralized computing environment. Considering all these, this work aims to propose an effective multi-level and Two-way Authentication between edge devices with Key Management in Fog Computing environments (TAKM-FC). It must ensure device authentication and optimizes communication and computation overheads and mitigates several attacks.

1.2 Contributions in the paper

The major contributions of TAKM-FC are highlighted as:

1. Proposed a fog-controlled mechanism to provide a two-way mutual, multi-level authentication between the smart device and user's mobile device through session key agreement.
2. Mathematical analysis of the TAKM-FC is discussed based on ROR model and theoretical analysis is discussed by considering a set of related attacks and compared with existing schemes.
3. The overhead analysis of the proposed scheme is analyzed and shows that the computation cost, communication cost and storage cost are improved by 11–21%, 8–19% and 6–13%, respectively, compared to existing schemes.
4. The TAKM-FC has been implemented and evaluated using iFogSim and measured throughput, end-to-end delay, packet loss, energy consumption, and network usage for different cases.

1.3 Organization of Paper

The rest of the paper is organized as follows: Sect. 2 elaborates and distinguishes the existing works and the background required for the work is given in Sect. 3. The detailed discussion of each phase in the proposed work is highlighted in Sect. 4. The security analysis including formal and informal security analysis is carried out with a comparison with existing works described in Sect. 5. The implementation of TAKM-FC in iFogSim and a discussion of the performance analysis considering computation, communication, and storage costs are given in Sect. 6. Finally, Sect. 7 provides the conclusion of the paper with future enhancements.

2 Related work

Bonomi et al. originally defined fog computing [3] in 2012, by describing the features and roles of fog nodes in computation. They have also demonstrated how fog engineering may be used in other areas, including the internet of vehicles, internet of everything, remote sensors, hospital management, smart networks, mechanical industries, and actuator systems. In [8], Stojmenovic et al. go into further detail on the motivation and benefits of fog computing as well as security concerns.

Most of the researchers have proposed various schemes to achieve authentication and key management for fog and cloud computing environments [9, 10]. Those schemes are broadly classified based on the mechanism such as (i) Cryptography (ii) Signature (iii) Verification. Cryptographic schemes can be symmetric (message authentication codes, hash function, etc.), asymmetric (public key infrastructure, elliptic curve cryptography, etc.) and other ID-based cryptographic schemes. The signature-based schemes are single-user signature and group signature-based schemes. Verification-based schemes are batch verification and cooperative message authentication-based schemes.

Several authors have proposed authentication systems with mutual authentication in the literature [11, 12]. The focus was on the user with a fog server and the user with smart device authentication, in which the users may use their mobile devices as end devices with much more limited resource usage than the fog server as a multi-factor authentication technique [13, 14]. P. Kumar et al. [15] provided a mutual authentication technique with untraceability using symmetric key-based functions. In [16], Braeken et al. described a system that uses a Trusted Third Party (TTP) to perform chaos-based operations. Later many additional mutual identity-based authentication techniques have been suggested utilizing TTP [17, 18]. But here the TTP must be remained active for complete key agreement phase in all of these schemes, but this is not possible for all applications.

Further authors Al Hamid et al. [19] came up with the idea of using TTP to drive the key management. The disadvantage of these techniques is that the fog server is in charge of creating the session key, leaving the other two entities inactive while the key is generated. As a result, these systems are extremely sensitive to key generation and management attacks. As a result, C. Ke et al. [20] and Wu. TY et al. [21] proposed a way to create the session key based on the shared

smart cards at end nodes itself rather than at the fog server. However, it may also lead to numerous ephemeral secret leaking attacks. To address this, authors C.-M. Chen et al. in [22] offered a secret key exchange, and D. Tiwari et al. in [23] expanded the scheme to include a multi-server mechanism. By taking into account the computational and storage costs, the technique suggested withstanding numerous attacks in a multi-server scenario [24]. Furthermore, because the tasks are executed on end nodes and fog nodes, which are publicly accessible and often active in the face of attacks, this is an essential assumption [25]. In terms of secure identity-based schemes, such as the one given in [26], which are intended for use in mobile computing environments where the end nodes are connected to an authentication server for getting the access rights.

Authors Wu. TY et al. in [27] says that without the use of TTP, authentication using dynamic ID exchange cannot offer total accountability. In [28], Alsahlani et al. expanded mutual authentication to include proof of a link to retain responsibility, although this needed a separate memory for security. However, M. Wazid et al. in [29] addresses the problem by just utilizing TTP for the deployment step, with the remaining stages running without it. Its main goal is to cut down on the computation costs. Yadhav et al. in [30] proposed the symmetric key authentication to provide anonymity and unlinkability for fog-powered smart devices, which was later enhanced in [31]. They claim that requiring physical credentials at IoT devices for authentication is ineffective, therefore he advocated employing the physical unclonable feature to provide mutual authentication without the need for physical credentials. M. Wazid et al. in [32] focuses on the internet of vehicle deployment and could achieve some of the known attacks. H.S. Ali et al. has combined the methods of fuzzy-verifiers and honeywords to design and proven a secured three factor authentication mechanism based on extended chaotic maps utilizing the AVISPA tool [33]. According to S. Lu et al. in [34], their protocol allows an authentication mechanisms and produces a session key. It may be upgraded to include a meter reading and a fog node. The method is secure because the learning parity with a noise problem is NP-complete, which implies it can withstand quantum computing attacks. The method is resistant to quantum assaults since it uses hash chains and authorized encryption. Furthermore, entities do not need a permanent master key and session keys are generated using a hash function. They looked at a smart emergency system in which an edge application sends out alert signals to pre-determined locations. In [36], U. Chatterjee et al. have considered an effective security solution for a smart health operations system involving a cloud using an Elliptic Curve Cryptography framework claiming that their framework preserves security and privacy features and qualities seen in other frameworks in the same context. The overview of the survey carried out on various existing works is shown in Table 1.

The major research gaps identified upon the literature of existing works as follows:

1. All the methods do not give privacy to unknowns, nor do they need a pairing procedure on the user's device. Forward privacy cannot be accomplished because the session key produced by the protocol mechanism is constant.

Table 1 Summary of related works carried out

References & year	Edge nodes	Third-Party	Key management technique	Limitations
[13] 2021	IoT-Fog-Cloud environment	Used	ECC, SHA, handshake mechanism	Complexity analysis is lacking and not compared with other handshaking mechanisms
[25] 2018	Client and server mechanism	Not used	Ephemeral-Secret cryptographic primitives	User authentication with edge device was not considered
[27] 2021	Mobile edge devices	Not used	Hash and cryptographic primitives	All possible attack threats cannot be addressed by dynamic security notion
[22] 2020	Smart devices	Used	Cryptographic primitives	The overall overhead including computation and communication costs can be decreased
[28] 2021	IoT in healthcare applications	Used	One-way cryptographic hash functions, bitwise XOR operation	Access levels with scalability of the system is not considered in the study
[29] 2019	Smart devices in edge layer	Used	Bitwise XOR operation with one-way cryptographic hash functions	The complexity of the model can be still minimized, overcoming all possible authentication attacks
[24] 2020	Multi-client multiserver architecture	Used	ECC based three-factor user authentication	Access levels with scalability of the system is not considered in the study
[21] 2019	Vehicular nodes	Not used	cryptographic primitives without bilinear pairings	Vehicle to vehicle and vehicle to fog communication was not considered
[14] 2019	Mobile edge nodes	Not used	ECC with two way authentication	The overall overhead including computation and communication costs can be decreased
[33] 2022	IoT devices	Used	Credential-based authentication	The computation and communication complexity is not analyzed, and all the security attacks cannot be addressed through dynamic security notion
[34] 2021	Smart meters in microgrid	Used	quantum computing	It was not more feasible and practical
[35] 2022	Wireless IoT devices	Used	elliptic curve cryptography	There was not much investigation on finding the level of access support related to other access structures

Table 1 (continued)

References & year	Edge nodes	Third-Party	Key management technique	Limitations
[36] 2022	Internet of things	Used	Fuzzy extractor with hash functions	The fog server must be able to control the different access control levels

2. Most of the existing schemes use the tripartite algorithm for key exchange, and they are also extremely sensitive to adversarial attacks.
3. The session key generation must be carried at the edge device level instead of depending on fog server.
4. Most of the existing mechanisms will consume high overhead due to lengthy certificates and are not suitable for decentralized computing environment.

To address these gaps, we propose a mechanism for secure authentication with a key agreement for fog-driven smart device access to users, which does not suffer from the drawbacks found in the current schemes.

3 System model and security considerations

The system model of the proposed TAKM-FC is introduced here with a related threat model. Then we provide all the possible security attacks related to authentication and key management. Finally introducing the preliminary background to the proposed scheme.

3.1 System model

Figure 1 shows the system model of the proposed TAKM-FC approach. The TAKM-FC has five entities: Trusted Third Party (TTP), User (*Uj*), Mobile Device (MD*j*), Fog Server (FS), and Smart Device (SD*i*). A TTP is a central entity that is responsible for deploying the *Uj*, FS, and SD*i*. The FS provides services as an intermediary between the *Uj* and SD*i*. SD*i* is a collection of sensors or smart nodes under a specific fog server that accumulates real-time data and stored in FS which can be accessed by a user upon request. The *Uj* is a user using the mobile device (MD*j*) that will connect to a fog server to obtain data collected by a smart device.

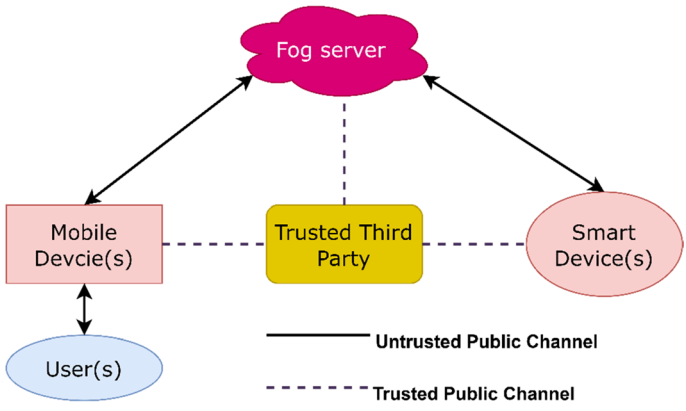

Fig. 1 System model

The case study considered here is that the smart devices (data producers) connected to the fog server will collect real-time data and send it to fog server for storage. When any user (data consumer) wishes to access the data of the smart device, the user must be granted permission to view and access the data. To achieve this requirement, we need to provide a two-way, multi-level authentication between the smart device and the user's mobile device.

3.2 Threat model

We have considered a threat model based on the Dolev-Yao [37] in the proposed work. It is elaborated as:

- The channel used between the entities is not fully encrypted so an adversary can eavesdrop and fetch the sensitive data.
- The user messages are sent through public channels where an adversary can perform unauthorized access, update and delete the data.
- An adversary can easily access and use the data from mobile device when it is lost by the owner or stolen by an opponent.
- An adversary can compromise the edge devices and fetch the session keys.
- An adversary can compromise the fog server and fetch the client information and shared keys.

