Computer Networks 185 (2021) 107731

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A clogging resistant secure authentication scheme for fog computing services

Zeshan Ali [a], Shehzad Ashraf Chaudhry [b], Khalid Mahmood [c], Sahil Garg [d], Zhihan Lv [e], Yousaf Bin Zikria [f,*]

[a] Department of Computer Science and Software Engineering International Islamic University, Islamabad, Pakistan
[b] Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey
[c] Department of Computer Science, Comsats University Islamabad, Sahiwal Campus, 57000, Pakistan
[d] Electrical Engineering Department, École de technologie supérieure, Montréal, QC H3C 1K3, Canada
[e] School of Data Science and Software Engineering, Qingdao University, Qingdao, China
[f] Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, South Korea

ARTICLE INFO

Keywords:
Authentication
Authentication protocol
Key management
Access control
Fog computing

ABSTRACT

Fog computing (FC) is an infrastructure consisting of decentralized computing, where computing resources such as storage, applications, and data are scattered among the cloud and data source. Fog computing inherits similar privacy and security concerns present in cloud computing, such as authentication and key management issues. Recently, Wazid et al. presented a scheme of authentication key exchange for fog computing called SAKA-FC to address these issues. We analyzed and identified that the SAKA-FC suffers from some severe vulnerabilities. Furthermore, we presented an improved scheme to mitigate these problems while retaining its strengths. The formal security analysis of the proposed scheme is validated through BAN logic. At the same time, the AVISPA tool is employed for automated formal security verification. Informal security analysis is conducted to attest that the proposal can confront the known attacks. Using computation and communication costs as the metrics, the proposed scheme is also compared with some state-of-the-art schemes. The proposed scheme achieves the same communication cost as of SAKA-FC, whereas the difference in computation cost is 24%. This increase in computation cost is justifiable as the proposal is resistant to clogging attacks and provides better security than the prior schemes.

1. Introduction

In fog computing, computational resources are distributed geographically and are decentralized [1]. Fog computation provides the computational resources as a service, the same as cloud computing, by employing identical service design. To ensure efficient resource utilization and management, fog computing uses similar technologies like cloud computing such as containers, virtualization, etc. [2,3]. Cloud computing consists of high capacity data centers. In contrast to this, fog computing comprises of geographically distributed resources with average capacity named fog nodes [4], which are closer to edge-devices as depicted in Fig. 1. This method renders a better experience to end-users, decreases service latency, and improves the Quality of service (QoS) [5–11]. Fog supports a broad range of future technologies and applications like artificial intelligence (AI) and the Internet of Things (IoT) [12]. Unlike cloud computing, fog computing is deemed more secure because the data is gathered and analyzed at local nodes, and various security checks are applied at different nodes. Various security checks make it harder for the attackers/hackers to have illegitimate access to data, whereas, in cloud computing, data is placed in a central location [13,14]. Since fog computing is an augmentation of cloud computing, and various security checks are applied at different nodes still, fog computing bears numerous privacy and security issues of cloud computing, causing severe concerns. An adversary can perform different sorts of attacks like forgery, impersonation, a man-in-the-middle, spoofing, spoofing, online/offline guessing of user passwords, ill wicked privileged insider/s, and the physical seizing of smart devices as the communication is over public (insecure) channels. So, to ensure that solely authorized users can access the system resources and ill/wicked adversary can be prevented from accessing the system, a robust authentication protocol needs to be employed.

Caiza et al. [15] discussed the various challenges of the Industrial Internet of Things (IIoT) to current infrastructure. How fog computing aims to solve these challenges and reduce energy consumption in industrial sensor networks, the problem of big data, processing and storage of real-time data, and enhancement in security. Caiza et al. also reviewed recent research regarding the latency, security, architecture,

* Corresponding author.
E-mail address: yousafbinzikria@ynu.ac.kr (Y.B. Zikria).

https://doi.org/10.1016/j.comnet.2020.107731
Received 28 August 2020; Received in revised form 16 November 2020; Accepted 2 December 2020
Available online 7 December 2020
1389-1286/

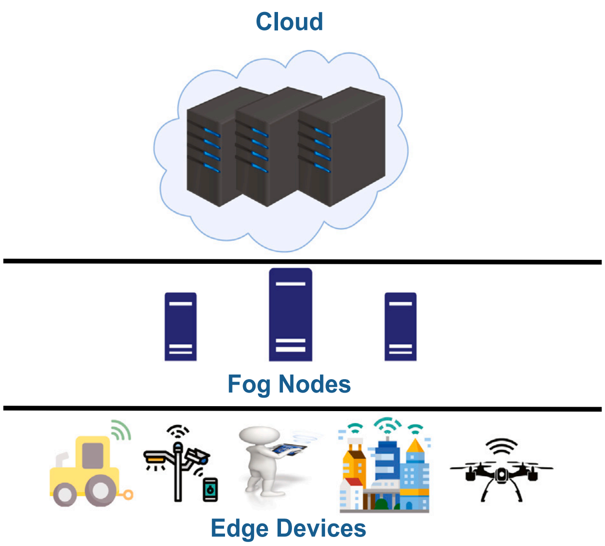

Fig. 1. General network model of fog computing environment.

and energy consumption that fog computing offers at the industrial level and present a survey of the contemporary features and challenges of this innovative technology. Qiu et al. [16] have discussed some state-of-the-art research related to intelligent security and optimization in fog computing. Lee et al. [17] have analyzed the privacy and security issues of fog computing, where the main focus is a man-in-the-middle attack, data protection, and management issues, and malicious and intrusion detection techniques, but a specific solution was absent. To prevent leakage of personal images in cloud computing, Xia et al. [18] proposed a scheme for image retrieval based on privacy preservation content. The pixels of the image were encrypted by employing the standard stream cipher. k-NN algorithm is used to secure images extracted security features, and water-mark protocol is used to obviate the illegal usage. Such schemes have remarkably improved biometric and face identification based on cloud computing while assuring security and privacy. Although the obstacle of bandwidth was yet not solved. To overcome the issues mentioned earlier, Hu et al. [19] introduced a face identification & resolution framework based on fog computing. In the proposed scheme, they offloaded the computational overload from cloud to fog nodes (FNs) by implementing task partitioning tactics. To benefit the computation and storage capability, the matching algorithm for face identifier and data storage is performed on the cloud. Simultaneously, the algorithms for image pre-processing, image detection, feature extraction, and generation of facial image identifiers are performed on fog nodes. Gope [20] introduced an authentication protocol to address multiple differing circumstances in the D2D-aided fog computing and introduced a privacy preserving security architecture. In the proposed scheme, end devices are verified without the involvement of the centralized authority. Alemneh et al. [21] introduced a two party trust management scheme based on subjective logic that permits the service provider to verify the requester's trustworthiness and facilitates a service requester to check whether a service provider can provide secure and stable services. The system is also resilient to a vast population of misbehaving nodes and can prevent trust-based attacks. To secure communication amongst the edge nodes and cloud, Alrawais et al. [22] introduced a key exchange protocol based on ciphertext policy attribute-based encryption (CP-ABE). It merged it with a digital signature technique to obtain verifiability, confidentiality, access control, and authentication. Another ECC based key exchange scheme proposed by Khan et al. [23] was proved as incorrect by Chaudhry [24]. Similarly, Irshad et al. also proposed two party key exchanges through an intermediate agent.

1.1. Motivation and contributions

Recently, Wazid et al. [25] introduced a lightweight scheme using symmetric hash functions and elliptic curve cryptography (ECC), for smart devices in fog computing termed as SAKA-FC. Wazid et al. averred that their scheme is robust against several known attacks, including replay, privileged insider, offline guessing, server and/or smart device impersonation attacks, amongst many others. However, after careful analysis, we identified that Wazid et al.'s SAKA-FC suffers from traceability attack, clogging attack, and employs useless parameters. We then proposed an improved scheme using symmetric key bases hash functions and Elliptic Curve Cryptography primitives to secure fog computing based architectures. The scheme resists clogging and related attacks. The rest of the paper is structured as follows: Section 1.2 explains the adopted adversarial model. SAKA-FC is reviewed in Section in Section 2, and the cryptanalysis of the same is performed in Section 3. Proposed scheme for fog computing is presented in Section 4. The Security and performance analysis of the proposed scheme related to other schemes is performed in Sections 5 and 6. Finally, the paper is concluded in Section 7.

1.2. Adversarial model

The well-known (DY) adversarial model [26] as utilized in [27–31] is considered in this paper. Where an adversary ($\mathcal{A}$) deemed to be equipped with subsequent capabilities:

1. Two parties communicate over the public channel and end-points are not trusted.
2. $\mathcal{A}$ has full control over the public communication channel.
3. $\mathcal{A}$ can alter or discard a message transmitting over the public channel and can also forge a message.
4. Private/secret key of the Trusted Authority (TA)/Central Authority (CA) can't be compromised.

Moreover, CK adversarial model [32], and eCK model [33] are also considered along with the DY model. As per the CK-adversary model, an adversary can also compromise the confidential credentials and the session keys and states in the sessions. Additionally, $\mathcal{A}$ can also capture the smart devices and perform a power analysis attack [34,35] to obtain stored information. As per the eCK model, the attacker is also allowed to launch a critical compromise impersonation attack.

2. Review of Wazid et al.'s scheme

The essential phases of the scheme proposed by Wazid et al. [25] are described in the subsequent subsections; whereas, various notations useful to understand technical details are listed in Table 1.

2.1. Pre-deployment processes

In this phase, fog servers, smart devices, and cloud servers are registered with the Trusted Authority (TA) before they are deployed in the network.

2.1.1. Smart devices registration process

TA selects identity ID_k, temporary identity TID_k for each smart device and computes $RID_k = h(K \parallel ID_k)$, $TC_k = h(K \| RTS_k \| ID_k)$ where K is the secret key of TA and RTS_k being the registration timestamp of smart device. TA then picks $F(x, y) = \sum_{m,n=0}^{t} a_{m,n} x^m y^n$ distinct symmetric bivariate polynomial of degree t over a Galois finite field ($GF(p)(= Z_p)$) and computes $F(TID_k, y) = \sum_{m,n=0}^{t} [a_{m,n}(TID_k)^m] y^n$ a polynomial share. Finally, the parameters $\{RID_k, TID_k, TC_k, F(TID_k, y)\}$ are stored in D_k's memory prior deployment in the field.

Table 1
Notations guide.

Symbols	Representations
U_i, MD_i	ith user and his/her mobile device
ID_i, PW_i, BIO_i	ith user's identity, password and biometric
D_k, ID_k	kth smart device and its identity
FS_j, ID_j	jth fog server and its identity
CS_l, ID_l	lth cloud server and its identity
RTS_k, RTS_j, RTS_l	Registration timestamp of smart device, fog server, and cloud server, respectively
r_i, r_f, r_k	Random numbers
K	1024 bit long secret key of TA
$d_i,\ P_i = d_i.G$	Private/public key pair of ith entity
$SK_{ik}(= SK_{ki})$	Session key between U_i and D_k
$h(.)$	Cryptographic one way hash function
TS_i, TS_f, TS_k	Current timestamps
$Gen(.), Rep(.)$	Fuzzy generation & reproduction functions
$\triangle T$	Delay tolerance
$i \stackrel{?}{=} j$	Relational equality check
$\oplus, \|\|$	EX-OR and concatenation operators
$\Longrightarrow, \longrightarrow$	Public and private channels
$\mathcal{I}, \mathcal{A}, U_{\mathcal{A}}$	Alternative notations used for adversary

2.1.2. Fog servers registration process

For each fog server FS_j, TA picks distinct identity ID_j and temporary identity TID_j and computes $F(TID_j, y) = \sum_{m,n=0}^{t}[a_{m,n}(TID_j)^m]y^n$, $TC_j = h(K\|RTS_j\|ID_j)$, and $RID_j = h(K \parallel ID_j)$ for each FS_j where RTS_j is the registration timestamp of FS_j and stores the parameters $\{\{RID_i|i = 1, 2, \ldots, n_i\}, (RID_j, TID_j, TC_j, F(TID_j, y))\}$ in a database of FS_j prior deployment.

2.1.3. Cloud servers registration processes

For each cloud server CS_l, TA picks a distinct identity ID_l, temporary identity TID_l and computes $RID_l = h(K \parallel ID_l)$, $TC_l = h(K\|RTS_l\|ID_l)$ where RTS_l is the registration timestamp of the CS_l. TA picks $G_{j,l}(x, y) = \sum_{m,n=0}^{t} b_{m,n}x^m y^n \in GF(p)[x, y]$ a unique symmetric bivariate polynomial of degree t for each pair of (CS_l, FS_j). TA then computes $F(TID_l, y) = \sum_{m,n=0}^{t}[a_{m,n}(TID_l)^m]y^n$ for each pair of (CS_l, FS_j) and stores the parameters $\{(RID_l, TID_l, TC_l), \{G_{j,l}(TID_l, y)|j = 1, 2, \ldots, n_f\}\}$ in cloud server CS_l's database and $\{G_{j,l}(TID_l, y)|l = 1, 2, \ldots, n_c\}$ in FS_j's database for each CS_l.

2.2. Key management (KM) process

This phase reviews Wazid et al.'s process of the key sharing between a smart device and a fog server as well as between a fog server and a cloud server.

2.2.1. KM for smart devices and fog servers

Subsequent are the steps performed over an insecure public channel to establish a secret key amongst D_k and FS_j:

1. D_k first picks an arbitrary nonce r_1 and present timestamp TS_1, calculates $r_1' = h(r_1\|TC_k\|TS_1)$ and transmits the message containing $\{TID_k, r_1', TS_1\}$ to FS_j.
2. On receiving $\{TID_k, r_1', TS_1\}$ from D_k, FS_j first checks the message freshness through verifying $|TS_1 - TS_1^*| \leq \triangle T$, if true FS_j picks a random nonce r_2 and selects present timestamp TS_2 and computes $AA_j = h(r_2 \parallel TC_j) \oplus h(F(TID_j, TID_k)\|r_1'\|TS_2)$, $F(TID_j, TID_k)$, $K_{jk} = h(F(TID_j, TID_k)\|r_1'\|\ h(r_2 \parallel TC_j)\ \|TS_2)$, $BB_j = h(K_{jk}\|TS_2)$ and sends the message containing $\{TID_j, AA_j, BB_j, TS_2\}$ over the public channel to D_k.
3. On receiving $\{TID_j, AA_j, BB_j, TS_2\}$ from FS_j, D_k checks the message freshness through verifying $|TS_2 - TS_2^* \leq \triangle T|$. If true, D_k calculates $F(TID_k,\ TID_j)$, $h(r_2 \parallel TC_j) = AA_j \oplus h(F(TID_k, TID_j)\|r_1'\|\ TS_2) = AA_j \oplus h(F(TID_j, TID_k)\|r_1'\|TS_2)$, $K_{kj} = h(F(TID_k, TID_j)\|r_1'\|h(r_2\|TC_j)\|TS_2)$, $BB_j' = h(\ K_{kj} \parallel TS_2)$ and checks whether $BB_j' \stackrel{?}{=} BB_j$, if true D_k uses the secret key $K_{kj}(= K_{jk})$ for future communication else discards the key.

2.2.2. KM for fog servers and cloud servers

Subsequent are the steps performed to establish a secret key amongst a fog server FS_j and a cloud server CS_l:

1. FS_j selects present timestamp TS_3 and an arbitrary nonce r_3, computes $r_3' = h(r_3\|TC_j\|TS_3)$, and transmits the message containing $\{TID_j, r_3', TS_3\}$ to CS_l via public channel.
2. On receiving $\{TID_j, r_3', TS_3\}$ from FS_j CS_l first checks the message freshness through verifying $|TS_3 - TS_3^* \leq \triangle T|$. If true CS_l picks current timestamp TS_4, an arbitrary number r_4 and calculates $G_{j-l}(TID_l, TID_j)$, $CC_l = h(r_4 \parallel TC_l) \oplus h(G_{j-l}(TID_l, TID_j)\|r_3'\|TS_4)$, $K_{lj} = h(G_{j-l}(TID_l,\ TID_j)\|r_3'\|\ h(r_4 \parallel TC_l) \parallel TS_4)DD_l = h(K_{lj} \parallel TS_4)$. Finally, CS_l transmits the message containing $\{TID_l, CC_l, DD_l, TS_4\}$ to FS_j over the public channel.
3. On receiving $\{TID_l, CC_l, DD_l, TS_4\}$ from CS_l, FS_j first checks the message freshness through verifying $|TS_4 - TS_4^*| \leq \triangle T$, if true FS_j calculates $G_{j-l}(TID_j, TID_l) = G_{j,l}(TID_l, TID_j), h(r_4 \parallel TC_l) = CC_l \oplus h(G_{j-l}(TID_j,\ TID_l)\|r_3'\|TS_4), K_{jl} = h(G_{j-l}(TID_j, TID_l)\|r_3'\|h(r_4 \parallel TC_l)\|TS_4), DD_l' = h(K_{jl}\|TS_4)$ and checks whether $DD_l' \stackrel{?}{=} DD_l$. If true, FS_j uses the secret key $K_{jl}(= K_{lj})$ to communicate securely with CS_l.

2.3. User registration process

In their scheme, Wazid et al. employed Elliptic-curve cryptography (ECC) by selecting a curve $E_p(\alpha, \beta)$ and a point $G \in E_p(\alpha, \beta)$. Moreover, Wazid et al. employed fuzzy extractor to implement biometric-authentication which comprises a pair of functions; (i) probabilistic random generation $Gen(.)$ and (ii) deterministic reproduction $(Rep(.))$. Where $Gen(.)$ takes personal biometric of a user and produces the arbitrary $l - bit$ long key $\sigma_i \in \{0, 1\}^l$, and a public reproduction parameter τ_i. While, the $Rep(.)$ takes τ_i along with user biometrics and inspects whether the variation among old and new biometrics is $\leq$ to the error tolerance threshold (t). $Rep(.)$ reproduces the genuine biometric as a return value.

If a user U_i wants to access the smart device D_k he/she needs to register first. Following is the procedure adopted by a U_i to register with the TA:

1. U_i picks an identity ID_i, arbitrary secret number s, a private key $d_i \in Z_p^*$ and calculates $RID_i = h(s \parallel ID_i), P_i = d_i.G$. Finally, U_i sends the message containing $\{RID_i,\ P_i\}$ to TA over a private channel.
2. Upon receiving the message from U_i, TA computes $TC_i = h(RID_i\|K\|RTS_i)$ where RTS_i is the registration timestamp of the U_i. Finally TA sends the reply containing $\{TC_i, \{(TID_j, h(TC_j))|j = 1, 2, \ldots, n_f\}\}$ to U_i over the private channel.
3. Upon receiving the reply from TA, U_i chooses PW_i and imprints BIO_i. The associated device MD_i computes $Gen\ (BIO_i) = (\sigma_i, \tau_i)$, $TC_i^* = TC_i \oplus h(ID_i \parallel \sigma_i)$, $d_i^* = d_i \oplus h(ID_i\|PW_i\|\sigma_i)$, $RID_i^* = RID_i \oplus h(d_i \parallel \sigma_i)$, $TC_j^* = h(TC_j) \oplus h(RID_i \parallel \sigma_i)$, and $RPD_i = h(ID_i \parallel TC_i\|PW_i\|\sigma_i)$ for $j = 1, 2, \ldots, n_f$. Finally, MD_i overwrites the information $\{s, RID_i, d_i, TC_i, \{h(TC_j)|j = 1, 2, \ldots, n_f\}\}$ with $\{RID_i^*, d_i^*, TC_i^*, RPB_i^*, \{(TID_j, TC_j^*)|j = 1, 2, \ldots, n_f\}, P_i, \tau_i, Gen(.), Rep(.), h(.), t\}$.

2.4. Login process

Subsequent are the steps performed by the U_i in order to login through MD_i and access D_k:

1. U_i submits $\{ID_i, PW_i\}$ pair along with BIO_i' to MD_i. The MD_i calculates $\sigma_i' = Rep(BIO_i', \tau_i)$, $TC_i = TC_i^* \oplus h(ID_i \parallel \sigma_i')$, $d_i = d_i^* \oplus h(ID_i\|PW_i\|\sigma_i')$, $RID_i = RID_i^* \oplus h(d_i \parallel \sigma_i')$, $RPB_i = h(ID_i\|TC_i\|PW_i \parallel \sigma_i')$, and checks whether $RPB_i' \stackrel{?}{=} RPB_i$. If true, next step is computed else session terminates.

2. MD_i further computes $h(TC_j) = TC_j^* \oplus h(RID_i \parallel \sigma_i')$, $R_i = r_i.G$, $a_i = d_i + r_i(mod\,p)$, $RID_i' = RID_i \oplus h(h(TC_j) \parallel TS_i)$, $F_i = RID_k \oplus h(h(TC_j) \parallel TS_i)$ and $E_i = h(TC_i \| d_i \| TS_i) \oplus h(h(TC_j) \parallel RID_i)$. Finally, MD_i transmits the message containing $M_{sg_1} = \{RID_i', R_i, a_i, E_i, F_i, TS_i\}$ to FS_j via public channel.

2.5. Authentication and key agreement process

Subsequent are the steps performed by U_i, FS_j and D_k to establish a session key once U_i sends login request:

1. Upon receiving the M_{sg_1}, FS_j first checks the message freshness by validating the condition $|TS_i - TS_i^* \leq \triangle T|$. If true FS_j computes $RID_i = RID_i' \oplus h(h(TC_j) \parallel TS_i)$ and checks whether it is present in the database and checks the validity of $a_i.G = P_i + R_i$. If both are true FS_j picks an arbitrary number r_f and present timestamp TS_f, calculates $K_{uf} = r_f.R_i = (r_i.\, r_f).G$, $P_f = r_f.G$, $RID_k = F_i \oplus h(h(TC_j) \parallel TS_i)$, $RID_k^* = RID_k \oplus h(K_{jk} \parallel TS_f)$, $RID_i^* = h(RID_i) \oplus h(K_{jk} \| RID_k \| TS_f)$, $h(TC_i \| d_i \| TS_i) = E_i \oplus h(h(TC_j) \parallel RID_i)$, $G_j = h(K_{jk} \| RID_k \| TS_f) \oplus h(K_{uf} \parallel h(TC_i \| d_i \| TS_i) \parallel h(RID_i))$, and $H_j = h(h(RID_i) \parallel RID_k \parallel G_j \| P_f \parallel TS_f)$. Finally, TS_j transmits the message containing $M_{sg_2} = \{RID_i^*, RID_k^*, G_j, H_j, P_f, TS_f\}$ to D_k.
2. On receiving M_{sg_2}, the D_k first checks the message freshness by validating the condition $|TS_f - TS_f^* \leq \triangle T|$. If true, D_k calculates $RID_k = RID_k^* \oplus h(K_{jk} \parallel TS_f), h(RID_i) = RID_i^* \oplus h(K_{jk} \| RID_k \| TS_f), H_j' = h(h(RID_i) \parallel RID_k \parallel G_j \| P_f \parallel TS_f)$, and verifies whether $H_j' \stackrel{?}{=} H_j$ if false session terminates. If true, D_k picks an arbitrary number r_k, present timestamp TS_k and calculates $I_j = G_j \oplus h(K_{jk} \| RID_k \| TS_f) = h(K_{uf} \parallel h(TC_i \parallel d_i \parallel TS_i) \parallel h(\, RID_i))$, $M_k = h(TC_k \parallel r_k) \oplus h(RID_k \parallel h(RID_i) \parallel TS_k)$, $SK_{ki} = h(I_j \parallel h(TC_k \| r_k \| TS_k))$, $RID_k^{**} = RID_k \oplus h(h(RID_i) \parallel TS_k)$, and $N_k = h(SK_{ki} \| P_f \| TS_k)$. Finally, D_k transmits the message containing $M_{sg_3} = \{RID_k^{**}, M_k, N_k, P_f, TS_k\}$ to U_i.
3. On receiving M_{sg_3}, U_i first checks the message freshness by validating the condition $|TS_k - TS_k^* \leq \triangle T|$. If true, U_i calculates $RID_k = RID_k^{**} \oplus h(h(RID_i) \parallel TS_k), h(TC_k \parallel r_k) = M_k \oplus h(RID_k \parallel h(RID_i) \parallel TS_k)$, $K_{uf} = r_i.P_f, SK_{ik} = h(h(K_{uf} \| h(TC_i \| d_i \| TS_i) \|\; h\; (RID_i)) \| h(TC_k \parallel r_k) \| TS_k) (=\, SK_{ki})$, and $N_k' = h(\, SK_{ik} \| P_f \| TS_k)$. Finally, U_i verifies the condition $N_k' \stackrel{?}{=} N_k$, if true $SK_{ik} (= SK_{ki})$ is used as a session key among the U_i and D_k for safe communication.

3. Cryptanalysis of Wazid et al.'s scheme

In this section, we show that the scheme of Wazid et al. (SAKA-FC) is vulnerable to traceability and clogging attacks. Moreover, their scheme has a useless parameter E_i transmitted over the insecure channel along with another insecure parameter F_i. The details are given in the following subsections:

3.1. Traceability attack

This attack can be simulated by considering an attacker $\mathcal{A}$, who registers with the system and gets its' mobile device MD_A engraved with $\{RID_a^*, TC_a^*, d_a^*, RPB_a, \{(TID_j, TC_j^*)|j = 1, 2..n_f\}, P_a, \tau_a\}$. Following steps shows the simulation of traceability attacks:

Step TA 1: $\mathcal{U}_A$ enters his identity, password and biometrics tuple $\{ID_a, PW_a, BIO_a\}$ and computes $\sigma_a' = Rep(BIO_a', \tau_a)$, $TC_a = TC_a^* \oplus h(ID_a \parallel \sigma_a')$, $d_a = d_a^* \oplus h(ID_a \parallel PW_a \parallel \sigma_a')$, $RID_i = RID_i^* \oplus h(d_i \parallel \sigma_i')$ and:

$$h(TC_j) = TC_j^* \oplus h(RID_i \parallel \sigma_i') \tag{1}$$

Step TA 2: Now $\mathcal{U}_A$ waits for login request message from any system user. Let U_i initiates login message by sending $M_{sg1} = \langle RID_i', R_i, a_i, E_i, F_i, TS_i\rangle$ to FS_j.