3.3 Security requirements

Because of the security prerequisites in a fog computing environment, some of the attacks need to be forestalled in the authentication protocol are elaborated as:

- *Forward/Backward secrecy* After the session gets over, no further messages/instructions will be considered from the respective user. Meantime, when a new user joins the existing communication group, the new user must not be open to previously communicated messages/instructions.
- *Replay attack* An unauthorized third party holds the record of blocked data in the computational system, wherein an attacker tries to mislead the other authorized user.
- *Impersonation attack* A successful impersonation attack in which an attacker assumes the legal identity of any of the authorized parties in the computing system environment.
- *Offline Password guessing attack* An unauthorized adversary can make use of previous messages transferred and the credentials stored in the mobile device (lost/stolen) and guess the unique password.
- *Ephemeral secret leakage attack* The secret session key of edge devices can be retrieved by the intruder when it is not generated based on both long-term (credentials) and short-term (random numbers).

- *Man-in-the-middle attack:* An attacker acts as a middle man, it receive a messages transmitted from an authorized sender, modifies and forwards them to other authorized recipients. The adversary may also delete the messages.
- *Insider attack:* An authorized user of a computing environment or system, tries to misuse and mislead access to the authorized system environment.
- *Mobile stole/lost attack* When a mobile device is lost or stolen by an attacker, he can access all the credentials and other data in the local storage of the mobile device.
- *Un-traceability* An adversary must not be able to track the different activities carried out by any devices in a computing environment during communications.
- *User anonymity* An attacker must be unaware of the real and unique identity of any communicating device in the environment while communicating.

3.4 Preliminaries

This section discusses the basic preliminary mechanisms used in the proposed *TAKM-FC* scheme.

3.4.1 Cryptographic primitives

Consider G as a multiplicative subgroup of Zp^*, p being the large prime value, the identity element could be, $f=1$ *and* g, g being the generator value of G. The assumption is made as discrete logarithms in $g \in G$ are computationally infeasible. The G is the base point among a set of points on the elliptic curve. Consider an elliptic curve ECC$p(x, y)$ over a prime Galois field GF(p) with (x,y) are being constants. The bivariate polynomial function of degree t in a finite Galois field (GF(p)$=Zp^*$) is calculated using Eq. 1.

$$F(x, y) = \sum_{m,n=0}^{t} \left(a_{m,n}\, x^m\right) y^n \tag{1}$$

The coefficients for the function are selected from the set GF(p) and are symmetric, so $F(x,y)=F(y,x)$ [38]. We have made use of the symmetric polynomial function at smart device and fog server for generating the storing the mutual symmetry.

3.4.2 Fuzzy extractor function

The fuzzy extractor function is the most used method for handling the user biometric effectively [39]. The fuzzy extractor functions include two operations as Generation and Reproduction operations.

Generation (Gen) It is used to generate the bio-key(α) for a specific user biometric with the help of public reproduction parameter (β). The Gen(.) reads the biometric (BIO) as input, process and produces the bio-key using Eq. 2.

$$(\alpha, \beta) = \text{Gen(BIO)} \tag{2}$$

Reproduction (Rep) It is the operation used for the verification process by extracting or recovering the bio-key based on current user biometrics and the same public reproduction parameter. The Rep(.) reads the user biometric (BIO′) and reproduction parameter *(β),* it recovers the bio-key using Eq. 3.

$$\alpha' = \text{Rep}(\text{BIO}', \beta) \tag{3}$$

We have used the fuzzy extractor function for user registration and login process in the proposed work.

4 Proposed TAKM-FC scheme

The proposed TAKM-FC scheme uses a fuzzy extractor function, bitwise XOR operations, a one-way hash function, and other cryptographic primitives. The fuzzy extractor function has been used for user login and authentication to the mobile device with cryptographic primitives. Later the bitwise XOR operations, a one-way hash function, and other cryptographic primitives are used for communication between the mobile device and smart device through the fog server.

The TAKM-FC scheme has three phases: Deployment and registration, Secret key management, and Authentication with session key generation. Figure 2 shows the overview of the TAKM-FC.

Phase 1 The fog server, smart device and user's mobile device must be registered with a trusted third party before they start their transactions. Then they can identify each other without the intervention of a third party.

Phase 2 The smart device and fog server mutually accept the channel by generating the common secret key at both ends.

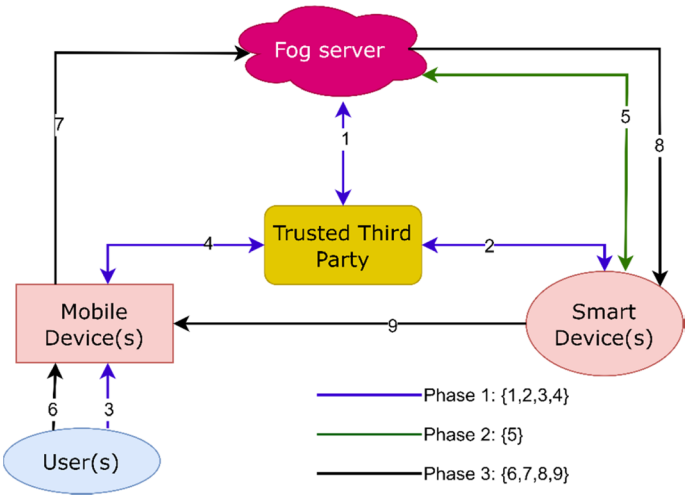

Fig. 2 Overview of the proposed scheme

Phase 3 The process of authentication will begin with a user login to the registered mobile device. The request will be sent to the fog server, upon successful authentication the request is forwarded to a smart device. The smart device will authenticate and generate the session key, then send it to a respective mobile device for successful authentication.

We have also considered 2 optional phases in the scheme.(1) Password Update based on new biometric: When a registered user wishes to change the password, it can be done without interacting with TTP based on his biometric and current password. (2) New Mobile Device Updating Phase: When a registered user wishes to change his/her mobile device or when it is lost, the new mobile device needs to be updated with TTP and then with FS. Each phase is discussed in detail in the following sections. Table 2 lists the notations used in the description of the proposed TAKM-FC scheme.

4.1 Phase 1: deployment and user registration

The fog server, smart devices, and mobile devices must be deployed by a trusted third party. The user must also register using mobile device based on his or her credentials. The deployment process begins with the assignment of the appropriate ID and temporary certificates by TTP.

4.1.1 Deployment of Fog server and smart devices

The deployment of fog server and smart device is given in Fig. 3. The TTP chooses a unique ID and generates a temporary certificate for *FS* and generate a polynomial bivariate for FS as: $F(\text{TFID}, y) = \sum_{m,n=0}^{t} \left(a_{m,n}(\text{TFID})^m\right) y^n$ using Eq. 1 as discussed in Sect. 3.4.1. Finally the *FS* is deployed by storing $< \text{TFID}, \text{TC}, F(\text{TFID}, y) >$ in its storage.

The *TTP* chooses a unique ID and generates a temporary certificate for all SDi, $i = 1,2,3,\ldots.n$ based on the current requesting timestamp. The *TTP* generate a polynomial bivariate for SDi as: $F(\text{TSID}i, y) = \sum_{m,n=0}^{t} \left(a_{m,n}(\text{TSID}i)^m\right) y^n$ using Eq. 1 as discussed in Sect. 3.4.1. Finally the SDi is deployed by storing $< \text{TSID}i, \text{TC}i, F(\text{TSID}i, y) >$ in its storage.

4.1.2 User registration with mobile device and deployment of mobile device

A user at the edge layer must register with a mobile device and in turn, the mobile device must be deployed by TTP. The user registration with a mobile device is using a set of user credentials such as user id with password and user biometric. The biometric is processed using a fuzzy extractor function using Eqs. 2 and 3 as discussed in Sect. 3.4. Later the mobile device will be deployed by TTP based on the user credentials. Any number of users *Uj* can be registered with multiple mobile devices MDj, where $j = 1,2,3\ldots.,x$ which can be deployed by the TTP.

Figure 4 illustrates the user registration and mobile device deployment process. The MDj reads the user credentials as input such as user-id, password, biometric

Table 2 Notations and descriptions

Notation	Description
TTP, K	Trusted third party, stored key at TTP
FS, SD_i, U_j, MD_j	Fog Server, i^{th} Smart Device, j^{th} user, his/her mobile device
FID, TFID, HFID, TC	Identity of FS, temporary id, hashed id, temporary certificate of FS
rts1,rts2,rts3	Timestamp at request for deployment
SID_i, $TSID_i$, TC_i	Identity of SD_i, temporary identity, temporary certificate of SD_i
$HSID_i$, $HSID_i$*	Hashed id of SD_i stored, hashed id generated at authentication
UID_j, PWD_j, $nPWD_j$	Identity of U_j, password of U_j, new password of U_j
$PPWD_j$	Protected password of U_j stored at MD_j
PPWDj'	New protected password when updated by user
PPWDj*	Protected password generated at login time
$HUID_j$, HUIDj*	Hashed id of U_j stored, hashed id generated at authentication process
BIO_j, BIOj', nBIO$_j$	Biometric of U_j at registration, biometric of U_j at login, New Biometric when updated
TC_j	Temporary certificate of U_j
r, R	Private values of U_j
r', R'	Private values of U_j when update with new mobile device
α_j	Biometric secret key of U_j for BIO_j at registration
$\alpha j'$	Biometric secret key of U_j for BIO_j at login
$n\alpha_j$	New Biometric secret key when updated
β_j	Public reproduction parameter of U_j for BIO_j
tt	Threshold time
n1, n2, n3, n4, n5	Assumed random numbers
n1, n2*, n3*, n4*, n5**	Generated random numbers
N1, N2	Private random number
SR	Secret random number
ts1, ts2, ts3, ts4, ts5	Timestamps of requests sent
ts1, ts2*, ts3*, ts4*, ts5**	Timestamps of requests received
$SecK_i$	Secret key generated at FS and SD for *ith* Smart Device
SK_{ij}, SK_{ji}	Session key generated at SD_i and MD_j for U_j
T1,T2,T3,T4,T5,T6,T7,T8	Temporary variables generated at sender
T2,T6*,T8**	Temporary variables generated at receiver
Gen()	Generation operation in Fuzzy extractor
Req()	Reproduction operation in Fuzzy extractor
h(.)	Cryptographic one-way hash function
$\|\|$	Concatenation
$\oplus$	Bitwise XOR operation
F(x,y)	Polynomial bivariate function

and private number *(*UID*j,* PWD*j,* BIO*j, r).* MD*j* generate the bio key and hashed ID based on private number then sends a registration request to TTP. The TTP in turn generates the temporary certificate and shares with MD*j*, also updates the