Step TA 3: $\mathcal{U}_A$ intercepts M_{sg1} and by using extracted $h(TC_j)$ and captured RID_i', TS_i computes:

$$RID_i = RID_i' \oplus h(h(TC_j) \parallel TS_i) \tag{2}$$

In Eq. (2), the RID_i is the alias identity of the original user U_i. The RID_i remains same for all sessions. Hence, a dishonest legal user $\mathcal{U}_A$ has successfully launch traceability attack.

3.2. Clogging attack

Through a clogging attack, [36,37], an active attacker can force a legitimate fog server and the smart device to process the attacker's fake request masquerading himself as a legitimate user of the system. It leads towards the resource clogging of both the fog server and the smart device, and this attack can represent a significant class of Denial of Service (DoS) and/or degradation in Quality of Service (QoS) attacks [38,39]. Wazid et al.'s scheme are also insecure against the clogging attack as per the forthcoming simulation in this subsection. Referring to the preceding Section 3.1, an attacker $\mathcal{U}_A$ can extract the generic parameter $h(TC_j)$ from his mobile device after getting registered with the system. $\mathcal{U}_A$ can now launch a clogging attack and can deceive both the fog server and the smart device on the fly. The attack can be simulated as follows:

Step CA 1: $\mathcal{U}_A$ waits for login request message from any system user. Let U_i initiates login message by sending
$M_{sg1} = \langle RID_i', R_i, a_i, E_i, F_i, TS_i\rangle$ to FS_j.

Step CA 2: $\mathcal{U}_A$ intercepts M_{sg1} and by using extracted $h(TC_j)$ and captured RID_i', computes $RID_i = RID_i' \oplus h(h(TC_j) \parallel TS_i)$ as shown in Eq. (2).
$\mathcal{U}_A$ now waits for the session termination and can launch user impersonation attack any time using a_i along with $h(TC_j)$ and RID_i.

Step CA 3: $\mathcal{U}_A$ generates 160 bit variable $\mathcal{Z}$ randomly, new time stamp TS_{ua} and computes:

$$RID_i' = RID_i \oplus h(h(TC_j) \parallel TS_{ua}) \tag{3}$$

$$E_{ua} = Z \oplus h(h(TC_j) \parallel RID_i) \tag{4}$$

$$F_{ua} = RID_k \oplus h(h(TC_j) \parallel TS_{ua}) \tag{5}$$

Step CA 4: $\mathcal{U}_A$ now sends the request message $M_{sg1} = \langle RID_i', R_i, a_i, E_{ua}, F_{ua}, TS_{ua}\rangle$ to FS_j, where the pair $\{R_i, a_i\}$ is the previously captured from original message by U_i.

Step CA 5: FS_j receives the request and verifies the freshness of timestamp, as the time stamp TS_{ua} is freshly generated. Therefore, FS_j computes $RID_i = RID_i' \oplus h(h(TC_j) \parallel TS_{ua})$ and verifies:

$$a_i.G = P_i + R_i? \tag{6}$$

As, the pair $\{a_i, R_i\}$ was generated genuinely by U_i in preceding session. Therefore, $\mathcal{U}_A$ passes this test. Therefore, FS_j processes the fake request by $\mathcal{U}_A$.

Step CA 6: FS_j now generate r_f, TS_f and computes $K_{uf} = r_f.R_i = (r_i r_f).G$, $Z = E_{ua} \oplus h(h(TC_j) \parallel RID_i)$, $P_f = r_f.G$, $RID_k = F_i \oplus h(h(TC_j) \parallel TS_i)$, $RID_i^* = h(RID_i) \oplus h(K_{jk} \| RID_k \| TS_f)$, $RID_k^* = RID_k \oplus h(K_{jk} \parallel TS_f)$, $G_j = h(K_{jk} \| RID_k \| TS_f) \oplus h(K_{uf} \| Z \| h(RID_i))$ and $H_j = h(h(RID_i) \| RID_k \| G_j \| P_f \| TS_f)$

Step CA 7: FS_j further sends $M_{sg2} = \langle RID_i^*, RID_k^*, G_j, H_j, P_f, TS_f\rangle$ to the smart device D_k.

Step CA 8: On receiving M_{sg2}, the D_k checks time freshness of TS_f, as it is generated freshly, so this test is passed. D_k further computes $RID_k = RID_k^* \oplus h(K_{jk} \parallel TS_f)$, $h(RID_i) = RID_i^* \oplus h(K_{jk} \| RID_k \| TS_f)$, and $H_j' = h(h(RID_i) \| RID_k \| G_j \| P_f \| TS_f)$.

Now D_k checks $H'_j \stackrel{?}{=} H_j$, if it holds, D_k consider both the FS_j and U_i as legitimate ones. As all these parameters $\{RID_i, RID_k, G_j, P_f, TS_f\}$ used in computation of H_j are generated by legitimate fog server FS_j. So this test is also passed.

Step CA 9: Now, D_K computes and sends other parameters $M_{sg3} = \langle RID_k^{**}, M_k, N_k, P_f, TS_k\rangle$ to U_i.

In the simulation steps above, we have seen that the attacker $\mathcal{U}_A$ can send a forged request on behalf of another user and forced both the fog server and the smart device to process this message. Both the server and device consider $\mathcal{U}_A$ as the legitimate U_i and shared a session key with U_i. This is a fact that $\mathcal{U}_A$ could not be able to compute the correct session key as he doesn't have access to r_i. Still, he has successfully launched a clogging attack, which can further degrade the system performance and significantly cause a DoS attack.

3.3. Useless/insecure parameters

E_i is a useless parameter in Wazid et al.'s scheme, as it can be easily extracted by the attacker using computed $h(TC_j)$ through Eq. (1) and RID_i through Eq. (2) as described in Section 3.1. Therefore, the attacker $\mathcal{U}_A$ can compute the hidden parameter on the fly, shown as follows:

$$h(TC_i\|d_i\|TS_i) = E_i \oplus h(h(TC_j) \parallel RID_i) \tag{7}$$

Moreover, the timestamp TS_i is sent in plain text; therefore, the attacker can efficiently compute the identity RID_k of D_k, shown as follows:

$$RID_k = F_i \oplus h(h(TC_j) \parallel TS_i) \tag{8}$$

Hence, the E_i is a useless parameter and F_i is an insecure parameter.

4. Proposed scheme

In this section, an improved scheme is presented to overcome the vulnerabilities present in Wazid et al.'s scheme.

4.1. Pre-deployment processes

In this phase, cloud and fog servers and smart devices are registered with the Trusted Authority (TA) before they are deployed in the network.

4.1.1. Cloud servers registration process

For each cloud server CS_l, TA picks a distinct identity ID_l and computes $d_l = h(K \parallel ID_l)$ as the private key of CS_l. TA then stores $\{ID_l, d_l\}$ in the memory of cloud server and deploys it in the network.

4.1.2. Fog servers registration process

For each fog server FS_j, TA picks distinct identity ID_j, selects private key $d_j = h(ID_j \parallel d_l)$ as per the corresponding cloud server CS_l and computes public key $P_j = d_j.G$. Now, TA stores $\{ID_j, d_j, P_j\}$ in fog server's memory as well as sends $\{ID_j, P_j\}$ to corresponding CS_l and the TA publicizes the pair $\{ID_j, P_j\}$.

4.1.3. Smart devices registration process

TA selects unique identity ID_k for each smart device and as per the corresponding FS_j computes $d_k = h(d_j\|ID_k\|ID_j)$. Finally, the parameters $\{ID_k, d_k\}$ are stored in D_k's memory prior deployment. Moreover, FS_j is informed about ID_k deployment and FS_j stores ID_k in it's memory. The same method is adopted when a device dynamically enters into the system.

4.2. Key management (KM) process

This phase describes the key management between a smart device and a fog server. Moreover, the key management between the fog server and cloud server is also explained here.

4.2.1. KM for smart devices and fog servers

Subsequent are the steps performed over an insecure public channel to establish a secret key amongst D_k and FS_j:

1. D_k first picks an arbitrary nonce r_1 and timestamp TS_1, calculates $R_1 = r_1.G$, $\overline{R}_1 = r_1.P_j$ and $\overline{r}_1 = h(R_1\|TS_1\|d_k)$ and transmits the message containing $\{ID_k, \overline{R}_1, \overline{r}_1, TS_1\}$ to FS_j.
2. Upon receiving this message, FS_j first checks the freshness of the message by checking the condition $|TS_1 - TS_1^*| \leq \triangle T$, if true FS_j calculates $R_1 = \overline{R}_1.d_j^{-1}$, $d_k = h(d_j\|ID_k\|ID_j)$ and checks $\overline{r}_1 \stackrel{?}{=} h(R_1\|TS_1\|d_k)$ and on success picks a random nonce r_2, present timestamp TS_2 and calculates $R_2 = r_2.G$, $\overline{R}_2 = r_2.P_k$, $K_{jk} = h(R_1\|R_2\|TS_2)$ and $\overline{r}_2 = h(R_2\|TS_2\|K_{jk})$. FS_j now sends the message containing $\{ID_j, \overline{R}_2, \overline{r}_2, TS_2\}$ to D_k.
3. Upon receiving the message from FS_j, D_k checks the freshness of the timestamp by examining the condition $|TS_2 - TS_2^* \leq \triangle T|$. If true, D_k calculates $R_2 = \overline{R}_2.d_k^{-1}$ and computes $K_{jk} = h(R_1\|R_2\|TS_2)$. Now D_k checks $\overline{r}_2 \stackrel{?}{=} h(R_2\|TS_2\|K_{jk})$. On success, D_k stores K_{jk} in it's memory for future secure communication.

4.2.2. KM for fog servers and cloud servers

Subsequent are the steps performed over an insecure public channel to establish a secret key amongst FS_j and CS_l:

1. FS_j first picks an arbitrary nonce r_3 and timestamp TS_3, calculates $R_3 = r_3.G$, $\overline{R}_3 = r_3.P_l$ and $\overline{r}_3 = h(R_3\|TS_3\|d_j)$ and transmits the message containing $\{ID_j, \overline{R}_3, \overline{r}_3, TS_3\}$ to CS_l.
2. On receiving $\{ID_j, \overline{R}_3, \overline{r}_3, TS_3\}$ message, CS_l first checks the message freshness by validating the condition $|TS_3 - TS_3^*| \leq \triangle T$, if true CS_l calculates $R_3 = \overline{R}_3.d_l^{-1}$, $d_l = h(K \parallel ID_l)$ and checks $\overline{r}_3 \stackrel{?}{=} h(R_3\|TS_3\|d_l)$ and on success picks a random nonce r_4, present timestamp TS_4 and calculates $R_4 = r_4.G$, $\overline{R}_4 = r_4.P_l$, $K_{lj} = h(R_3\|R_4\|TS_4)$ and $\overline{r}_4 = h(R_4\|TS_4\|K_{lj})$. CS_l now sends the message containing $\{ID_l, \overline{R}_4, \overline{r}_4, TS_4\}$ to FS_j.
3. On receiving the message containing $\{ID_l, \overline{R}_4, \overline{r}_4, TS_4\}$ from CS_l, FS_j first checks the message freshness by validating the condition $|TS_4 - TS_4^* \leq \triangle T|$. If true, FS_j calculates $R_4 = \overline{R}_4.d_j^{-1}$ and computes $K_{jl} = h(R_4\|R_4\|TS_4)$. Now FS_l checks $\overline{r}_4 \stackrel{?}{=} h(R_4\|TS_4\|K_{lj})$. On success, FS_j stores K_{jl} in it's memory for future secure communication.

4.3. User registration process

If a user U_i wants to access the smart device D_k he/she needs to register first. Following is the procedure as depicted in Fig. 2, adopted by a U_i to register with the TA:

1. U_i picks a unique ID_i, a private key $d_i \in Z_p^*$ and calculates $P_i = d_i.G$. Finally, U_i sends the message containing $\{ID_i, P_i\}$ to TA using secure channel.
2. On receiving $\{ID_i, P_i\}$ from U_i, TA computes $TC_i = h(ID_i \parallel K)$. Then TA sends the reply containing $\{TC_i, \{ID_k | k = 1, 2...n_d\}, \{ID_j, P_j | j = 1, 2, \ldots, n_f\}\}$ to U_i using secure channel.
3. On receiving the reply containing $\{TC_i, \{ID_k | k = 1, 2...n_d\}, \{ID_j, P_j | j = 1, 2, \ldots, n_f\}\}$ from TA, U_i chooses password PW_i and imprints BIO_i. Next U_i calculates $Gen\ (BIO_i) = (\sigma_i, \tau_i)$, $d_i^* = d_i \oplus h(ID_i\|PW_i\|\sigma_i)$, $TC_i^* = TC_i \oplus h(ID_i \parallel \sigma_i)$, $RPB_i =$

U_i	TA
Picks ID_i and private-key $d_i \in \mathcal{Z}_p^*$ and Compute. $P_i = d_i.G$.	
$\xrightarrow[(U_i \rightarrow TA)]{ID_i, P_i}$	
	Compute $TC_i = h(ID_i\|\|K)$, Store $\{ID_j, P_j \| j = 1, 2, ..., n_f\}$ $\{ID_k \| k = 1, 2, ..., n_d\}$ Stores and publicizes $\{ID_i, P_i\}$
	$\xleftarrow[(U_i \leftarrow TA)]{TC_i, ID_k, ID_j, P_j}$
Pick PW_i and input BIO_i Compute. $Gen(BIO_i) = (\sigma_i, \tau_i)$ $TC_i^* = TC_i \oplus h(ID_i\|\|\sigma_i)$ $d_i^* = d_i \oplus h(ID_i\|\|PW_i\|\|\sigma_i)$ $ID_i^* = ID_i \oplus h(d_i\|\|\sigma_i)$ $RPB_i = h(ID_i\|\|TC_i\|\|PW_i\|\|\sigma_i)$ Replace $\{ID_i, d_i, TC_i\}$ $\{ID_i^*, d_i^*, TC_i^*, RPB_i, P_i, \{ID_k \| k = 1, 2, ..., n_d\}, \{(ID_j, P_j) \| j = 1, 2, ..., n_f\}, \tau_i, Gen(.), Rep(.), t, h(.)\}$	

Fig. 2. User registration process.

$h(ID_i\|TC_i\|PW_i \ \| \ \sigma_i)$, $ID_i^* = ID_i \oplus h(d_i \ \| \ \sigma_i)$. Finally, MD_i overwrites the information $\{ID_i, d_i, TC_i\}$ and now the device contains $\{ID_i^*, TC_i^*, d_i^*, RPB_i^*, P_i, \{ID_k | k = 1, 2...n_d\}, \{ID_j, P_j | j = 1, 2, \dots, n_f\}, \tau_i, Gen(.), Rep(.), h(.)\}$, where n_d are the number of device identities of the devices, and n_f are the number of registered fog servers. The information about the identities of both fog servers and smart devices, as well as the public keys of the fog server, are already public. In case the number is large, U_i can skip storing it in its memory and can use some trusted public repository each time it needs these values and keep only the information of the frequently accessed entities in its memory.

4.4. Login & authentication process

Subsequent are the steps as depicted in Fig. 3, performed by the U_i in order to log in through MD_i and to access D_k by establishing a mutual session key (between U_i/MD_i and D_k) after authenticating each other through the mediation of corresponding fog server FS_j:

1. Firstly, U_i submits $\{ID_i, PW_i\}$ and imprints BIO_i. Now, MD_i computes $\sigma_i' = Rep(BIO_i', \tau_i)$, $TC_i = TC_i^* \oplus h(ID_i \ \| \ \sigma_i')$, $d_i = d_i^* \oplus h(ID_i\|PW_i\|\sigma_i')$, $ID_i = ID_i^* \oplus h(d_i \ \| \ \sigma_i')$ and $RPB_i = h(ID_i\|TC_i\|PW_i \ \| \ \sigma_i')$, MD_i checks the condition $RPB_i' \stackrel{?}{=} RPB_i$, if true U_i provides ID_j, ID_k and MD_i fetches the P_j to corresponding ID_j. MD_i picks an arbitrary nonce r_i, a present timestamp TS_i and computes $R_i = r_i.G$, $\overline{R}_i = r_i P_j$, $a_i = TS_i.d_i + r_i$, and $\overline{ID}_i = ID_i \oplus h(R_i \ \| \ TS_i)$, $E_i = h(R_i\|\overline{R}_i\|a_i \ \| \ TS_i)$ $F_i = ID_k \oplus h(\overline{R}_i\|R_i\|TS_i)$. Finally MD_i transmits the message containing $M_{sg1} = \{\overline{ID}_i, \overline{R}_i, a_i, F_i, \ E_i, TS_i\}$ to FS_j over the public channel.
2. On receiving the M_{sg1}, the FS_j first checks the message freshness by validating the condition $|TS_i - TS_i^*| \leq \triangle T$. If true, MD_i computes $R_i = d_j^{-1}\overline{R}_i$, $ID_i = \overline{ID}_i \oplus h(R_i \ \| \ TS_i)$ and checks if $a_i.G = TS_i.P_i + R_i$ and $E_i \stackrel{?}{=} h(R_i\|\overline{R}_i\|a_i \ \| \ TS_i)$. On validity of both preceding conditions, FS_j picks an arbitrary nonce r_f, a present timestamp TS_f and further computes $K_{uf} = r_f.R_i = (r_i r_f).G$, $P_f = r_f.G$, $ID_k = F_i \oplus h(\overline{R}_i\|R_i\|TS_i)$, $d_k = h(d_j \ \| \ ID_k \ \| \ |ID_j)$, $ID_i^* = ID_i \oplus h(d_k\|ID_k\|TS_f)$, $\overline{ID}_k = ID_k \oplus h(d_k \ \| \ TS_f)$, $G_j = h(d_k\|ID_k\|TS_f) \oplus h(K_{uf} \ \| \ h(R_i \ \| \ TS_i) \ \| \ |ID_i)$ and $H_j = h(ID_i\|ID_k\|G_j\|P_f\|TS_f \ \| \ d_k)$. FS_j transmits the message $M_{sg2} = \{ID_i^*, \overline{ID}_k, P_f, H_j, G_j, TS_f\}$ to D_k over insecure channel.

Table 2
Postulates of BAN logic.

Rule	Meaning
$\frac{A\|\equiv A\stackrel{K}{\longleftrightarrow}B, A\triangleleft<X>_K}{A\|\equiv Y\|\sim K}$	Message-meaning rule
$\frac{A\|\equiv\#\{X\}, A\|\equiv B\|\sim X}{A\|\equiv B\|\equiv X}$	Nonce-verification rule
$\frac{A\|\equiv B, A\|\equiv C}{A\|\equiv(B,C)}$	Acceptance conjunction
$\frac{A\|\equiv B\|\equiv(X,Y)}{A\|\equiv B\|\equiv X}$	Belief rule
$\frac{A\|\equiv\#X}{A\|\equiv\#(X,Y)}$	Fresh conjuncatenation rule
$\frac{A\|\equiv B\|\equiv X, A\|\equiv B\|\Rightarrow X}{A\|\equiv X}$	Jurisdiction rule
$\frac{A\|\equiv\#\{X\}, A\|\equiv B\|\equiv X}{A\|\equiv A\stackrel{K}{\longleftrightarrow}B}$	Session key

Table 3
Notations of BAN logic.

Notation	Meaning
$A\| \equiv B$	A believes a statement B
$A \stackrel{K}{\longleftrightarrow} Y$	Share a key K between A and Y
$\#B$	B is fresh
$A \triangleleft B$	A sees B
$A\| \sim B$	A said B
$(B, C)_K$	B, C is hashed by key K
$\{B\}_K$	B is hashed with key K
$\langle B \rangle_K$	B is encrypted with key K

3. On receiving M_{sg2} from FS_j, D_k first checks the message freshness by validating the condition $|TS_f - TS_f^*| \stackrel{?}{\leq} \triangle T$. If true, D_k computes $ID_k = \overline{ID}_k \oplus h(d_k \ \| \ TS_f)$ and $ID_i = ID_i^* \oplus h(d_k\|ID_k\|TS_f)$, D_k verifies the authenticity of FS_j by examining the condition $H_j = h(ID_i\|ID_k\|G_j \ \| \ P_f\|TS_f\|d_k)$ if false, session terminates. If true, D_k picks an arbitrary nonce r_k, present timestamp TS_k and computes $I_j = G_j \oplus h(K_{uf}\|h(R_i \ \| \ TS_i)\|ID_i)$, $ID_k^* = ID_k \oplus h(ID_i\|TS_k\|I_j)$, $SK_{ki} = h(I_j\|r_k\|TS_k)$, $M_k = h(TC_k \ \| \ r_k) \oplus h(I_j)$ and $N_k = h(SK_{ki}\|P_f\|TS_k)$. Finally, D_k transmits the message $M_{sg3} = \{ID_k^*, M_k, N_k, P_f, TS_k\}$ to U_i via open channel.
4. On receiving M_{sg3} from D_k, U_i first checks the freshness of the message by validating the condition $|TS_k - TS_k^*| \leq \triangle T$. If true, U_i calculates $ID_k = ID_k^* \oplus h(ID_i \ \| \ TS_k)$, $K_{uf} = r_i.P_f$, $I_j = h(K_{uf}\|h(R_i \ \| \ TS_i)\|\overline{ID}_i)$, $r_k = M_k \oplus h(I_j)$, $SK_{ik} = h(I_j\|r_k\|TS_k)(= SK_{ki})$ and $N_k' = h(SK_{ik} \ \| \ P_f \ \| \ TS_k)$. Finally, U_i checks if $N_k' \stackrel{?}{=} N_k$, if true U_i saves the key $SK_{ik}(= SK_{ki})$.

5. Security analysis

This section presents the formal Burrows–Abadi– Needham logic [40] (BAN logic) based security analysis augmented through automated analysis and informal security discussion.

5.1. Formal BAN logic based authentication proof

The formal BAN logic is employed in this subsection to prove the authentication security of the proposed scheme.

5.1.1. Logical postulates & notations for BAN logic

The adopted logical postulates and notations of BAN logic with related meaning are given in Table 2. We used some formal notations to describe BAN logic which is given in Table 3:

U_i/MD_i	FS_j	D_k
Input ID_i, PW_i and BIO_i, Compute $\sigma_i^{'} = Rep(BIO_i^{'}, \tau_i)$,		
$TC_i = TC_i^* \oplus h(ID_i\|\|\sigma_i^{'})$, $d_i = d_i^* \oplus h(ID_i\|\|PW_i\|\|\sigma_i^{'})$, $RPB_i = h(ID_i\|\|TC_i\|\|PW_i\|\|\sigma_i^{'})$, $\boxed{RPB_i^{'} \overset{?}{=} RPB_i}$	$\|TS_i - TS_i^*\| \overset{?}{\leq} \triangle T$ than, Calculate $R_i = d_j^{-1}\overline{R}_i$ $ID_i = \overline{ID}_i \oplus h(R_i\|\|TS_i)$ $\boxed{a_i.G \overset{?}{=} TS_i.P_i + R_i}$ $\boxed{E_i \overset{?}{=} h(R_i\|\|\overline{R}_i\|\|a_i\|\|TS_i)}$	$\|TS_f - TS_f^*\| \overset{?}{\leq} \triangle T$ than,
If true, generate TS_i and r_i. Calculate. $R_i = r_i.G, \overline{R}_i = r_i.P_j$, $a_i = TS_i.d_i + r_i$ $\overline{ID}_i = ID_i \oplus h(R_i\|\|TS_i)$, $E_i = h(R_i\|\|\overline{R}_i\|\|a_i\|\|TS_i)$, $F_i = ID_k \oplus h(\overline{R}_i\|\|R_i\|\|TS_i)$, $\xrightarrow[(U_i \rightarrow FS_j \text{ via open channel})]{M_{sg1} = \langle \overline{ID}_i, \overline{R}_i, a_i, F_i, E_i, TS_i \rangle}$	If both true, than generate TS_f and r_f. Compute $K_{uf} = r_f.R_i = (r_i r_f).G$, $P_f = r_f.G$, $ID_k = F_i \oplus h(\overline{R}_i\|\|R_i\|\|TS_i)$, $d_k = h(d_j\|\|ID_k\|\|ID_j)$, $ID_i^* = ID_i \oplus h(d_k\|\|ID_k\|\|TS_f)$, $\overline{ID}_k = ID_k \oplus h(d_k\|\|TS_f)$, $G_j = h(d_k\|\|ID_k\|\|TS_f) \oplus h(K_{uf}\|\|h(R_i\|\|TS_i)\|\|ID_i)$ $H_j = h(ID_i\|\|ID_k\|\|G_j\|\|P_f\|\|TS_f\|\|d_k)$. $\xrightarrow[(FS_j \rightarrow D_k \text{ via open channel})]{M_{sg2} = \langle ID_i^*, \overline{ID}_k, G_j, H_j, P_f, TS_f \rangle}$	Calculate. $ID_k = \overline{ID}_k \oplus h(d_k\|\|TS_f)$ $ID_i = ID_i^* \oplus h(d_k\|\|ID_k\|\|TS_f)$, $H_j^{'} = h(ID_i\|\|ID_k\|\|G_j\|\|P_f\|\|TS_f\|\|d_k)$, $\boxed{H_j^{'} \overset{?}{=} H_j}$ Compute $I_j = G_j \oplus h(d_k\|\|ID_k\|\|TS_f)$ $= h(K_{uf}\|\|h(R_i\|\|TS_i)\|\|ID_i)$ Generate r_k, TS_k. Calculate $ID_k^* = ID_k \oplus h(ID_i\|\|TS_k\|\|I_j)$,
$\|TS_k - TS_k^*\| \overset{?}{\leq} \triangle T$ than, Calculate. $K_{uf} = r_i.P_f$. $I_j = h(K_{uf}\|\|h(R_i\|\|TS_i)\|\|ID_i)$ $ID_k = ID_k^* \oplus h(ID_i\|\|TS_k\|\|I_j)$, $r_k = M_k \oplus h(I_j)$, $SK_{ik} = h(I_j\|\|r_k\|\|TS_k)(= SK_{ki})$, $N_k^{'} = h(SK_{ik}\|\|P_f\|\|TS_k)$, $\boxed{N_k^{'} \overset{?}{=} N_k}$ store SK_{ik}		$SK_{ki} = h(I_j\|\|r_k\|\|TS_k)$, $M_k = r_k \oplus h(I_j)$, $N_k = h(SK_{ki}\|\|P_f\|\|TS_k)$. $\xleftarrow[(D_k \leftarrow U_i \text{ via open channel})]{M_{sg3} = \langle ID_k^*, M_k, N_k, P_f, TS_k \rangle}$
	$\xleftrightarrow[(U_i \leftrightarrow D_k)]{Save\ the\ session-key\ SK_{ik}(=SK_{ki})}$	

Fig. 3. Proposed scheme.