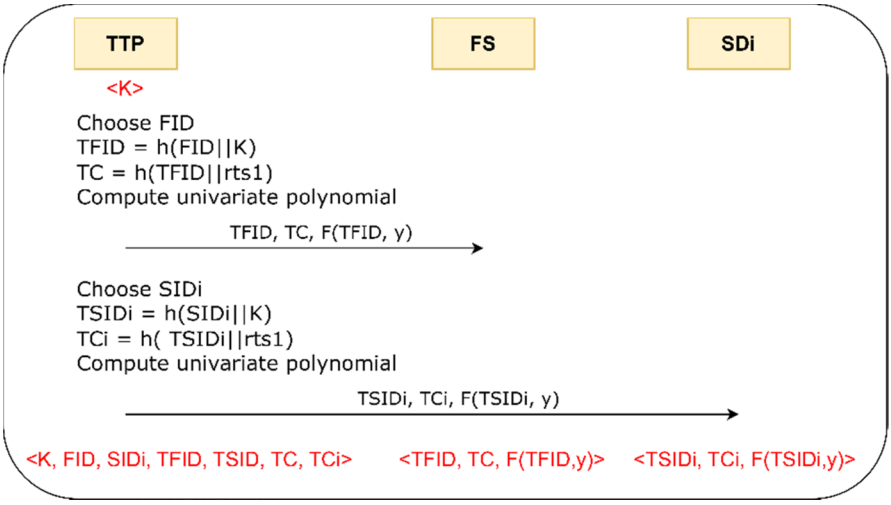

Fig. 3 Deployment of fog server and smart device

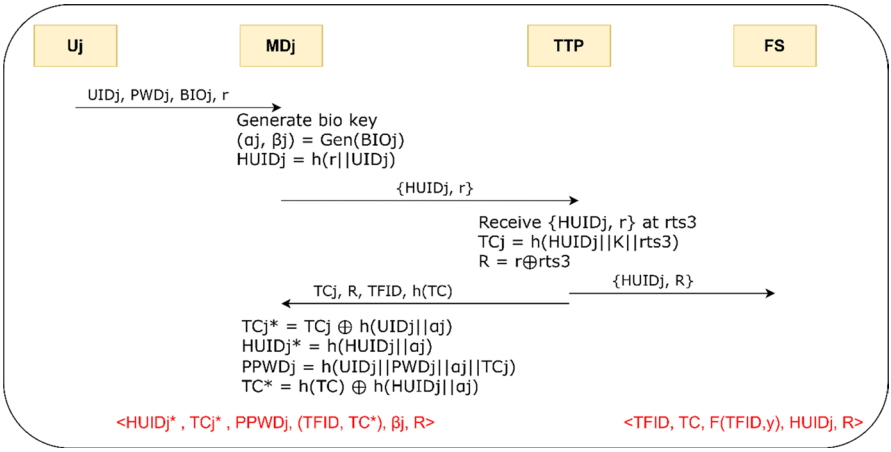

Fig. 4 User registration and deployment of mobile device

user information with FS. The MDj will update the related credentials and store in its local storage as $< \text{HUID}j *, \text{TC}j *, \text{PPWD}j, (\text{TFID}, \text{TC} *), \beta j, R >$ and deletes the other original parameters $< r,\ \text{HUID}j,\ \text{TC}j,\ h(\text{TC}) >$. The FS updates its memory as $< \text{TFID}, \text{TC}, F(\text{TFID}, y), \text{HUID}j, R >$ having the registered user data.

4.2 Phase 2: secret key management

Once the SDi and FS are deployed separately, then they must also accept each other for further communications. Here, we discuss the mutual acceptance and a secret key (SecK) is generated at both ends (SDi and FS) upon acceptance. As shown in

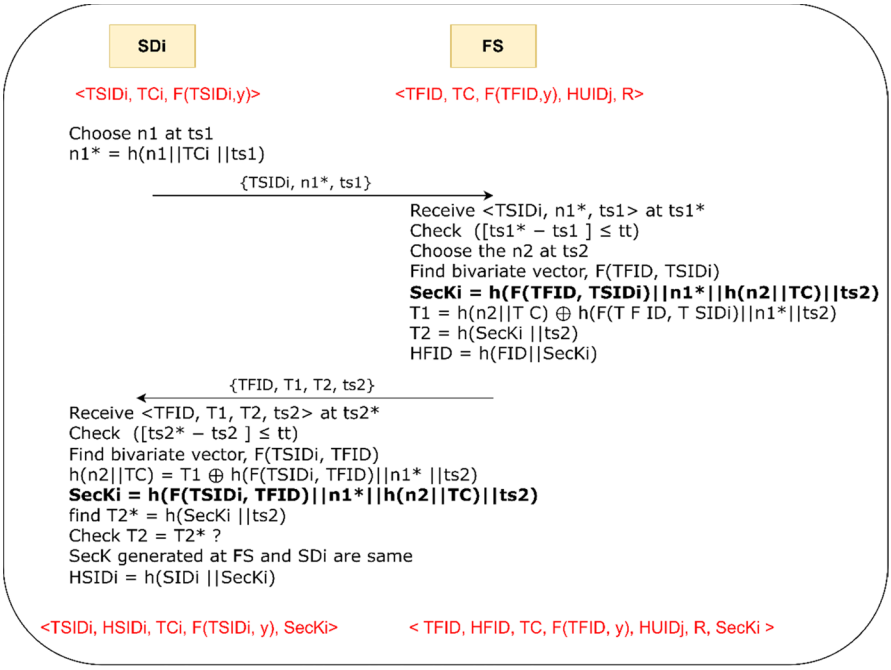

Fig. 5 Secret key management at fog server and smart device

Fig. 5 the SD*i* sends a request having its TSID and a current random number to FS. The FS will extract the received vector and generate the secret key based on received and stored credentials. FS will also produce the temporary variables based on the secret key to be shared with SD*i* for verification. Once the SD*i* receives the credentials from FS, it will generate the secret key by itself. The SD*i* will also generate the temporary variable using its secret key. The agreement between SD*i* and FS is considered to be successful if the generated and received temporary values ($T2 = T2*$) are same. Then the SecK is stored at both ends (SD*i* and FS) for further communications.

4.3 Phase 3: authentication with session key generation

The process of authentication begins only after the entities (SD*i*, FS, MD*j*) are deployed successfully. A registered user may request for accessing the smart device which requires the mutual authentication. A mutual authentication between the MD*j* and SD*i* will happen through the *FS*. Upon the successful two-way mutual authentication, a session key will be generated at MD*j* and SD*i*. The process of authentication with session key generation is given in Fig. 6. A registered user *Uj* will provide the required credentials and login to the MD*j* to reach the fog server. The mobile device validates the *Uj* based on current user credentials like user id, password and bio-metric before forwarding the request to *FS*. Upon successful login, the MD*j* will

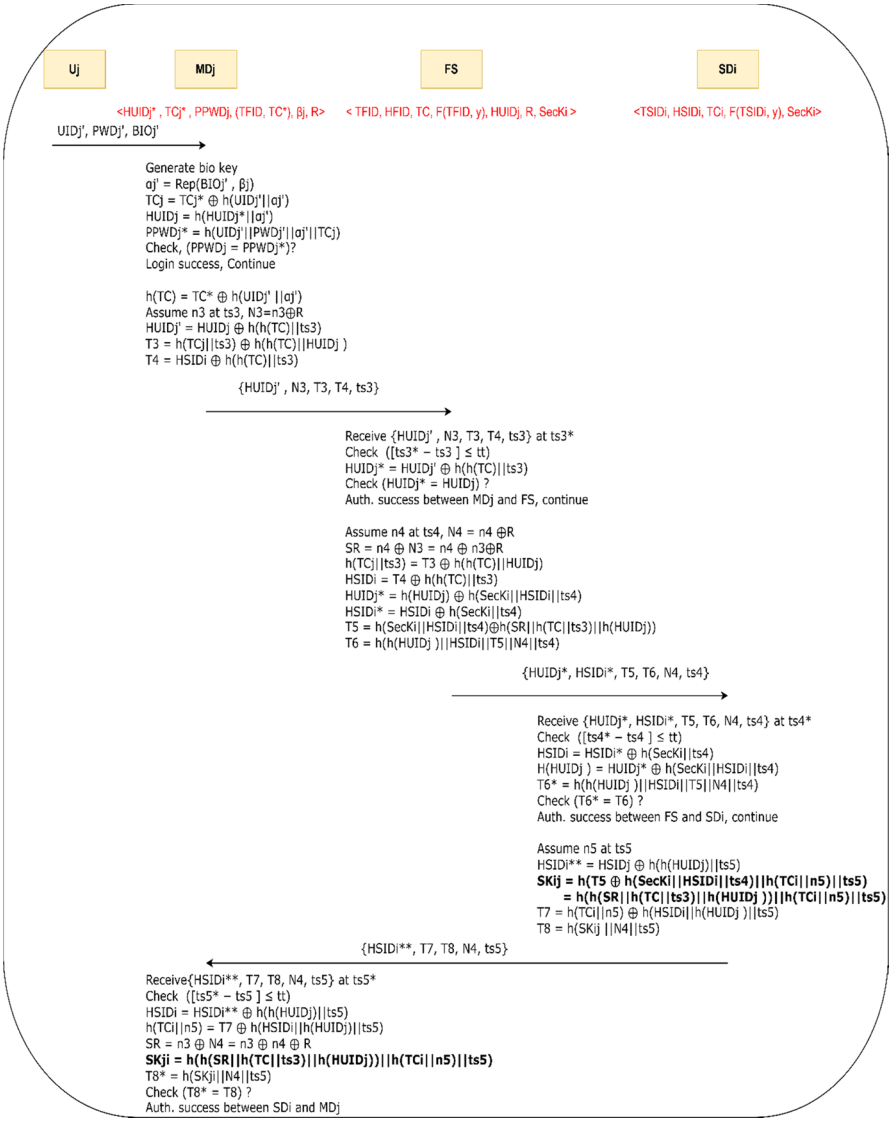

Fig. 6 Authentication with session key management

prepare the message having the hashed-id of user, temporary credentials and send the requesting message to *FS*.

The *FS* will verify the received request based on arrival time and validate the *Uj* based on the generated and stored credentials. Upon successful authentication of *Uj*, the FS will forward the request to SD*i* having the hashed-id of *Uj*, temporary credentials. The SD*i* will verify the received request based on arrival time and validate the user based on the generated and received credentials. The smart device will

prepare the message having the hashed-id of SD*i*, temporary credentials and send the message to MD*j*. The MD*j* will verify the received request based on arrival time and validate the smart device based on the generated and received credentials. The mutual authentication is successful when the session keys (SK*ij* and SK*ji*) generated at both ends (SD*i* and MD*j*) are to be same.

4.4 Password update based on new biometric phase

The updating of user password is carried out for two reasons when the user may not be willing to use the same password and bio-key for a long time or based on security concerns. The registered user (*Uj*) will be updating them in a registered mobile device (MD*j*) without the intervention of TTP or FS. As illustrated in Fig. 7, an *Uj* has to provide his/her original credentials. The MD*j* will verify the user access by generating the protected password and comparing it with the stored one. If the request is from authorized *Uj*, then MD*j* reads the new credentials from the user and generates the protected password based on the current new credentials. Finally, *MDj* will update its database by deleting the original data.

4.5 New mobile device updating phase

When the mobile device is stolen or lost, then there is a necessity for the User (*Uj*) to register to Fog Server (FS) with a New Mobile Device (nMD*j*). The new mobile device can be updated only by the legitimate user. As shown in Fig. 8 the legitimate user will have to provide the credentials, which need to be verified at MD*j* then the request will be forwarded to TTP. The TTP will generate the new certificate

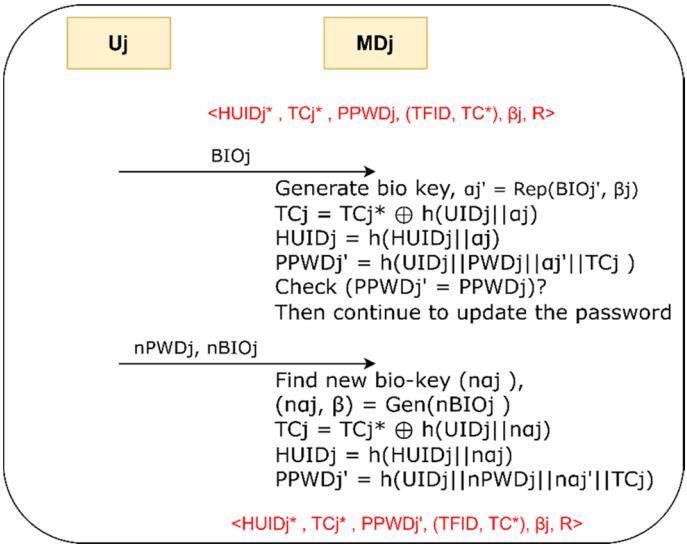

Fig. 7 User password update in mobile device

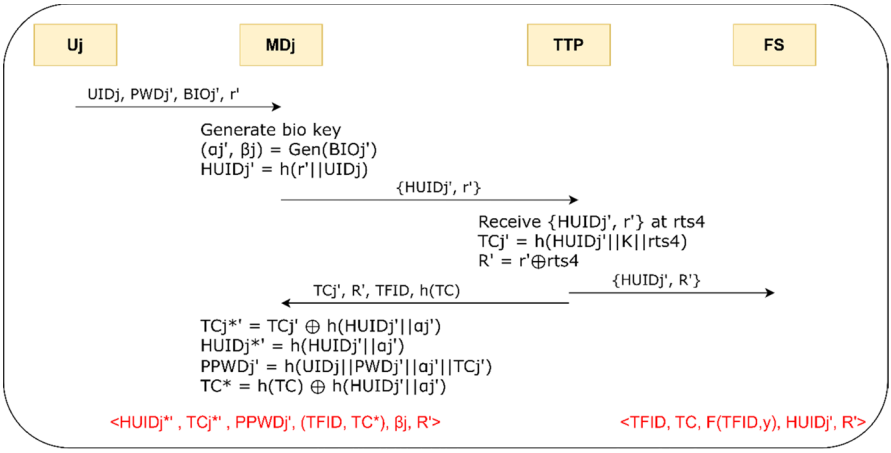

Fig. 8 New mobile device update with *TTP*

for the new MDj and deploy the new MDj. The information about the new MDj is also shared with FS for further communication. The MDj will update its storage as $< \text{HUID}j*\prime, \text{TC}j*\prime, \text{PPWD}j\prime, (\text{TFID}, \text{TC}*), \beta j, R\prime >$ and deletes the other original parameters. The FS updates its memory as $< \text{TFID}, \text{TC}, F(\text{TFID}, y), \text{HUID}j\prime, R\prime >$.

5 Security analysis

Here we discuss about the security analysis of TAKM-FC including the formal and informal security analysis. The formal analysis and verification are based on the mathematical model suitable for authentication schemes. D. Wang et al. in [40] have used the RO (Random Oracle) model and BAN (Burrows–Abadi–Needham) [41] for verification of authentication schemes and they have proved as they could not be used effectively as they will not consider the syntactical issues in the scheme. M. Wazid et al. [42]. and L. Wu et al. [43] have proved that the effective verification of an authentication scheme can be achieved based on ROR (Real-Or-Random) model. So the robustness of the TAKM-FC is analyzed using ROR model.

5.1 ROR model-based formal analysis

The security analysis of proposed work is carried out based on ROR model. As per the ROR model, the analysis is a game between adversary A and challenger C. The elements used in the analysis are as follows.

Participants Fog Server (FS), Smart Device (SDi), Mobile Device (MDj) are the participants in the proposed scheme. I^{t1}_{FS}, $I^{t2}_{\text{SD}i}$ and $I^{t3}_{\text{MD}j}$ are the instances of FS, SDi and MDj at *t1, t2,* and *t3* which are considered to be as oracles for the analysis.

Accepted state Messages being exchanged between the participants must be in order and delivered at the instance I^t for current session, then it can be considered as an acceptable state. The session identification number (sid) will be assigned to such an accepted state.

Partnering Two instances (I^{t1}, I^{t2}) are said to be partners only when they both are in an acceptable state and also share the common sid upon mutual acceptance.

Freshness The instances $I^{t2}_{\mathrm{SD}i}$ and $I^{t3}_{\mathrm{MD}j}$ are considered as fresh when the session key (SK) generated at SDi and MDj is kept private.

Adversary The ROR for the TAKM-FC is analyzed based on DY model [37]. An adversary is one who can have control on the interactions between the end participants, an adversary can also update or delete the content in the message. The adversary can perform the following operations:

- Execute *(I^{t1}, I^{t2}):* An adversary executes this to read the exchanged messages between the end participants FS, SDi and MDj. This is to apply the eavesdropping attack in analysis.
- Send (I^t, M): An adversary can send a message M to the end participant at active instance *(I^t)* and expect a reply from it. This is to apply the reply attack in analysis.
- Reveal *(I^t):* An adversary tries to capture the session key (SK) generated at SDi and MDj at instance I^t.
- Test(I^t): An adversary runs the test queries for multiple times to verify whether the session key shared is real or random.
- Random Oracle: The analysis considers a one-way hash function *h(.)* as a random oracle. The end participants can and also an adversary can access the *h(.)*. So an adversary can also create the hash like genuine participants and use it.

Theorem At a polynomial time Pt, adversary A attempts to steal the secret session key (SK) established between SDi and MDj. The advantage of adversary A in compromise and breaking the security of TAKM-FC by fetching the session key (SK) can be estimated in Eq. 4.

$$\mathrm{Adv}_A^{\mathrm{TAKM-FC}}(Pt) \leq \left(\frac{qh^2}{|\mathrm{Hash}|}\right) + (2.\mathrm{Adv}_A^{\mathrm{ECDDHP}}(Pt)) \tag{4}$$

where qh is number of hash queries, |Hash| is the range space of *(.)* and $\mathrm{Adv}_A^{\mathrm{TAKM-FC}}$ is the advantage of A in breaking the ECDDHP.

Proof Series of games are used between challenger C and adversary A to prove the theorem [42, 43]. We have considered three games, Game$i$ *($i=0,1,2$),* the advantage of A winning the game is given in Eq. 5.

$$\mathrm{Adv}^{\mathrm{TAKM-FC}}_{A,\mathrm{Game}_i} = \Pr(\mathrm{Succ}^A_{\mathrm{Game}_i}) \tag{5}$$

Here Succ is the event when A tries to guess the bit c in the Game$i$. Each game is discussed in detail as follows:

$Game_0^A$ Adversary A launches an attack on the TAKM-FC by selecting a random bit b before the beginning of $Game_0{}^A$ based on which the security definition is given in Eq. 6.

$$\mathrm{Adv}_A^{\mathrm{TAKM-FC}}(Pt) \leq |2\mathrm{Adv}_{A,\mathrm{Game}_0}^{\mathrm{TAKM-FC}} - 1| \tag{6}$$

$Game_1^A$ Adversary performs the eavesdropping attack by using execute and test queries. An adversary executes the queries to retrieve the session key and verify whether it is real or random. The session key being generated as $\mathrm{SK}ji = h(h(\mathrm{SR}||h(\mathrm{TC}||ts3)||h(\mathrm{HUID}j))||h(\mathrm{TC}i||n5)||ts5)$ is based on the credentials exchanged between the participants during the authentication phase. The messages exchanged in authentication phase of TAKM-FC are $<\mathrm{HUID}j'$, *N3, T3, T4, ts3*$>$, $<\mathrm{HUID}j*$, $\mathrm{HSID}i*$, *T5, T6, N4, ts4*$>$, and $<\mathrm{HSID}i**$, *T7, T8, N4, ts5*$>$ which has short term credentials *(*SR, *ts3, n5, ts5)* and long-term credentials *(*TC, HUID*j,* TC*i)* are protected using a collision resistant one-way hash function. So even after the adversary fetches the messages being exchanged, it is difficult to find the session key and probability of winning the *Game1* is not increased. So *Game1* is not different than *Game0* as shown in Eq. 7.

$$\mathrm{Adv}_{A,\mathrm{Game}_1}^{\mathrm{TAKM-FC}} = \mathrm{Adv}_{A,\mathrm{Game}_0}^{\mathrm{TAKM-FC}} \tag{7}$$

$Game_2^A$ Adversary uses send and random oracle queries and try to change the messages being exchanged during login and authentication phase. The messages Msg1, Msg2 and Msg3 are secured through one-way hash function and they are employed using random number at current timestamps. So it is difficult for adversary to steal and change the content of messages using send and random oracles as shown in Eq. 8.

$$|\mathrm{Adv}_{A,\mathrm{Game}_1}^{\mathrm{TAKM-FC}} - \mathrm{Adv}_{A,\mathrm{Game}_2}^{\mathrm{TAKM-FC}}| \leq \left(\frac{qh^2}{2.|\mathrm{Hash}|}\right) \tag{8}$$

An adversary try to break the security of *TAKM-FC* by simulating the hash queries and solve *ECDDHP*. Adversary A must be aware about all the credentials required to generate the session key, but those are highly secured using one-way hash. So adversary must solve *ECDDHP* to resolve the secured session key that is impossible. Generation of hash collision and solving the *ECDDHP* to break the security of *TAKM-FC* is defined in Eq. 9:

$$|\mathrm{Adv}_{A,\mathrm{Game}_1}^{\mathrm{TAKM-FC}} - \mathrm{Adv}_{A,\mathrm{Game}_2}^{\mathrm{TAKM-FC}}| \leq \left(\frac{qh^2}{2.|\mathrm{Hash}|}\right) + \mathrm{Adv}_A^{\mathrm{ECDDHP}}(Pt) \tag{9}$$

Adversary executes the game for guessing attack as in Eq. 10.