5.1.2. Security goal establishment

Subsequent are the established security goals of the BAN logic:

$G_1 : U_i \mid\equiv U_i \overset{SK_{ki}}{\longleftrightarrow} D_k$

$G_2 : U_i \mid\equiv D_k \mid\equiv U_i \overset{SK_{ki}}{\longleftrightarrow} D_k$

$G_3 : D_k \mid\equiv U_i \overset{SK_{ki}}{\longleftrightarrow} D_k$

$G_4 : D_k \mid\equiv U_i \mid\equiv U_i \overset{SK_{ki}}{\longleftrightarrow} D_k$

5.1.3. Messages generic form

Subsequent are the idealized transformation of our scheme:

$MSG_0 : U_i \rightarrow FS_j : ID_i \oplus h(R_i\|TS_f), r_i.P_j, TS_i.d_i + r_i, ID_k \oplus h(\overline{R_i}\|R_i\|TS_i), h(R_i\|\overline{R_i}\|a_i\|TS_i), TS_i$

$MSG_1 : FS_j \rightarrow D_k : (ID_i \oplus h(d_k\|ID_k\|TS_f), ID_k \oplus h(d_k\|TS_f), h(d_k\|ID_k\|TS_f) \oplus h(K_{uf}\|h(R_i\|TS_i)\|ID_i), h(ID_i\|ID_k\|G_j \| P_f\|TS_f\| d_k), r_f.G, TS_f)$

$MSG_2 : D_k \rightarrow U_i : (ID_k \oplus h(ID_i\|TS_k\|K_{uf}), r_k \oplus h(I_j), h(SK_{ki}\|P_f\|TS_k), r_f.G, TS_k)$

5.1.4. Messages idealized form

Following are the idealized transformation of our introduced scheme:

$MSG_0 : FS_j \rightarrow D_k : (\langle ID_i, TS_i\rangle_{R_i}, \langle r_i.P_j\rangle, \langle TS_i, d_i + r_i\rangle, \langle ID_k, TS_i\rangle_{(R_i,\overline{R_i})}, \langle a_i, TS_i\rangle_{(R_i,\overline{R_i})}, TS_i)$

$MSG_1 : FS_j \rightarrow D_k : (\langle ID_i\rangle_{FS_j \overset{d_k,ID_k}{\longleftrightarrow} D_k}, \langle ID_k\rangle_{FS_j \overset{d_k}{\longleftrightarrow} SD_k}, \langle d_k, ID_k\rangle_{(K_{uf},r_f,R_i)}, \langle ID_i, ID_k\rangle_{FS_j \overset{d_k}{\longleftrightarrow} D_k}, r_f.G, TS_f)$

$MSG_2 : D_k \rightarrow U_i : (\langle ID_i\rangle_{FS_j \overset{d_k}{\longleftrightarrow} D_k}, \langle ID_k\rangle_{FS_j \overset{d_k}{\longleftrightarrow} SD_k}, \langle d_k, ID_k\rangle_{(K_{uf},r_f,R_i)}, \langle ID_i, ID_k\rangle_{FS_j \overset{d_k}{\longleftrightarrow} D_k}, r_f.G, TS_f)$

5.1.5. Assumptions

$A_1 : U_i \mid\equiv \#(TS_k)$

$A_2 : D_k \mid\equiv \#(TS_f)$

$A_3 : FS_j \mid\equiv (FS_j \overset{d_k}{\longleftrightarrow} D_k)$

$A_4 : FS_j \mid\equiv (FS_j \overset{ID_k}{\longleftrightarrow} D_k)$

$A_5 : D_k \mid\equiv (Ui \overset{ID_k}{\longleftrightarrow} D_k)$

$A_6 : D_k \mid\equiv FS_j \mid\Rightarrow FS_j \mid\sim X$

$A_7 : U_i \mid\equiv D_k \mid\Rightarrow (U_i \overset{SK_{ki}}{\longleftrightarrow} D_k)$

$A_8 : U_i \mid\equiv (Ui \overset{ID_i}{\longleftrightarrow} D_k)$

The mutual authentication between U_i and D_k is proved using the following steps:

S_1: From MSG_1, we get:

$\langle D_k \triangleleft (ID_i)_{FS_j \overset{d_k,ID_k}{\longleftrightarrow} D_k}, (ID_k)_{FS_j \overset{d_k}{\longleftrightarrow} SD_k}, (d_k, ID_k)_{(K_{uf},r_f,R_i)}, (ID_i, ID_k)_{FS_j \overset{d_k}{\longleftrightarrow} D_k}, r_f.G, TS_f\rangle$

S_2: Based on S_1, Assumptions A_1, A_3, A_4 and message-meaning rule, we get:

$D_k \mid\equiv FS_j \mid\sim \langle (ID_i), (ID_k), (d_k, ID_k)_{(K_{uf},r_f,R_i)}, (ID_i, ID_k), r_f.G, TS_f\rangle$

S_3: Based on S_2, Nonce verification rule and Freshness rule, we get:

$D_k \mid\equiv FS_j \mid\equiv \langle (ID_i), (ID_k), (d_k, ID_k)_{(K_{uf},r_f,R_i)}, (ID_i, ID_k), r_f.G\rangle$

S_4: Based on S_3, Assumption A_6 and Jurisdiction rule, we get:

$D_k \mid\equiv \langle (ID_i), (ID_k), (d_k, ID_k)_{(K_{uf},r_f,R_i)}, (ID_i, ID_k), r_f.G\rangle$

S_5: Based on S_4 and Belief rule, we get:

$D_k \mid\equiv U_i \overset{SK_{ki}}{\longleftrightarrow} D_k$ **(Goal 3)**

S_6: Based on S_5, Session key rule, we get:

```
role role_USERS(USERS,TA,FOGSERVER,SMARTDEVICE:agent,Skuita:
symmetric_key,H:hash_func,SND,RCV:channel(dy))
played_by USERS
def=
 local State:nat, IDi,BRi,Ai,Fi,BIDi,Pi,TCi,IDj,Di,TSk,Ri,
 Dj,Pj,Randi, TSi,Rk,IDk,G,Rf,K,TSf,IDl,Dl:text,F:hash_func
 init State := 0
 transition
  1. State=0
   /\ RCV(start) =|> State':=1 /\ Di':=new() /\ Pi' := F
  (Di'.G)
  /\ SND({Pi'.IDi}_Skuita)
  2. State=1
   /\ RCV({H(IDi.K).IDk.IDj.F(H(IDj.H(K.IDl)).G)}_Skuita)
   =|> State':=2 /\ TSi':=new() /\ Randi':=new() /\ Ri':=F(Randi'
   .G) /\ Dj' := H(IDj.H(K.IDl)) /\ Pj' := F(Dj'.G)
   /\ BRi' := F(Randi'.Pj') /\ Ai' := F(F(TSi'.Di).Randi')
   /\ Fi' := xor(IDk,H(BRi'.Ri'.TSi')) /\ BIDi' := xor(IDi,
   H(Ri'.TSi'))
   /\ SND(BIDi'.BRi'.Ai'.Fi'.TSi')
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   /\secret(Ri',sec_1,{USERS})
   /\witness(USERS,FOGSERVER,auth_4,TSi')
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  5. State=2
   /\RCV(xor(IDk,H(IDi.TSk'.F(F(Ri.Rf).G))).xor(Rk',H(H(F
   (F(F((Randi.G).TSi).Rf).G).H(F((Randi.G).TSi).TSi).IDi))).H(H(
   H(F(F(F((Randi.G).TSi).Rf).G).H(F((Randi.G).TSi).TSi).IDi)
   .Rk'.TSk').F(Rf.G).TSk').F(Rf.G).TSk') =|> State':=3
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   /\ request(USERS,SMARTDEVICE,auth_6,TSk')
   /\ secret(Rf,sec_2,{FOGSERVER})
   /\ secret(Ri,sec_1,{USERS})
   /\ secret(Rk',sec_3,{SMARTDEVICE})
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end role
```

(a) Role specification for user (U)

```
role role_SMARTDEVICE(SMARTDEVICE,USERS,TA,FOGSERVER:agent,H:hash_
func,SND,RCV:channel(dy))
played_by SMARTDEVICE
def=
 local
  State:nat,IDi,BRi,Ai,Fi,BIDi,Pi,TCi,IDj,Di,TSk,Ri,Dj,Pj,Randi,TSi,
  Rk,IDk,G,Rf,K,TSf,IDl,Dl,Pf,Dk,SIDi,BIDk,Gj,Hj,Kuf,Ij,SIDk,Mk,Nk,
  SKki:text,F:hash_func
 init State := 0
 transition
  4. State=0
   /\RCV(xor(IDi,H(H(H(IDj.H(K.IDl)).IDk.IDj).IDk.TSf')).xor(IDk,H(H
   (H(IDj.H(K.IDl)).IDk.IDj).TSf')).xor(H(H(H(IDj.H(K.IDl)).IDk.IDj)
   .IDk.TSf'),H(F(F(Randi.Rf').G).H(F(H(IDj.H(K.IDl)).F(Randi'.F(H(
   IDj.H(K.IDl)).G))).TSi').IDi)).H(IDi.IDk.xor(H(H(H(IDj.H(K.IDl)).
   IDk.IDj).IDk.TSf'),H(F(F(Randi.Rf').G).H(F(H(IDj.H(K.IDl)).F(
   Randi'.F(H(IDj.H(K.IDl)).G)).TSi').IDi)).F(Rf'.G).TSf'.Dk').F(
   Rf'.G).TSf')=|>
  State':=1
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   /\ request(SMARTDEVICE,FOGSERVER,auth_5,TSf')
   /\ secret(Rf',sec_2,{FOGSERVER})
   /\ secret(Ri,sec_1,{USERS})
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   /\TSk':=new()/\Rk':=new()/\Pf':=F(Rf.G)/\Kuf':=F(F(Ri.Rf).G)/\Ri'
   :=F((Randi.G).TSi)/\Dl':=H(K.IDl)/\Dj':=H(IDj.Dl')/\Pj' := F(Dj'.
   G)/\BRi':=F(F(Randi.Pj'))/\Ij':=H(Kuf'.H(Ri'.TSi).IDi)/\SIDk':=
   xor(IDk,H(IDi.TSk'.Kuf'))/\Mk':=xor(Rk',H(Ij'))
   /\SKki':=H(Ij'.Rk'.TSk')
   /\Nk' := H(SKki'.Pf'.TSk')
   /\ SND(SIDk'.Mk'.Nk'.Pf'.TSk')
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   /\ witness(SMARTDEVICE,USERS,auth_6,TSk')
   /\ secret(Rk',sec_3,{SMARTDEVICE})
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end role
```

(b) Role specification for smart device (SD).

Fig. 4. Role specification for user and smart device.

$D_k| \equiv U_i| \equiv U_i \overset{SK_{ki}}{\longleftrightarrow} D_k$ **(Goal 4)**

S_6: From MSG_2, we get:

$U_i \triangleleft \langle U_i \overset{SK_{ki}}{\longleftrightarrow} D_k \rangle$

S_7: Based on S_6 and message-meaning rule, we get:

$U_i| \equiv D_k| \sim \langle U_i \overset{SK_{ki}}{\longleftrightarrow} D_k \rangle_{TS_k}$

S_8: Based on S_7, assumption A_2, Nonce verification rule and Freshness rule, we get:

$U_i| \equiv D_k| \equiv U_i \overset{SK_{ki}}{\longleftrightarrow} D_k$ **(Goal 2)**

S_9: Based on S_8, assumption A_7 and Jurisdiction rule, we get:

$U_i| \equiv U_i \overset{SK_{ki}}{\longleftrightarrow} D_k$ **(Goal 1)**

5.2. Formal automated analysis using AVISPA tool

In this subsection, we perform the automated security analysis of the proposed scheme through AVISPA simulation tool [41], which can verify the scheme's security against replay and man in middle attacks. Subsequent are steps of AVISPA simulation:

1. The HLPSL (High Level Protocol Specification Language) provides the role platform for the role-oriented implementation of the protocol/scheme steps in high level language, which is then interpreted into IF (Intermediate Format) through it translator HLPSL2IF [41].
2. The OF (Output Format) then performs the security verification using the interpreted IF.

The role specifications for user/mobile device (U_i/MD_i), smart device (D_k), trusted authority (TA), and fog server (FS_j) are depicted in Figs. 4(a), (b), 5(a), and (b), respectively. The roles for environment, session and goal along with the simulation results are depicted in Fig. 6.

The AVISPA results, as depicted in Fig. 6(b) and (c) prove the design robustness of the proposed scheme against the replay and man in middle attacks. The OFMC backend tested 1576 in 32.94 in 8 piles depth, whereas, through CL-AtSe backend, 5624 states were analyzed within 0.69 and 0.20 s translation and computation time spent for the respective backend process.

5.3. Informal security analysis

In this section, we have analyzed the proposed scheme's security under the adversarial model, as outlined in Section 1.1. In the subsequent subsections, it is depicted that the proposed scheme can withstand many well-know attacks:

5.3.1. Clogging attack

$\mathcal{A}$ can try to launch a clogging attack by faking the initial request message $M_{sg1} = \langle \overline{ID}_i, \overline{R}_i, a_i, F_i, E_i, TS_i \rangle$. The attacker simulation may initiate by selecting a random variable, current timestamp pair $\{r_i, TS_i\}$ and then by computing $R_i = r_i.G$, $\overline{R}_i = r_i P_j$, $\overline{ID}_i = ID_i \oplus h(R_i \parallel TS_i)$ and $F_i = ID_k \oplus h(\overline{R}_i \| R_i \| TS_i)$. The attacker may also try to construct $a_i = TS_i.d_i + r_i$ and $E_i = h(R_i \parallel \overline{R}_i \| a_i \| TS_i)$. However, to computed valid $a_i = TS_i.d_i + r_i$, attacker needs private key d_i of the user and further the a_i is used in computation of E_i. The attacker may not be able to generate valid pair $\{a_i, E_i\}$. If $\mathcal{A}$ tries to send an old value of a_i or try to construct a_i without private key of the user, it may not pass both authentication checks $a_i.G = TS_i.P_i + R_i$ and $E_i \overset{?}{=} h(R_i \parallel \overline{R}_i \| a_i \| TS_i)$, because, the old value of a_i cannot be reused as it contains current timestamp TS_i and the verification also contains multiplication of public key of the user $P_i = d_i.P$. Therefore, the proposed scheme can identify a clogging attack at first instance and does not allow the replay of the old forged message to pass the authentication checks by FS_j. Hence, after detecting clogging at the first instance, the FS_j may never send the authentication request to the device D_k. Therefore, the proposed scheme resists resource clogging of any type.

5.3.2. Anonymity and untraceability

In the proposed scheme, users identity is never shared openly or sent over the public channel. Still, instead, the identity ID_i is protected by the collision resistance hash function $(h(.))$ and bit-wise operator $(\oplus)$ and masked identity $\overline{ID_i}$ is sent. Also, for each new session, the parameters $\{r_i, r_f, r_k, TS_i, TS_f, TS_k\}$ are freshly picked, which also makes the scheme untraceable.

```
role role_TA(TA,USERS,FOGSERVER,SMARTDEVICE:agent,Skuita:symmetric
_key,H:hash_func,SND,RCV:channel(dy))
played_by TA
def=

 local
  State:nat,
  IDi,Pi,TCi,IDj,Di,TSk,Ri,Dj,Pj,Randi,TSi,Rk,IDk,G,Rf,K,
  IDl,Dl:text,
  F:hash_func
  init State := 0
 transition

  1. State=0
   /\RCV({F(Di'.G').IDi}_Skuita) =|> State':=1
   /\TCi':=H(IDi.K)
   /\ Dl' := H(K.IDl)
   /\ Dj' := H(IDj.Dl')
   /\ Pj' := F(Dj'.G)

   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   /\ secret(K,sec_4,{TA})
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   /\ SND({TCi'.IDk.IDj.Pj'}_Skuita)

end role
```

(a) Role specification for trusted authority (TA)

```
role role_FOGSERVER(FOGSERVER,USERS,TA,SMARTDEVICE:agent,H:hash_func
,SND,RCV:channel(dy))
played_by FOGSERVER
def=
 local State:nat,IDi,BRi,Ai,Fi,BIDi,Pi,TCi,IDj,Di,TSk,Ri,Dj,Pj,Randi
 ,TSi,Rk,IDk,G,Rf,K,TSf,IDl,Dl,Pf,Dk,SIDi,BIDk,Gj,Hj:text,F:
 hash_func init State := 0
 transition
  3. State=0
  /\RCV(xor(IDi,H(F(Randi'.G).TSi')).F(Randi'.F(H(IDj.H(K.
  IDl)).G)).F(F(TSi'.Di').Randi').xor(IDk,F(Randi'.F(H(IDj.H(K.IDl))
  .G).F(Randi'.G).TSi')).TSi') =|> State':=1
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   /\ request(FOGSERVER,USERS,auth_4,TSi')
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   /\Dj':=H(IDj.H(K.IDl)) /\TSf':=new() /\f':=new() /\Pf':=F(Rf'.G)
   /\Dk':=H(Dj'.IDk.IDj) /\SIDi':=xor(IDi,H(Dk'.IDk.TSf'))/\ Pj' :=
   F(Dj'.G) /\BIDk':=xor(IDk,H(Dk'.TSf'))/\BRi':=F(Randi'.Pj')/\ Gj'
   := xor(H(H(Dj'.IDk.IDj).IDk.TSf'),H(F(F(Randi'.Rf').G).H(Ri'.TSi'
   ).IDi))/\ Ri' := F(Dj'.BRi')/\ Hj' := H(IDi.IDk.Gj'.Pf'.TSf'.Dk')
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   /\ witness(FOGSERVER,SMARTDEVICE,auth_5,TSf')
   /\ secret(Rf',sec_2,{FOGSERVER})
   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   /\ SND(SIDi'.BIDk'.Gj'.Hj'.Pf'.TSf')

end role
```

(b) Role specification for fog server (FS)

Fig. 5. Role specification for trusted authority and fog server.

```
role session(SMARTDEVICE,USERS,TA,FOGSERVER:agent,Skuita:symmetric
_key,H:hash_func)
def=
 local SND4,RCV4,SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)

 composition
  role_USERS(USERS,TA,FOGSERVER,SMARTDEVICE,Skuita,H,SND1,RCV1)
  /\role_FOGSERVER(FOGSERVER,USERS,TA,SMARTDEVICE,H,SND3,RCV3)
  /\role_TA(TA,USERS,FOGSERVER,SMARTDEVICE,Skuita,H,SND2,RCV2)
  /\role_SMARTDEVICE(SMARTDEVICE,USERS,TA,FOGSERVER,H,SND4,RCV4)
end role

role environment()

def=
 const
  fogserver,users,smartdevice,ta:agent,
  tsi,tsf,tsk:text,
  skuita:symmetric_key,
  h:hash_func,
  sec_1,sec_2,sec_3,sec_4,auth_4,auth_5,auth_6,
  auth_7:protocol_id
 intruder_knowledge = {h,tsi,tsf,tsk}
 composition
  session(i,users,ta,fogserver,skuita,h)
  /\ session(smartdevice,users,ta,i,skuita,h)
  /\ session(smartdevice,users,i,fogserver,skuita,h)
  /\ session(smartdevice,i,ta,fogserver,skuita,h)
  /\ session(smartdevice,users,ta,fogserver,skuita,h)
end role

goal
 secrecy_of sec_1
 secrecy_of sec_2
 secrecy_of sec_3
 secrecy_of sec_4 authentication_on auth_4 authentication_on auth_5
 authentication_on auth_6 authentication_on auth_7
end goal

environment()
```

(a) Role specification for session, goal and environment

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/vmhlpsl.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 32.94s
  visitedNodes: 1576 nodes
  depth: 8 plies
```

(b) OFMC backend result

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/vmhlpsl.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed    : 5624 states
  Reachable   : 624 states
  Translation: 0.69 seconds
  Computation: 0.20 seconds
```

(c) CL-AtSe backend result

Fig. 6. Role specification for session, goal and environment and result of the analysis.

5.3.3. User impersonation attack

$\mathcal{A}$ may try to impersonate/pose as a legitimate user in order to harm or misuse the system resources. Assume that $\mathcal{A}$ forges a message $M_{sg1} = \langle \overline{ID}_i, \overline{R}_i, a_i, F_i, E_i, TS_i \rangle$ in order to impersonate as a U_i. To do so $\mathcal{A}$ picks an arbitrary number $r_i^{\mathcal{A}}$ and present timestamp $TS_i^{\mathcal{A}}$. But, to forge a message $\mathcal{A}$ requires the knowledge of $\{R_i, ID_i, ID_k, d_i\}$; however, all these values are unknown to the adversary. Therefore, the proposed protocol is secure against the user impersonation attack.

5.3.4. Privileged-insider attack

Assume that $\mathcal{A}$ is a privileged insider and can apprehend the MD_i of U_i after the registration. Now $\mathcal{A}$ can read all the stored information in the MD_i though power analysis [34,35,42]. But to acquire any secret parameter from MD_i, $\mathcal{A}$ requires the knowledge of ID_i, PW_i and σ_i. All these values are unknown to the adversary, and the adversary cannot launch an insider attack.

5.3.5. Man-in-the-middle attack

Let $\mathcal{A}$ has captured the login message containing $M_{sg1} = \langle \overline{ID}_i, \overline{R}_i, a_i, F_i, TS_i \rangle$ and forges its own message. To generate his own message, an adversary can easily use $TS_1^{\mathcal{A}}$ and $r_i^{\mathcal{A}}$ but, in order to compute the remaining values, the adversary needs R_i, d_i and TC_j as all these three values are unknown to the adversary. So, he/she cannot generate its own message. Therefore the proposed protocol is secure against man-in-middle attack.

5.3.6. Replay attack

The present timestamp is incorporated in the parameters through a hash function to prevent a reply attack. In our proposed protocol, the transmission delay $\triangle T$ is significantly small for the adversary to reply to the message. If an adversary tries to replay the message, he cannot pass the check of message freshness. So, the adversary can't launch the replay attack.

5.3.7. Offline parameters guessing attack

Suppose that an $\mathcal{A}$ kens all the sensitive information saved in user's MD_i which includes $\{ID_i^*, d_i^*, TC_i^*, RPB_i, P_i, \{ID_k|k = 1, 2, \ldots, n_d\}, \{ID_j, P_j|j = 1, 2, \ldots, n_f\}, \tau_i, Gen(.), Rep(.), t, h(.)\}$. Now to guess ID_i, d_i and TC_i, $\mathcal{A}$ requires the knowledge of ID_i, PW_i and BIO_i, which are not available to $\mathcal{A}$. Therefore, the proposed scheme can provide resilience against offline parameters guessing attacks.

5.3.8. FS_j impersonation attack

$\mathcal{A}$ may try to impersonate as a fog server and send a forged message to the smart device. It can result in misuse of smart device resources and a decrease in QoS as the smart device will be busy processing requests sent by the adversary. Suppose $\mathcal{A}$ picks an arbitrary nonce $r_i^{\mathcal{A}}, r_f^{\mathcal{A}}$ and present timestamp $TS_i^{\mathcal{A}}, TS_f^{\mathcal{A}}$ to impersonate as a FS_j. To forge a message $M_{sg2} = \langle ID_i^*, \overline{ID}_k, G_j, H_j, P_f, TS_f \rangle$, $\mathcal{A}$ needs additional parameters $\{ID_k, ID_j, ID_i, R_i, d_j, d_k\}$ which are unknown to adversary. Therefore, the proposed protocol is secure against impersonation of FS_j.

5.3.9. Smart device impersonation attack

$\mathcal{A}$ may also try to impersonate as a smart device and send a forged message to the user and lure him into communicating and sharing information. As described in Section 5.3.8 that $\mathcal{A}$ requires specific parameters to impersonate, likewise in order to impersonate as a smart device, $\mathcal{A}$ requires the parameters $\{ID_i, ID_k, TC_k\}$ to forge the message $M_{sg3} = \langle ID_k^*, M_k, N_k, P_f, TS_k \rangle$. Therefore, the adversary cannot launch this attack.

5.3.10. Mobile device stolen attack

As described in Section 5.3.7 that even if the mobile device of the U_i is stolen/misplaced, $\mathcal{A}$ still cannot retrieve any sensitive information from MD_i because, this requires the knowledge of $\{ID_i, PW_i, \sigma_i\}$. Hence the scheme can withstand a mobile device stolen attack.

6. Comparative analysis

In this section the proposed scheme has been compared with existing scheme including: the schemes of Wazid et al. [25], Amin et al. [43], Ma et al. [44], and Chen et al. [45].