$$\mathrm{Adv}_{A,\mathrm{Game}_2}^{\mathrm{TAKM-FC}} = \left(\frac{1}{2}\right) \tag{10}$$

e application of triangular inequality based on previous equations lead to the following derivation:

$$\left(\frac{1}{2}\right).\mathrm{Adv}_{A}^{\mathrm{TAKM-FC}}(Pt) = \left|\mathrm{Adv}_{A,\mathrm{Game}_0}^{\mathrm{TAKM-FC}} - \left(\frac{1}{2}\right)\right| = \left|\mathrm{Adv}_{A,\mathrm{Game}_0}^{\mathrm{TAKM-FC}} - \mathrm{Adv}_{A,\mathrm{Game}_2}^{\mathrm{TAKM-FC}}\right|$$
$$= \left|\mathrm{Adv}_{A,\mathrm{Game}_1}^{\mathrm{TAKM-FC}} - \mathrm{Adv}_{A,\mathrm{Game}_2}^{\mathrm{TAKM-FC}}\right| = \left(\frac{qh^2}{2.|\mathrm{Hash}|}\right) + \mathrm{Adv}_{A}^{\mathrm{ECDDHP}}(Pt)$$

Finally, multiply by factor of 2 on both sides fetching Eq. 11:

$$\mathrm{Adv}_{A}^{\mathrm{TAKM-FC}}(Pt) \leq \left(\frac{qh^2}{|\mathrm{Hash}|}\right) + (2.\mathrm{Adv}_{A}^{\mathrm{ECDDHP}}(Pt)) \tag{11}$$

5.2 Informal security analysis

The informal security analysis is made by deliberating some known attacks which need to be resisted. The different security features considered are C1: Replay attack, C2: Offline Password guessing attack, C3: Impersonation attack, C4: Ephemeral secret leakage attack, C5: Man-in-the-middle attack, C6: Mobile stolen/lost attack, C7: User anonymity, C8: un-traceability attack, C9: Forward Secrecy attack, C10: Insider attack. Table 3 presents a comparison of all considered security features in the proposed scheme with other related schemes [33:CredAuth][34:QuantAuth] [35:LAMAS][36:ECCAuth] and it demonstrates that the proposed mechanism satisfies all the said security concerns.

Replay attack We have used two key points to resist replay attacks. They are random numbers with the current timestamp in each sent data segment. As a result, obtaining the random numbers *n1* to *n5* and *ts1* to *ts5* from the public channel is extremely difficult for the opponent. As a result, the adversary will be unable to repeat the messages without knowing the random number; hence, the suggested technique will be able to counteract the replay attack.

User Impersonation attack The adversary may try to log in and send the message to *FS* acting on behalf of authorized user *Uj*. An adversary may try to create the message < HUID*j', N3, T3, T4, ts3* > to send to *FS*. But to create this message an adversary must have the original details as user biometric based which itself you generate all other credentials. Also, an adversary must know the private value, r to generate R based on a random number. Without all these details an adversary cannot create the expected message, so the proposed scheme eliminates the user impersonation attack.

Fog Server Impersonation attack An adversary may try to send a message < HUID*j*,* HSID*i*, T5, T6, N4, ts4* > to a smart device on behalf of the fog

Table 3 Comparison of security features

Scheme	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
[33:CredAuth]	Y	N	Y	N	Y	Y	Y	Y	Y	Y
[34:QuantAuth	Y	Y	N	Y	Y	Y	Y	Y	Y	N
[35:LAMAS]	Y	Y	Y	Y	N	Y	N	Y	Y	Y
[36:ECCAuth]	N	Y	Y	Y	N	Y	Y	Y	Y	Y
TAKM-FC	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

server. The adversary may also assume some random numbers and current timestamps, which are used to generate the temporary variables random secret number of the message. But it also needs the hashed identifier of the user and smart device, and importantly the secret key agreed by both FS and SD*i*, which is hard to find or assume by an adversary. So without these, an adversary cannot create a valid message to send to SD*i*. So the proposed scheme eliminates the fog server impersonation attack.

Smart Device Impersonation attack An adversary may try to a message < HSID*i***, *T7, T8, N4, ts5* > to the user's mobile device on behalf of the smart device. An adversary may also assume some random numbers and current timestamps, which are used to generate the temporary variables random secret number of the message. But it also needs the hashed identifier of the user and smart device, and importantly the secret key agreed by both FS and SD*i* which is hard to find or assume by an adversary. So without these, an adversary cannot create a valid message to send to MD*j*. So the proposed scheme eliminates the smart device impersonation attack.

Offline Password Guessing attack Once an adversary has stolen the mobile device MD*j* of registered authorized user *Uj*, he can retrieve all the credentials stored in the device using a power analysis attack. That is he can access < HUID*j**, *T Cj**, *P* PW *Dj,(*T F ID*, T C*), βj, R* > but he cannot know or try to guess the deleted parameters. Without *r*, there is no use of *R* as *R* is generated by TTP using *r*. For an adversary it is hard to guess the bio-key used further, he may also fetch the bio-key but it is hard to guess the original user id, UID and PWD, which are not stored. So in the proposed scheme, it is difficult for adversaries to guess the password.

Ephemeral secret leakage attack If an adversary thinks of obtaining the session key (SK), first he needs to obtain the ephemeral secret key (SK*ij* or SK*ji*) shared between FS and SD*i*. For finding the secret key he must know the random numbers (*n1–n5*) used. If we assume the adversary can obtain these random numbers by chance, even then the adversary must find the hashed IDs used. But to find the hashed id's, he must get the original id of *Uj*, SD*i*, and FS, which is impossible to get from the public channel. This indicates that the suggested method may withstand an ephemeral secret leakage attack.

Man-in-the-middle attack Suppose an adversary wants to act in between the *Uj* and FS. An adversary may receive the message from a user as < HUID*j'*, *N3, T3, T4, ts3* >, but he cannot modify his identity as if the user (HUID*j*) because to do so an adversary must have the private value of the user *r.* Also, the HUID*j*, in turn, will be updated as HUID*j'* by using the bio-key *αj*. So the value r and αj are not traceable and modifiable by any middle person. So that any message sent to a middle person other than the actual one, the FS can easily detect and terminate the connection. So the proposed scheme overcomes the man-in-the-middle attack.

User anonymity and un-traceability attack: In the login and authentication phases, we utilized random numbers (*n1–n5*) and timestamps (*ts1* to *ts5*), both of which change from session to session. As a result, the adversary is unable to deduce the true identities of *Uj,* SD*i*, and FS. Because of this, the suggested system may be adaptable to protect the user's privacy. Furthermore, the identity of fog servers and

users is obscured by the random number, which is also unique between sessions, making this method safe against untraceability attacks.

Mobile device stolen/lost attack As explained in the password guessing attack, when the MD*j* is stolen or lost, an adversary can fetch all the credentials stored in it. But it is hard to guess the actual PWD*j*, UID*j*, and BIO*j*. Unless an adversary knows these parameters, he cannot initiate any communication with FS or SD*i*. So no use even an adversary collects the MD*j* or no worries for a registered user for further communications as the registered user can re-initiate using a new mobile device as discussed in 4.6.

5.3 Formal security verification based on ProVerif

The ProVerif [44] is used to perform the security analysis of cryptographic protocols. We used this tool to assess the security characteristics of authentication mechanism in a formal way. ProVerif is an automated cryptographic protocol validator that employs Horn clauses to express the protocol. It is commonly used for verification of the exchange of security keys and achieve mutual authentication, which includes both asymmetric and symmetric encryption methods, also hash functions, digital signature algorithms and other methods. The security analysis is made based on the DY model [37] as enumerated in the Sect. 3.2. To analyze the security of the proposed scheme, we have created two channels (private and public). A private channel is used at the time of the registration phase between the entities and TTP. A public channel denotes interface media between the entities at the authentication and key generation phase. Multiple events have been considered at authenticating entities, to generate the secret key and session key. The overall results are shown in Fig. 9. As a consequence of the aforementioned verification result, we may infer that the proposed mechanism accomplishes an effective mutual authentication and also maintain the secrecy of the session key, so that an adversary A is unable to intrude and break the process or get the session key.

6 Performance evaluation

Here, we present the overall performance analysis of the TAKM-FC scheme. First, calculate the overhead metrics of TAKM-FC and then analyze by comparing the results of TAKM-FC with existing schemes [33:CredAuth][34:QuantAuth] [35:LAMAS][36:ECCAuth]. These papers have been considered for comparison as they have proposed authentication using different cryptographic primitives such as credential-based, ECC-based and quantum-based mechanisms. Later we present the implementation and result analysis using iFogSim. The experimentation of the *TAKM-FC* is carried out in the system having a configuration as 2.4GHz CPU, Intel i7, 16 GB RAM and 1 TB hard disk.

ProVerif text output:

```
-- Query not attacker(UID[])
Completing...
Starting query not attacker(UID[])
RESULT not attacker (UID []) is true.
--------------------------------------------------------------------------------
-- Query not attacker(PPWD[])
Completing...
Starting query not attacker(PPWD[])
RESULT not attacker (PPWD []) is true.
--------------------------------------------------------------------------------
-- Query not attacker(K[])
Completing...
Starting query not attacker(K[])
RESULT not attacker (K []) is true.
--------------------------------------------------------------------------------
-- Query not attacker(SecK[])
Completing...
Starting query not attacker(SecK[])
RESULT not attacker (SecK []) is true.
--------------------------------------------------------------------------------
-- Query not attacker(SK[])
Completing...
Starting query not attacker(SK[])
RESULT not attacker (SK []) is true.
--------------------------------------------------------------------------------
-- Query event(termMD(hfid)) ==> event(acceptsFS(hfid))
Completing...
Starting query event(termMD(hfid)) ==> event(acceptsFS(hfid))
RESULT event (termMD (hfid)) ==> event (acceptsFS (hfid)) is true.
--------------------------------------------------------------------------------
-- Query inj-event(termFS(hsid)) ==> inj-event(acceptsSD(hsid))
Completing...
Starting query inj-event(termFS(hsid)) ==> inj-event(acceptsSD(hsid))
RESULT inj-event (termFS (hsid)) ==> inj-event (acceptsSD (hsid)) is true.
--------------------------------------------------------------------------------
-- Query inj-event(termSD(huid)) ==> inj-event(acceptsMD(huid))
Completing...
Starting query inj-event(termSD(huid)) ==> inj-event(acceptsMD(huid))
RESULT inj-event (termSD (huid)) ==> inj-event (acceptsMD (huid)) is true.
--------------------------------------------------------------------------------
```

Fig. 9 Simulation results using ProVerif

6.1 Overhead analysis of the proposed TAKM-FC scheme

The overhead analysis of the proposed scheme is presented based on computation, communication and storage cost. Later the result analysis is carried out by comparing with the existing schemes. The cryptographic operations used in the scheme have been implemented using Python 3.8 using the respective libraries such as NumPy and pycrypto to find the execution time of each function. The experiment is conducted for 10 times and recorded the average time consumed in milliseconds. Table 4 shows the time consumption for each cryptographic primitive.