6.1. Security requirements

Table 4 depicts the security feature comparison of proposed scheme with existing schemes [25,43–45]. The comparisons explained through in Table 4 show that proposed scheme extends much better security features as compared with SAKA-FC [25] proposed by Wazid et al.

6.2. Communication overhead comparison

The communication cost estimate is presented in Table 5. For comparison, we consider: the identity is 128 bits, a random number is 128 bits, a timestamp is 32 bits, a hash digest is 160 bits (if SHA-1 is employed [46]), cost for ECC point $R = (P_x = 160, P_y = 160)$ is 320 bits, and 128 bits block-size is considered for symmetric enc/dec-ryption , respectively.

Now if we take the aforesaid costs into consideration, the communication cost of the $M_{sg1} = \langle \overline{ID}_i, \overline{R}_i, a_i, F_i, E_i, TS_i \rangle$ is $\langle 160, 320, 160, 160, 160, 32 \rangle = 992$ bits, $M_{sg2} = \langle ID_i^*, \overline{ID}_k, G_j, H_j, P_f, TS_f \rangle$ is $\langle 160, 160, 160, 160, 320, 32 \rangle = 992$ bits, and cost for $M_{sg3} = \langle ID_k^*, M_k, N_k, P_f, TS_k \rangle$ is $\langle 160, 160, 160, 320, 32 \rangle = 832$ bits. Summing all these, the total communication cost of the proposed scheme during the login and authentication phase becomes 2816 bits.

As depicted in Table 5 that the communication cost of the proposed scheme is equal to [25] and less than [25,44,45] except [43]. However, proposed scheme is better in security than [25] as shown in Table 4 and has less computation power than other schemes [43–45] as explained in next subsection. The communication cost comparison is also depicted in Fig. 7.

Table 4
Comparison of functionality features.

	[25]	[43]	[44]	[45]	Our
Fx_1	×	✓	✓	✓	✓
Fx_2	×	✓	✓	✓	✓
Fx_3	✓	✓	✓	✓	✓
Fx_4	✓	✓	✓	✓	✓
Fx_5	✓	✓	✓	✓	✓
Fx_6	✓	✓	✓	✓	✓
Fx_7	✓	✓	✓	✓	✓
Fx_8	✓	✓	✓	✓	✓
Fx_9	✓	×	✓	✓	✓
Fx_{10}	✓	✓	✓	✓	✓

Note: Fx_1: Clogging Attack; Fx_2: User anonymity/untraceability; Fx_3: Resistance against user impersonation attack; Fx_4: Resistance against insider attack; Fx_5: Resistance against MITM Attack; Fx_6: Resistance against replay attack; Fx_7: Protection against off-line parameters guessing attack; Fx_8: Resistance against FS_j/Server impersonation; Fx_9: Suitable for multi-server environment; Fx_{10}: Secure against stolen smart-card attack; where Fx_n is the nth compared feature. Feature Exists: ✓; Feature does not Exist: ×.

Table 5
Communication cost comparison.

Schemes	# of messages	# of bits
Wazid et al. [25]	3	2816
Amin et al. [43]	4	2144
Ma et al. [44]	4	4800
Chen et al. [45]	4	4768
Proposed scheme	3	2816

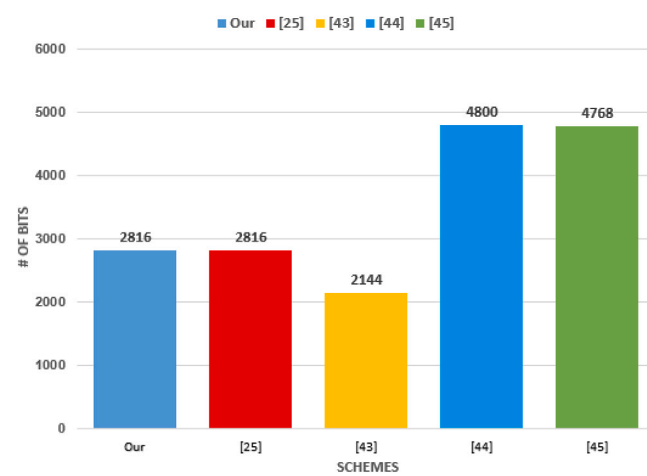

Fig. 7. Communication cost comparison.

6.3. Computation overhead comparison

In this section, the computation of various schemes has been compared. As discussed in [25], computation time required for the hash, ECC point addition and multiplication, symmetric encryption/ decryption, asymmetric encryption/ decryption, identity based encryption/decryption, identity based signature/ verification, modular multiplication and for fuzzy extractor is 0.5, 63.075 and 10.875, 8.7, 870, 60.75, 60.75, 522 and 63.075, respectively in ms. it is noted that $T_{fe} \approx T_{ecm}$, $T_m \approx 60T_{sym}$ and $T_{asym} \approx 100T_{sym}$. The approximate time needed for each cryptographic operation and the related notation are also illustrated in Table 6.

As depicted in Table 7 that the computation cost of the proposed scheme is a bit high as compared to [25]. While the proposed scheme provides resistance to clogging and related attacks and scheme proposed in [25] lacks untraceablity and is vulnerable to clogging attack as proved in Section 3.2. Moreover, proposed scheme performs better than other competing schemes [43–45], as shown in Table 4 as well as in Fig. 8.

Table 6
Approximate time required for various operations.

Notation	Description	≈ computation time in ms
T_h	Hash function	0.5
T_{ecm}	ECC point multiplication	63.075
T_{eca}	ECC point point addition	10.875
T_{fe}	Fuzzy extractor function	63.075

Table 7
Computation cost comparison.

Schemes	U/MD	FS/FN	$CS/MS/SP$	Smart device	≈ Total
[25]	$1T_{fe}+16T_h$	$10T_h+3T_{ecm}$	–	$10T_h$	≈ 407.325
	$+2T_{ecm}$	$+1T_{eca}$	–		
[43]	$8T_h+2T_{ecm}$	–	$2T_{ecm}+4T_h$	$9T_h+2T_{ecm}$	≈ 1454.7
	$+1T_{eca}+1T_m$			$+1T_{eca}+1T_m$	
[44]	–	$4T_h+4T_{ecm}$	$9T_h+8T_{ecm}$	–	≈ 954.125
[45]	$1T_{fe}+6T_h$	$12T_h+3T_{ecm}$	$4T_h+4T_{ecm}$	–	≈ 641.75
	$+2T_{ecm}$				
Proposed	$1T_{fe}+10T_h$	$4T_{ecm}+8T_h$	–	$8T_h$	≈ 528.475
	$+3T_{ecm}$	$+1T_{eca}$	–		

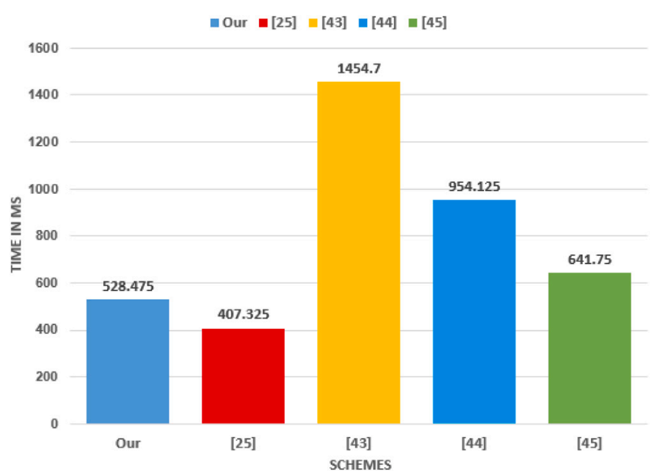

Fig. 8. Computation cost comparison.

7. Conclusion

The need for low latency communication increases as more time-critical systems are developed with each passing day, so is the importance of edge/fog computing. Edge/fog computing is becoming the focal point of recent research due to the increase in its adoption. Edge/fog computing is the extension of cloud computing, and due to this, it borrows the strengths and weaknesses of it. One of the main concerns about fog computing is security. Authentication schemes are put in place to ensure that only legal users can access the resources and stop the ill-willed users from accessing system resources. To overcome this issue, many researchers have proposed authentication schemes. In this paper, we examined a recently proposed key management and user authentication scheme for fog computing SAKA-FC by Wazid et al. After careful analysis, we identified that it is insecure against traceability and user impersonation attack and is also inefficient. To subdue the problems mentioned above, we presented an enhanced scheme. The security of the proposed scheme is proved through formal, informal, and automated methods. The proposed scheme provides all the security features and resistance against many known attacks with equal communication overhead. There is a minor increase in computation time as of Wazid et al.'s SAKA-FC. However, due to robustness and the same communication cost, the proposed scheme is best suitable for securing the communication between users and smart devices in fog computing-based architectures.

CRediT authorship contribution statement

Zeeshan Ali: Writing - original draft, Methodology, Software. **Shehzad Ashraf Chaudhry:** Conceptualization, Methodology, Validation. **Khalid Mahmood:** Writing - review & editing, Validation, Visualization. **Sahil Garg:** Visualization, Investigation, Validation. **Zhihan Lv:** Validation, Formal analysis, Visualization. **Yousaf Bin Zikria:** Supervision, Methodology, Validation, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors extend their appreciation to the International Islamic University, Islamabad, Pakistan, Istanbul Gelisim University, Istanbul, Turkey. The research is supported in part by the National Natural Science Foundation of China (No. 61902203) and Key Research and Development Plan - Major Scientific and Technological Innovation Projects of ShanDong Province, China (2019JZZY020101).

References

[1] K.E.S. Desikan, V.J. Kotagi, C.S.R. Murthy, Topology control in fog computing enabled IoT networks for smart cities, Comput. Netw. (2020) 107270, http://dx.doi.org/10.1016/j.comnet.2020.107270.
[2] J. Bellendorf, Z.Á. Mann, Classification of optimization problems in fog computing, Future Gener. Comput. Syst. 107 (2020) 158–176, http://dx.doi.org/10.1016/j.future.2020.01.036.
[3] S. Garg, K. Kaur, G. Kaddoum, S.H. Ahmed, D.N.K. Jayakody, SDN-Based secure and privacy-preserving scheme for vehicular networks: A 5G perspective, IEEE Trans. Veh. Technol. 68 (9) (2019) 8421–8434.
[4] M. Iorga, L. Feldman, R. Barton, M.J. Martin, N. Goren, C. Mahmoudi, Fog Computing Conceptual Model, Technical Report, National Institute of Standards and Technology, 2018, http://dx.doi.org/10.6028/nist.sp.500-325.
[5] S. Garg, A. Singh, K. Kaur, G.S. Aujla, S. Batra, N. Kumar, M.S. Obaidat, Edge computing-based security framework for big data analytics in VANETs, IEEE Netw. 33 (2) (2019) 72–81.
[6] N. Abbas, M. Asim, N. Tariq, T. Baker, S. Abbas, A mechanism for securing IoT-enabled applications at the fog layer, J. Sensor Actuator Netw. 8 (1) (2019) 16, http://dx.doi.org/10.3390/jsan8010016.
[7] M. Al-khafajiy, T. Baker, H. Al-Libawy, Z. Maamar, M. Aloqaily, Y. Jararweh, Improving fog computing performance via fog-2-fog collaboration, Future Gener. Comput. Syst. 100 (2019) 266–280, http://dx.doi.org/10.1016/j.future.2019.05.015.
[8] I. Azimi, A. Anzanpour, A.M. Rahmani, T. Pahikkala, M. Levorato, P. Liljeberg, N. Dutt, HiCH: Hierarchical fog-assisted computing architecture for healthcare IoT, ACM Trans. Embedded Comput. Syst. 16 (5s) (2017) 1–20, http://dx.doi.org/10.1145/3126501.
[9] S.A. Chaudhry, I.L. Kim, S. Rho, M.S. Farash, T. Shon, An improved anonymous authentication scheme for distributed mobile cloud computing services, Cluster Comput. 22 (1) (2019) 1595–1609.
[10] K. Kaur, S. Garg, G.S. Aujla, N. Kumar, J.J.P.C. Rodrigues, M. Guizani, Edge computing in the industrial internet of things environment: Software-defined-networks-based edge-cloud interplay, IEEE Commun. Mag. 56 (2) (2018) 44–51.
[11] T. Baker, M. Asim, Á. MacDermott, F. Iqbal, F. Kamoun, B. Shah, O. Alfandi, M. Hammoudeh, A secure fog-based platform for SCADA-based IoT critical infrastructure, Softw. - Pract. Exp. 50 (5) (2020) 503–518, http://dx.doi.org/10.1002/spe.2688.
[12] M. Chiang, T. Zhang, Fog and IoT: An overview of research opportunities, IEEE Internet Things J. 3 (6) (2016) 854–864, http://dx.doi.org/10.1109/jiot.2016.2584538.
[13] H. Zahmatkesh, F. Al-Turjman, Fog computing for sustainable smart cities in the IoT era: Caching techniques and enabling technologies - an overview, Sustainable Cities Soc. 59 (2020) 102139, http://dx.doi.org/10.1016/j.scs.2020.102139.
[14] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A.Y. Zomaya, R. Ranjan, A hybrid deep learning-based model for anomaly detection in cloud datacenter networks, IEEE Trans. Netw. Serv. Manag. 16 (3) (2019) 924–935.