Table 4 Computation cost parameters

Parameter	Description	Time (seconds)
Th	Hash function	0.0026
*T*sym	AES symmetric encryption	0.378
Tfe	Fuzzy extractor operation	0.0171
Tm	ECC multiplication	0.0171

6.1.1 Computation cost

It is the total amount of time consumed to execute a task in the device, the device can be an edge device, fog server, or cloud server. Any task in the device is compiled with a set of cryptographic primitives and other mathematical operations to attain the results. Among the phases of TAKM-FC, the authentication with key management phase is considered for computation cost analysis by leaving other phases as they are not performed regularly. The authentication with key management phase accumulated with various functional primitives such as concatenation, XOR operation, fuzzy extractor and one-way hash function. Here we have considered only the fuzzy extractor and one-way hash function for computation cost evaluation, as other primitives consume very least processing time. Considering the execution time of each primitive as shown in Table 4, a computation cost of the proposed scheme is computed as:

$1Tfe + 33Th = 0.0171 + 33(0.0026) = 0.1029$ s.

The analysis results show that the TAKM-FC has achieved 18.46%, 21.74%, 11.90% and 13.38% lesser computation overhead compared with existing schemes [33–36], respectively. The comparison is shown in Fig. 10. The analysis of computation overhead is carried out by varying the number of edge devices from 5 to 30 and results are recorded. Figure 11 shows the comparison of computation overhead. The computation cost overhead is minimal when compared to other schemes, even after it increases with an increase in the number of edge devices.

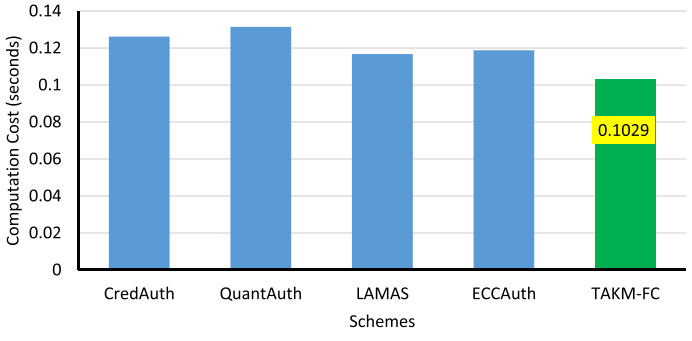

Fig. 10 Computation cost comparison

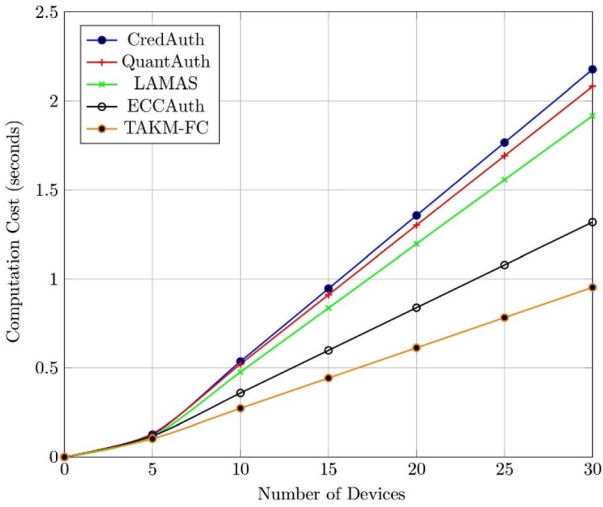

Fig. 11 Analysis of computation overhead

6.1.2 Communication cost

It is the number of bits consumed in the process of communication between the entities. In the authentication with key management process, we share three messages between MD*j*, FS, and SD*i*. Parameters considered for evaluating the communication overhead are shown in Table 5. The messages are evaluated with overhead as, the message from MDj to FS: $<$HUID*j*', N 3, T 3, T 4, *ts*3$>$ $=<160+320+160+160+32>=832$ bits. The message from FS to SD*i*: $<$HUID*j*, HSID*i*, *T 5, T 6, N 4, ts4*$>$ $=<160+160+160+160+320+32>=992$ bits. The message from SD*i* to MD*j*: $<$HSID*i* * *, *T 7, T 8, N 4, ts5*$>$ $=<160+160+160+320+32>=832$ bits. So the total number of bits shared between the entities are: $832+992+832=2656$ bits.

The results show that the proposed TAKM-FC achieved 19.29%, 11.17%, 8.79% and 15.44% lesser communication overhead compared with existing schemes [33–36], respectively. The comparison is shown in Fig. 12 and shows as it consumes less communication overhead between edge devices and fog server. The analysis of

Table 5 Communication cost parameters

Parameter	Description	Length (bits)
UID*j*	User ID	128
PWD*j*, PPWD*j*	Password's	128
H **	Hashed ID's	160
K,SecK, SK	Keys used	160
T1–T8	Temporary values	160
N3–N4	Private random numbers	320

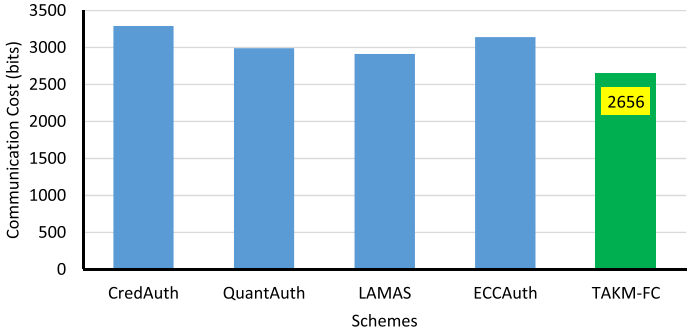

Fig. 12 Communication cost comparison

communication overhead is carried out by varying the number of edge devices from 5 to 30 and results are recorded. Figure 13 shows the comparison of communication overhead. The communication cost overhead is minimal when compared to other schemes, even after it increases with an increase in the number of edge devices.

6.1.3 Storage cost

It is considered to be the total bits stored in the local storage of any device consisting of generated or received credentials useful for communication. Here, we have considered the storage of the smart device, mobile device and fog server for analysis. The size of parameters in each device is based on Table 5. The MDj will be storing the credentials as $< \text{HUID}j, \text{TC}j, \text{PPWD}j, (\text{TFID}, \text{TC}), \beta j, R >$ at the registration

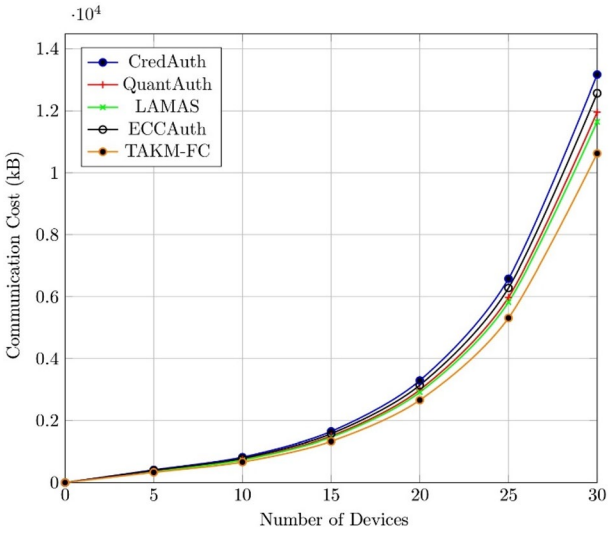

Fig. 13 Analysis of communication overhead

phase and SK*ij* at the authentication phase. So storage bits at mobile device is $<160+160+128+160+160+320>=1088+160=1248$ bits.

The SD*i* will be storing $<$TSID*i*, HSID*i*, TC*i*, *F* (TSID*i*, *y*), SecK$>$ at the registration phase and SK*ji* at the authentication phase. So total storage bits in the smart device is $<160+160+160+160+160>=800+160=960$ bits.

The FS stores $<$TFID, HFID, TC, *F* (TFID, *y*), HUID*j*, *R*, SecK*i*$>=<160+160+160+160+320+160>=1120$ bits. The storage cost of the TAKM-FC is 3328 bits and compared with other schemes as shown in Fig. 14. The analysis results show that TAKM-FC has achieved 13.87%, 10.44%, 6.62% and 7.91% lesser storage overhead compared with existing schemes [33–36], respectively. It is proven to be the mobile device, smart device and fog server in the proposed scheme consume very less storage. The analysis of storage overhead is carried out by varying the number of edge devices from 5 to 30 and results are recorded. The comparison of computation overhead is shown in Fig. 15.

6.2 Performance evaluation using iFogSim

The proposed TAKM-FC is implemented in the iFogSim simulator [45] for evaluating the performance of the work. The performance metrics considered for analysis are throughput, end-to-end delay, rate of packet loss, energy consumption and network usage. The topology in the simulation is created by having a cloud server and one fog server with set of mobile devices and a set of smart devices. The number of smart devices and a number of user mobile devices are considered in three scenarios having 20, 30 and 40 smart devices with variable user mobile devices from 1 to 5. The simulation parameters used in iFogSim are given in Table 6.

For all the considered scenarios, we deliberate the simulation performance metrics: throughput, end-to-end delay, rate of packet loss, energy consumption and network usage.

Throughput Is the number of units of work processed or a number of requests made per second from the concurrent users who are using the same application system at the same time. It is found using Eq. 12.

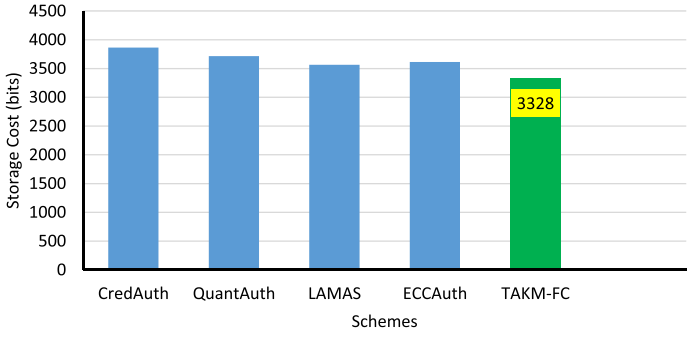

Fig. 14 Storage cost comparison

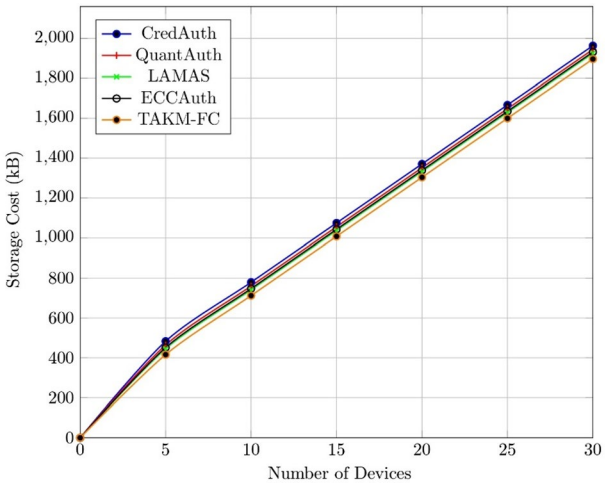

Fig. 15 Analysis of storage overhead

Table 6 Parameter definitions

Parameter	Cloud	Fog
Level	0	1
RAM(MB)	40,000	4000
CPU(MIPS)	44,800	2800
Downlink BW(MB)	10,000	10,000
Uplink BW(MB)	100	10,000
CostPerMIPS	0.01	0.0
Idle Power(watt)	16*83.25	83.4333
Busy Power(watt)	16*103	107.339

$$\text{Throughput}(bps) = \frac{\text{Total number of packets} \times \text{size of each packet(bits)}}{\text{Total time taken(s)}} \tag{12}$$

It is observed in Fig. 16 that an increase in the number of devices in communication increases the total number of messages being exchanged and increases the throughput. It was observed that there is a variation in throughput of about 11.26–11.80% for 1 user with 10–30 smart devices, 7.0–6.2% for 2 users with 10–30 smart devices, 6.14–6.43% for 3 users with 10–30 smart devices, 12.36–11.00% for 4 users with 10–30 smart devices and 20.69–17.22% for 5 users with 10–30 smart devices.