[15] G. Caiza, M. Saeteros, W. Oñate, M.V. Garcia, Fog computing at industrial level, architecture, latency, energy, and security: A review, Heliyon 6 (4) (2020) e03706, http://dx.doi.org/10.1016/j.heliyon.2020.e03706.

[16] M. Qiu, S.-Y. Kung, K. Gai, Intelligent security and optimization in edge/fog computing, Future Gener. Comput. Syst. 107 (2020) 1140–1142, http://dx.doi.org/10.1016/j.future.2019.06.002.

[17] K. Lee, D. Kim, D. Ha, U. Rajput, H. Oh, On security and privacy issues of fog computing supported internet of things environment, in: 2015 6th International Conference on the Network of the Future (NOF), IEEE, 2015, pp. 1–3, http://dx.doi.org/10.1109/nof.2015.7333287.

[18] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, K. Ren, A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing, IEEE Trans. Inf. For. Secur. 11 (11) (2016) 2594–2608, http://dx.doi.org/10.1109/tifs.2016.2590944.

[19] P. Hu, H. Ning, T. Qiu, H. Song, Y. Wang, X. Yao, Security and privacy preservation scheme of face identification and resolution framework using fog computing in internet of things, IEEE Internet Things J. 4 (5) (2017) 1143–1155, http://dx.doi.org/10.1109/jiot.2017.2659783.

[20] P. Gope, LAAP: Lightweight anonymous authentication protocol for D2D-aided fog computing paradigm, Comput. Secur. 86 (2019) 223–237, http://dx.doi.org/10.1016/j.cose.2019.06.003.

[21] E. Alemneh, S.-M. Senouci, P. Brunet, T. Tegegne, A two-way trust management system for fog computing, Future Gener. Comput. Syst. 106 (2020) 206–220, http://dx.doi.org/10.1016/j.future.2019.12.045.

[22] A. Alrawais, A. Alhothaily, C. Hu, X. Xing, X. Cheng, An attribute-based encryption scheme to secure fog communications, IEEE Access 5 (2017) 9131–9138, http://dx.doi.org/10.1109/access.2017.2705076.

[23] A.A. Khan, V. Kumar, M. Ahmad, S. Rana, D. Mishra, PALK: Password-based anonymous lightweight key agreement framework for smart grid, Int. J. Electr. Power Energy Syst. 121 (2020) 106121.

[24] S.A. Chaudhry, Correcting "PALK: Password-based anonymous lightweight key agreement framework for smart grid", Int. J. Electr. Power Energy Syst. 125 (2021) 106529, http://dx.doi.org/10.1016/j.ijepes.2020.106529.

[25] M. Wazid, A.K. Das, N. Kumar, A.V. Vasilakos, Design of secure key management and user authentication scheme for fog computing services, Future Gener. Comput. Syst. 91 (2019) 475–492, http://dx.doi.org/10.1016/j.future.2018.09.017.

[26] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.

[27] S.A. Chaudhry, T. Shon, F. Al-Turjman, M.H. Alsharif, Correcting design flaws: An improved and cloud assisted key agreement scheme in cyber physical systems, Comput. Commun. 153 (2020) 527–537.

[28] A. Ghani, K. Mansoor, S. Mehmood, S.A. Chaudhry, A.U. Rahman, M. Najmus Saqib, Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key, Int. J. Commun. Syst. 32 (16) (2019) e4139.

[29] S.A. Chaudhry, H. Alhakami, A. Baz, F. Al-Turjman, Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure, IEEE Access 8 (2020) 101235–101243.

[30] Z. Ali, A. Ghani, I. Khan, S.A. Chaudhry, S.H. Islam, D. Giri, A robust authentication and access control protocol for securing wireless healthcare sensor networks, J. Inf. Secur. Appl. 52 (2020) 102502, http://dx.doi.org/10.1016/j.jisa.2020.102502.

[31] Z. Ali, S.A. Chaudhry, M.S. Ramzan, F. Al-Turjman, Securing smart city surveillance: A lightweight authentication mechanism for unmanned vehicles, IEEE Access 8 (2020) 43711–43724, http://dx.doi.org/10.1109/access.2020.2977817.

[32] R. Canetti, H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, in: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2001, pp. 453–474.

[33] B. LaMacchia, K. Lauter, A. Mityagin, Stronger security of authenticated key exchange, in: International Conference on Provable Security, Springer, 2007, pp. 1–16.

[34] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, M.T.M. Shalmani, On the power of power analysis in the real world: A complete break of the keeloq code hopping scheme, in: D. Wagner (Ed.), Advances in Cryptology – CRYPTO 2008, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 203–220, http://dx.doi.org/10.1007/978-3-540-85174-5_12.

[35] T. Messerges, E. Dabbish, R. Sloan, Examining smart-card security under the threat of power analysis attacks, IEEE Trans. Comput. 51 (5) (2002) 541–552, http://dx.doi.org/10.1109/tc.2002.1004593.

[36] Y. Zhang, M. Simsek, B. Kantarci, Self organizing feature map for fake task attack modelling in mobile crowdsensing, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6, http://dx.doi.org/10.1109/globecom38437.2019.9014197.

[37] K. Park, H. Lee, On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets, ACM SIGCOMM Comput. Commun. Rev. 31 (4) (2001) 15–26, http://dx.doi.org/10.1145/964723.383061.

[38] R. Oppliger, Protecting key exchange and management protocols against resource clogging attacks, in: Secure Information Networks, Springer US, 1999, pp. 163–175, http://dx.doi.org/10.1007/978-0-387-35568-9_11.

[39] S. Roy, C. Khatwani, Cryptanalysis and improvement of ECC based authentication and key exchanging protocols, Cryptography 1 (1) (2017) 9, http://dx.doi.org/10.3390/cryptography1010009.

[40] M. Burrows, M. Abadi, R.M. Needham, A logic of authentication, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 426 (1871) (1989) 233–271.

[41] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P.H. Drielsma, P.C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, L. Vigneron, The AVISPA tool for the automated validation of internet security protocols and applications, in: Computer Aided Verification, Springer Berlin Heidelberg, 2005, pp. 281–285, http://dx.doi.org/10.1007/11513988_27.

[42] P. Kocher, J. Jaffe, B. Jun, Differential power analysis, in: Advances in Cryptology — CRYPTO' 99, Springer Berlin Heidelberg, 1999, pp. 388–397, http://dx.doi.org/10.1007/3-540-48405-1_25.

[43] R. Amin, S. Kunal, A. Saha, D. Das, A. Alamri, CFSec: Password based secure communication protocol in cloud-fog environment, J. Parallel Distrib. Comput. 140 (2020) 52–62, http://dx.doi.org/10.1016/j.jpdc.2020.02.005.

[44] M. Ma, D. He, H. Wang, N. Kumar, K.-K.R. Choo, An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks, IEEE Internet Things J. 6 (5) (2019) 8065–8075, http://dx.doi.org/10.1109/jiot.2019.2902840.

[45] C.-M. Chen, Y. Huang, K.-H. Wang, S. Kumari, M.-E. Wu, A secure authenticated and key exchange scheme for fog computing, Enterp. Inf. Syst. (2020) 1–16, http://dx.doi.org/10.1080/17517575.2020.1712746.

[46] r.D. Eastlake, P. Jones, US secure hash algorithm 1 (SHA1), RFC 3174 (2001) 1–22, URL: http://www.rfc-editor.org/info/rfc3174.

Zeeshan Ali received the bachelor's degree from NUML University Islamabad and MS(CS) degree in information security from International Islamic University Islamabad in 2020. He has published four articles in conferences and journals and has submitted some of the articles in top journals. His research interests include blockchain, computer networking, network security, network communication, information security, cryptography, encryption, and authentication.

Shehzad Ashraf Chaudhry received the master's and Ph.D. degrees (with Distinction) from International Islamic University Islamabad, Pakistan, in 2009 and 2016, respectively. He is currently working as an Associate Professor with the Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey before this he has served International Islamic University and University of Sialkot, Pakistan. Dr. Shehzad completed his Master and PhD in computer sciences with Distinction in 2009 and 2016, respectively. He has authored over 100 scientific publications appeared in different international journals and proceedings, including more than 79 in web of science journals. Some of his findings are published in top rated prestigious journals including ACM Transaction on Internet Technology, IEEE: Internet of Things Journal, Transaction on Reliability, Transaction on Industry Application, Systems Journal, Access, Elsevier: Future Generation computer Systems, International Journal of Electrical Power & Energy Systems, Computer Networks, Computer Communications etc. With an H-index of 25 and an I- 10 index 52, his work has been cited over 1990 times. He has also supervised over 35 graduate students in their research. His current research interests include lightweight cryptography, elliptic/hyper elliptic curve cryptography, multimedia security, E- payment systems, MANETs, SIP authentication, smart grid security, IP multimedia subsystem, and next generation networks. He occasionally writes on issues of higher education in Pakistan.

Dr. Chaudhry was a recipient of the Gold Medal for achieving 4.0/4.0 CGPA in his Masters. Considering his research, Pakistan Council for Science and Technology granted him the Prestigious Research Productivity Award, while affirming him among Top Productive Computer Scientist in Pakistan. Recently, he is listed among Top 2% Computer Scientists across the world in Stanford University's report. He is also serving as guest editor for many WoS indexed journals and have served/serving as a TPC member of various international conferences. He is also an active reviewer of many WoS indexed journals.

Khalid Mahmood received the M.S. degree in computer science from Riphah International University, Islamabad, Pakistan, in 2010, and the Ph.D. degree in computer science from International Islamic University, Islamabad, in 2018. The title of his Ph.D. dissertation is Secure Authenticated Key Agreement Schemes for Smart Grid Communication in Power Sector. He is currently working with COMSATS University Islamabad, Sahiwal Campus. His research interests include lightweight cryptography, smart grid authentication, authenticated key agreement schemes, and design and development of lightweight authentication protocols using lightweight cryptographic solutions for diverse infrastructures or systems, such as vehicular ad hoc networks, smart grid, and telecare medical information systems.

Sahil Garg (Member, IEEE) received the Ph.D. degree from the Thapar Institute of Engineering and Technology, Patiala, India, in 2018. He is currently a Postdoctoral Research Fellow with the École de technologie supérieure, Université du Québec, Montréal, QC, Canada. He has many research contributions in the area of machine learning, big data analytics, security and privacy, the Internet of Things, and cloud computing. He has over 60 publications in high ranked journals and conferences, including more than 40 top-tier journal articles and more than 20 reputed conference papers. He is a member of ACM. He was awarded the IEEE ICC Best Paper Award, in 2018, at Kansas City, Missouri. He is currently a Managing Editor of Springer's Human-centric Computing and Information Sciences (HCIS) journal. He is also an Associate Editor of the IEEE Network Magazine, the IEEE System Journal, Applied Soft Computing (Elsevier), Future Generation Computer Systems (FGCS) (Elsevier), and International Journal of Communication Systems (IJCS) (Wiley). He also serves as the Workshops and Symposia Officer for the IEEE ComSoc Emerging Technology Initiative on Aerial Communications. He guest-edited a number of special issues in top-cited journals, including the IEEE Transactions on Intelligent Transportation Systems, the IEEE Transactions on Industrial Informatics, the IEEE Internet on Things Journal, the IEEE Network, and Future Generation Computer Systems (Elsevier). He serves/served as the workshop chair/publicity co-chair for several IEEE/ACM conferences, including the IEEE Infocom, the IEEE Globecom, the IEEE ICC, ACM MobiCom, and so on.

Zhihan Lv (Senior Member, IEEE) received the Ph.D. degree in computer applied technology from Ocean University of China, Qingdao, China, in 2012. He is currently an Associate Professor with Qingdao University, Qingdao, China. He has completed several projects successfully on PC, Website, smartphone, and smartglasses. His research interests include big data analytics, multimedia, augmented reality, virtual reality, computer vision, 3-D visualization and graphics, serious game, HCI, bigdata, and GIS.

Yousaf Bin Zikria (SM'17) is currently working as an Assistant Professor in the Department of Information and Communication Engineering, Yeungnam University, South Korea. He authored more than 60 articles, conferences, book chapters, and patents. He published papers at the top venue that includes IEEE Communications, Surveys, and Technologies, IEEE Wireless Communications Magazine, Elsevier Future Generation Computer Systems, Elsevier Sustainable Cities and Society, etc. He has managed 12 FT/SI in SCI/E indexed journals. His research interests include IoT, 5G, Machine Learning, wireless communications and networks, WSNs, routing protocols, CRAHN, CRASN, transport protocols, VANETS, embedded system and, network and information security. He also held the prestigious CISA, JNCIS-SEC, JNCIS-ER, JNCIA-ER, JNCIA-EX, and Advance Routing Switching and WAN Technologies certifications.

Google Scholar: https://scholar.google.com/citations?user=K90qMyMAAAAJ&hl=en

Website: https://sites.google.com/view/ybzikria

Researchgate: https://www.researchgate.net/profile/Yousaf_Zikria.