End-to-End Delay It is an average time taken for communications to reach their destination from the respective source node. It is calculated using Eq. 13.

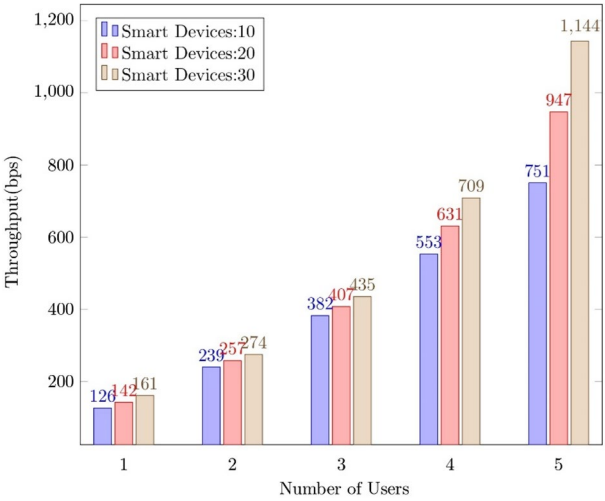

Fig. 16 Throughput versus number of users

$$\text{Delay} = \frac{\text{Packet Arrival Time} - \text{Packet Sent Time}}{\text{Total messages}} \tag{13}$$

This is required to build a session key between the communicating parties and is crucial to monitor for authentication with the key management method. For a more efficient authentication method with a smaller number of users, a delay should have a lower value. As shown in Fig. 17, the value of delay increases when increase in the number of users, as it turn they increases the number of messages exchanged. It was

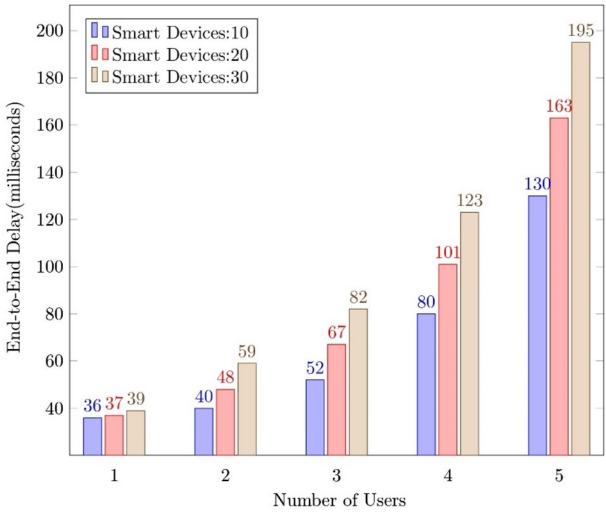

Fig. 17 End-to-end delay versus number of users

observed that there is an variation in delay of about 2.7–5.12% for 1 user with 10–30 smart devices, 16.6–18.6% for 2 users with 10–30 smart devices, 22.3–18.2% for 3 users with 10–30 smart devices, 20.7–17.8% for 4 users with 10 to 30 smart devices and 20.2–16.4% for 5 users with 10–30 smart devices.

Packet Loss It is the number of packets that get dropped while two devices were communicating. It is calculated using an Eq. 14.

$$\text{Packet LossRate} = \frac{\text{Total packets sent} - \text{Total packets recived}}{\text{Total packets sent}} \tag{14}$$

As shown in Fig. 18, the packet loss rate rises when the increase in the number of users, owing to the fact that increase in congestion as more messages are exchanged. It was observed that there is a variation in packet loss rate of about 11.11–50.0% for 1 user with 10–30 smart devices, 38.0–52.5% for 2 users with 10–30 smart devices, 45.9–49.33% for 3 users with 10–30 smart devices, 60.5–70.2% for 4 users with 10–30 smart devices and 72.8–75.7% for 5 users with 10–30 smart devices.

Energy Consumption It is the total energy utilized based on the power supply to the entire environment including cloud, fog, and edge devices. As shown in Fig. 19, the energy consumption increases when increase in the number of smart and mobile devices. It was observed that there is a variation in energy consumption of about 33.8–63.6% for 1 user with 10–30 smart devices, 34.3–63.8% for 2 users with 10–30 smart devices, 34.7–63.9% for 3 users with 10–30 smart devices, 35.1–64.08% for 4 users with 10–30 smart devices and 35.7–64.3% for 5 users with 10–30 smart devices.

Network Usage Network usage is the utilization of nodes and the occurrence of traffic while communicating between the nodes in the network. It is depicted

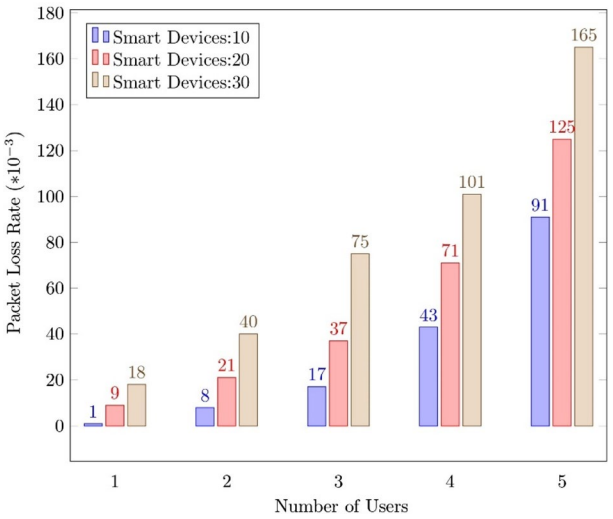

Fig. 18 Rate of packet loss versus number of users

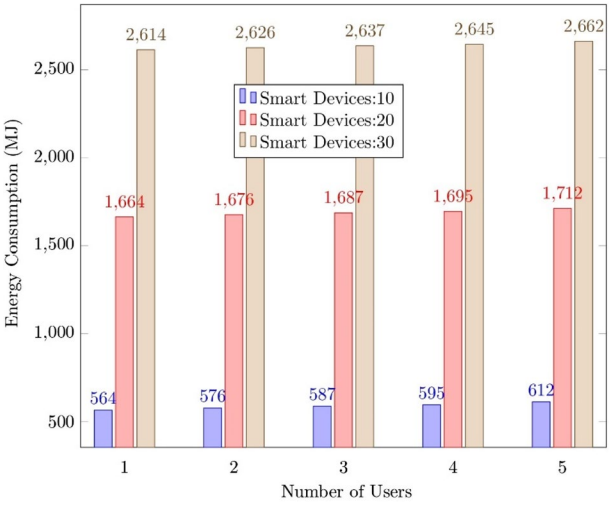

Fig. 19 Overall energy consumption

in Fig. 20. The traffic at the edge layer increases when increase in the number of smart and mobile devices, in turn the traffic at fog server also increases relatively. It was observed that there is a variation in network usage of about 84.4–42.5% for 1 user with 10–30 smart devices, 78.2–33.09% for 2 users with 10–30 smart devices, 80.4–39.8% for 3 users with 10–30 smart devices, 81.04–64.7% for 4 users with 10–30 smart devices and 81.1% to 67.9% for 5 users with 10 to 30 smart devices.

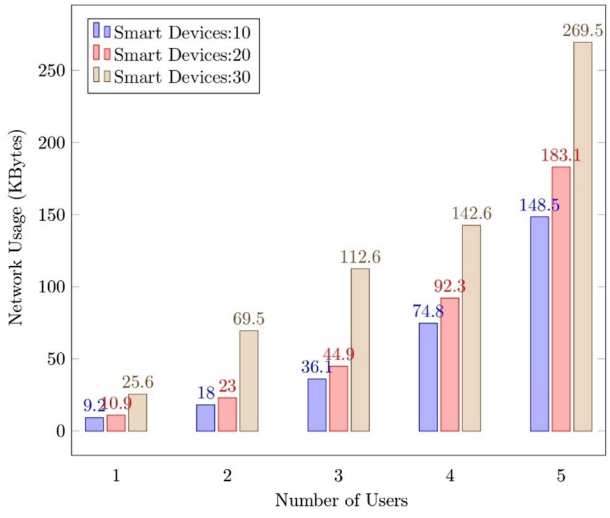

Fig. 20 Network usage

7 Conclusions and future work

The fog computing environment is more prone to security issues when the number of edge devices is increasing in digital communications. The insecurity activities can be controlled by providing multi-level and mutual authentication between the edge devices. This paper proposes a multi-level and two-way authentication with effective key management between the smart device and mobile device. A final session key is generated at both edge devices after the successful authentication between them. The proposed work made use of fuzzy extractor function for efficient user login to the mobile device, cryptographic primitives are used during key management and authentication. Security analysis is carried out considering the mathematical ROR model and theoretical information analysis and also TAKM-FC is verified for some of the known attacks by using the security verification tool ProVerif. We have also illustrated how the proposed scheme is efficient compared to other schemes by discussing the computation, communication, and storage cost overhead. The proposed scheme has been implemented in iFogSim and verified throughput, end-to-end delay, packet loss, energy consumption, and network usage.

In future, the work will be extended with blockchain technology to increase the efficiency in key management between edge devices. The blockchain provides a distributed platform to store and manage the key pair that achieves anonymity, authenticity, nonframeability, and unforgeability. Also, the semantic-based method of access control extended to the next level of security by having the history of the user's request and by designing fine-grained policies based on heterogeneous devices in a real social network is needed to study other factors such as trust.

Author contributions NCG-Conceptualization, Methodology, Data curation, Experimentation, Writing—Original draft preparation. SSM-Conceptualization, Methodology, Visualization, Investigation, Validation, Supervision. BMA-Methodology, Investigation, Validation, Writing—Reviewing and Editing, Supervision. RB-Validation, Writing—Reviewing and Editing.

Funding The authors did not receive any funds, grants, or other support for conducting this study, preparation of this manuscript, and submitting the work.

Data availability Data sharing is not applicable to this article as no datasets were generated during the current study.

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. The authors have no conflict of interest to declare that are relevant to the content of this article.

Ethical approval Ethical approval is not required and not applicable to publish this paper.

References

1. Namasudra S, Roy P (2018) PpBAC. J Organ End User Comput 30:14–31. https://doi.org/10.4018/joeuc.2018100102
2. Xiong H, Wang Y, Li W, Chen C-M (2019) Flexible, efficient, and secure access delegation in cloud computing. ACM Trans Manag Inf Syst 10:1–20. https://doi.org/10.1145/3318212
3. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing-MCC'12. https://doi.org/10.1145/2342509.2342513
4. Singh SP, Nayyar A, Kumar R (2019) Sharma A, Fog computing: from architecture to edge computing and big data processing. J Supercomput 75:2070–2105. https://doi.org/10.1007/s11227-018-2701-2
5. Manvi SS, Gowda NC (2019) Trust management in fog computing. Appl Integr Tech Methods Distrib Syst Technol. https://doi.org/10.4018/978-1-5225-8295-3.ch002
6. Murtaza MH, Tahir H, Tahir S, Alizai ZA, Riaz Q, Hussain M (2022) A portable hardware security module and cryptographic key generator. J Inf Secur Appl 70:103332. https://doi.org/10.1016/j.jisa.2022.103332
7. Mehdi M, Ajani MT, Tahir H, Tahir S, Alizai Z, Khan F, Riaz Q, Hussain M (2021) PUF-based key generation scheme for secure group communication using MEMS. Electronics 10:1691. https://doi.org/10.3390/electronics10141691
8. Stojmenovic I, Wen S (2014) The fog computing paradigm scenarios and security issues. Ann Comput Sci Inf Syst. https://doi.org/10.15439/2014f503
9. Kaliya N, Pawar D (2023) Unboxing fog security: a review of fog security and authentication mechanisms. Computing. https://doi.org/10.1007/s00607-023-01208-3
10. Al-Mekhlafi ZG, Al-Shareeda MA, Manickam S, Mohammed BA, Alreshidi A, Alazmi M, Alshudukhi JS, Alsaffar M, Rassem TH (2023) Efficient authentication scheme for 5G-enabled vehicular networks using fog computing. Sensors 23:3543. https://doi.org/10.3390/s23073543
11. Luqman M, Faridi AR (2023) Authentication of fog-assisted IoT networks using advanced encryption credibility approach with modified Diffie–Hellman encryption. Concurr Comput. https://doi.org/10.1002/cpe.7742
12. Saravanakumar S, Saravanan T (2023) Secure personal authentication in fog devices via multimodal rank-level fusion. Concurr Comput. https://doi.org/10.1002/cpe.7673
13. Loffi L, Westphall CM, Grudtner LD, Westphall CB (2021) Mutual authentication with multifactor in IoT-Fog-Cloud environment. J Netw Comput Appl 176:102932. https://doi.org/10.1016/j.jnca.2020.102932
14. Mo J, Hu Z, Chen H, Shen W (2019) An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing. Wirel Commun Mob Comput 2019:1–12. https://doi.org/10.1155/2019/4520685
15. Kumar P, Braeken A, Gurtov A, Iinatti J, Ha PH (2017) Anonymous secure framework in connected smart home environments. IEEE Trans Inform Forensic Secur 12:968–979. https://doi.org/10.1109/tifs.2016.2647225
16. Braeken A, Kumar P, Liyanage M, Hue TTK (2017) An efficient anonymous authentication protocol in multiple server communication networks (EAAM). J Supercomput 74:1695–1714. https://doi.org/10.1007/s11227-017-2190-8
17. Odelu V, Das AK, Wazid M, Conti M (2016) Provably Secure authenticated key agreement scheme for smart grid. IEEE Trans Smart Grid. https://doi.org/10.1109/tsg.2016.2602282
18. Guo J, Du Y, Zhang Y, Li M (2021) A provably secure ECC-based access and handover authentication protocol for space information networks. J Netw Comput Appl 193:103183. https://doi.org/10.1016/j.jnca.2021.103183
19. Al Hamid HA, Rahman SMM, Hossain MS, Almogren A, Alamri A (2017) A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography. IEEE Access 5:22313–22328. https://doi.org/10.1109/access.2017.2757844
20. Ke C, Zhu Z, Xiao F, Huang Z, Meng Y (2022) SDN-based privacy and functional authentication scheme for fog nodes of smart healthcare. IEEE Internet Things J. https://doi.org/10.1109/jiot.2022.3161935

21. Wu TY, Lee Z, Yang L (2021) Provably secure authentication key exchange scheme using fog nodes in vehicular ad hoc networks. J Supercomput 77:6992–7020. https://doi.org/10.1007/s11227-020-03548-9
22. Chen C-M, Huang Y, Wang K-H, Kumari S, Wu M-E (2020) A secure authenticated and key exchange scheme for fog computing. Enterprise Information Systems 15:1200–1215. https://doi.org/10.1080/17517575.2020.1712746
23. Tiwari D, Chaturvedi GK, Gangadharan GR (2019) ACDAS: Authenticated controlled data access and sharing scheme for cloud storage. Int J Commun Syst 32:e4072. https://doi.org/10.1002/dac.4072
24. Akram MA, Ghaffar Z, Mahmood K, Kumari S, Agarwal K, Chen C-M (2020) An anonymous authenticated key-agreement scheme for multi-server infrastructure. Hum Cent Comput Inf Sci. https://doi.org/10.1186/s13673-020-00227-9
25. Liu C-L, Tsai W-J, Chang T-Y, Liu T-M (2018) Ephemeral-secret-leakage secure ID based three-party authenticated key agreement protocol for mobile distributed computing environments. Symmetry 10:84. https://doi.org/10.3390/sym10040084
26. Patonico S, Braeken A, Steenhaut K (2019) Identity-based and anonymous key agreement protocol for fog computing resistant in the Canetti–Krawczyk security model. Wireless Netw. https://doi.org/10.1007/s11276-019-02084-6
27. Wu TY, Meng Q, Yang L, Guo X, Kumari S (2022) A provably secure lightweight authentication protocol in mobile edge computing environments. J Supercomput 78:13893–13914. https://doi.org/10.1007/s11227-022-04411-9
28. Alsahlani AYF, Popa A (2021) LMAAS-IoT: Lightweight multi-factor authentication and authorization scheme for real-time data access in IoT cloud-based environment. J Netw Comput Appl 192:103177. https://doi.org/10.1016/j.jnca.2021.103177
29. Wazid M, Das AK, Kumar N, Vasilakos AV (2019) Design of secure key management and user authentication scheme for fog computing services. Futur Gener Comput Syst 91:475–492. https://doi.org/10.1016/j.future.2018.09.017
30. Yadav AK, Braeken A, Misra M (2023) Symmetric key-based authentication and key agreement scheme resistant against semi-trusted third party for fog and dew computing. J Supercomput. https://doi.org/10.1007/s11227-023-05064-y
31. Yan X, Ma M (2021) A lightweight and secure handover authentication scheme for 5G network using neighbour base stations. J Netw Comput Appl 193:103204. https://doi.org/10.1016/j.jnca.2021.103204
32. Wazid M, Bagga P, Das AK, Shetty S, Rodrigues JJPC, Park Y (2019) AKMIoV: authenticated key management protocol in fog computing-based internet of vehicles deployment. IEEE Internet Things J 6:8804–8817. https://doi.org/10.1109/jiot.2019.2923611
33. Ali HS, Sridevi R (2022) Credential-based authentication mechanism for IoT devices in fog-cloud computing. ICT Anal Appl. https://doi.org/10.1007/978-981-16-5655-2_30
34. Lu S, Li X (2021) Quantum-resistant lightweight authentication and key agreement protocol for fog-based microgrids. IEEE Access 9:27588–27600. https://doi.org/10.1109/access.2021.3058180
35. Hamada M, Salem SA, Salem FM (2022) LAMAS: Lightweight anonymous mutual authentication scheme for securing fog computing environments. Ain Shams Eng J 13:101752. https://doi.org/10.1016/j.asej.2022.101752
36. Chatterjee U, Ray S, Khan MK, Dasgupta M, Chen C-M (2022) An ECC based lightweight remote user authentication and key management scheme for IoT communication in context of fog computing. Computing 104:1359–1395. https://doi.org/10.1007/s00607-022-01055-8
37. Dolev D, Yao A (1983) On the security of public key protocols. IEEE Trans Inform Theory 29:198–208. https://doi.org/10.1109/tit.1983.1056650
38. Das AK, Sengupta I (2008) An effective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials. In: 2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08). https://doi.org/10.1109/comswa.2008.4554370
39. Dodis Y, Reyzin L, Smith A (2004) Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In: Proceedings of the advances in cryptology (Eurocrypt'04), LNCS, vol 3027
40. Wang D, He D, Wang P, Chu C-H (2015) Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. IEEE Trans Dependable Secure Comput 12:428–442. https://doi.org/10.1109/tdsc.2014.2355850

41. Burrows M, Abadi M, Needham R (1990) A logic of authentication. ACM Trans Comput Syst 8:18–36. https://doi.org/10.1145/77648.77649
42. Wazid M, Das AK, Odelu V, Kumar N, Susilo W (2020) Secure remote user authenticated key establishment protocol for smart home environment. IEEE Trans Dependable Secure Comput 17:391–406. https://doi.org/10.1109/tdsc.2017.2764083
43. Wu L, Wang J, Choo K-KR, He D (2019) Secure key agreement and key protection for mobile device user authentication. IEEE Trans Inform Forensics Secur 14:319–330. https://doi.org/10.1109/tifs.2018.2850299
44. Blanchet B, Smyth B, Cheval V, Sylvestre M (2018) ProVerif 2.00: automatic cryptographic protocol verifier, user manual and tutorial. p 05–16
45. Awaisi KS, Abbas A, Khan SU, Mahmud R, Buyya R (2021) Simulating fog computing applications using iFogSim toolkit. Mob Edge Comput. https://doi.org/10.1007/978-3-030-69893-5_22

Authors and Affiliations

Naveen Chandra Gowda[1,2] · Sunilkumar S. Manvi[2] · A. Bharathi Malakreddy[3] · Rajkumar Buyya[4]

✉ Naveen Chandra Gowda
ncgowdru@gmail.com

Sunilkumar S. Manvi
ssmanvi@reva.edu.in

A. Bharathi Malakreddy
bharathi_m@bmsit.in

Rajkumar Buyya
rbuyya@unimelb.edu.au

[1] BMS Institute of Technology and Management, Bengaluru, India

[2] School of Computer Science and Engineering, REVA University, Bengaluru, India

[3] Department of AI and ML, BMS Institute of Technology and Management, Bengaluru, India

[4] School of Computing and Information Systems, University of Melbourne, Melbourne, VIC 3010, Australia