Wuhan University
Journal of Natural Sciences

2024, Vol.29 No.3, 209-218

Article ID 1007-1202(2024)03-0209-10 DOI https://doi.org/10.1051/wujns/2024293209

Cite this article: HUO Yuyan, KANG Baoyuan, NIU Shufang, *et al*. Analysis and Improvement of an Authentication Scheme for Fog Computing Services[J]. *Wuhan Univ J of Nat Sci*, 2024, 29(3): 209-218.

Analysis and Improvement of an Authentication Scheme for Fog Computing Services

□ **HUO Yuyan[1], KANG Baoyuan[1†], NIU Shufang[2], LI Anqian[1], ZUO Xinyu[1]**

1. School of Software, Tiangong University, Tianjin 300387, China;
2. School of Computer Science and Technology, Tiangong University, Tianjin 300387, China

Abstract: Fog computing utilizes devices in the edge network to transmit data with very low latency and supports high mobility. However, fog computing inherits security and privacy problems from cloud computing. Therefore, various privacy schemes for fog computing have been proposed to prevent different types of attacks. Recently, Weng *et al* proposed a fog computing authentication scheme; after analyzing, we found that Weng *et al*'s scheme cannot resist user tracking attack and user impersonation attack. Then, we propose an improved scheme through adding a password, modifying the calculation method of E_i, and adding timestamps. In addition, we also compare the improved scheme with existing authentication schemes in terms of security and computational efficiency. The results show that the improved scheme is more secure and has less computation.

Key words: authentication scheme; fog computing; security

CLC number: TP309.2

0 Introduction

Cloud computing is a business computing model. It distributes computer tasks to resource pools made up of large numbers of computers, enabling various applications to obtain computing power, storage space, and information services as needed. However, compared with traditional decentralized computing, cloud computing centralizes computing resources, and risks are centralized together. Therefore, cloud computing cannot meet the needs of high mobility, location-aware, and low-latency applications [1]. Fog computing was born to eliminate the limitations of cloud computing and is used to be a link between the Internet of Things devices and the cloud [1]. In fog computing, fog services are distributed at the margin of the network and close to terminal equipment geographically [2], so some data can be processed and stored directly in the fog layer. As a result, fog computing can reduce the pressure on the cloud, improve transmission rates, and reduce latency.

Fog computing can effectively decentralize computational and analytical power and help reduce bandwidth usage. However, fog nodes are often deployed in remote and unprotected places [3], and rely on insecure public channels for data transmission between users and fog servers, as well as fog servers and cloud servers. There-

Received date: 2023-09-28

Biography: HUO Yuyan, female, Master candidate, research directions: cryptography. E-mail: yuyanhuo@aliyun.com

† Corresponding author. E-mail: baoyuankang@aliyun.com

fore, secure identity authentication is critical in fog computing.

Many researchers have recently proposed identity authentication schemes in fog computing environments. Lampot first proposed a remote authentication scheme in an insecure environment in 1981[4]. Then, many two-factor authentication schemes based on Hashes, smart cards, and temporary certificates were proposed[5-10], but most of them have security issues. Then, lots of three-factor authentication schemes[1,11-15] based on fog computing were proposed. In 2019, Ma *et al*[16] proposed an authenticated key agreement protocol without bilinear pairing and claimed that their scheme achieves mutual authentication, generates a securely agreed session key for secret communication, and supports privacy protection. In 2021, Chen *et al*[17] proposed an authenticated key exchange scheme for fog computing. However, after analysis, Rana *et al* [18] found that Chen *et al*'s scheme[17] does not provide user anonymity and is also not resistant to tamper-proof device stolen attack, user impersonation attack, fog node impersonation attack, insider attack, and known session key attack. In 2021, Weng *et al*[15] proposed a lightweight anonymous mutual authentication and secure communication scheme and claimed that their scheme only uses one-way Hash functions, and XOR operations and security can be ensured.

In this paper, we point out the shortcomings of Weng *et al*'s[15] scheme. Weng *et al*'s scheme cannot resist user traceability attack and user impersonation attack. Therefore, we propose an improved scheme through adding a password, modifying the calculation method of E_i, and adding timestamps. We also compare security features and computation costs between the improved scheme and the other four schemes[3, 15-17].

The rest of the paper is structured as follows: Section 1 briefly reviews Weng *et al*'s scheme. Section 2 analyses the shortcomings of Weng *et al*'s scheme. In Section 3, an improved scheme is presented. Section 4 provides a security analysis and comparison of the enhanced scheme. Section 5 concludes the paper.

1 Review of Weng *et al*'s Scheme

In Weng *et al*'s scheme[15], there are two mutual authentication and key agreement phases: one is the authentication and key agreement phase of edge user EU_i and fog server FS_j, and the other is the authentication and key agreement phase of edge device ED_k and fog server FS_j. Through analysis, we find that the two phases similarly implement mutual authentication and key agreement. So, in this section, we only review the authentication and key agreement phase of edge user EU_i and fog server FS_j. There are three participants in Weng *et al*'s scheme: cloud server, fog servers, and edge users. Fog servers and edge users all register to the cloud server, and then edge users and fog servers authenticate with each other and agree on the same session keys with the help of the cloud server. Weng *et al*'s scheme[15] consists of the following steps. The notations used in this article are listed in Table 1.

Table 1 Notations table

Symbol	Description
CS	The cloud server
FS_j	The j-th fog server
EU_i, MD_i	i-th edge user and his/her smart device
ID_i, ID_j	The identity of EU_i, FS_j, respectively
PW_i, BIO_i	i-th edge user's password and biometric
$\mathrm{Gen}(\cdot)$, $\mathrm{Rep}(\cdot)$	Fuzzy generation and reproduction function
σ_i, τ_i	Biometric secret key and Public reproduction parameter
TID_i, TID_j	The pseudonym of EU_i and FS_j, respectively
K_{cf}, K_{cu}	The long-term secret keys chosen by CS
MK	The master secret key of CS
n_u, n_f	160-bit random secret number of EU_i
r_c, r_f, r_u	128-bit random number of CS, FS_j and EU_i, respectively

1.1 System Initialization

The cloud server CS generates a master secret key MK, and two long-term secret keys K_{cf} and K_{cu}, and keeps them secret. CS chooses a one-way Hash function $h(\cdot)$.

1.2 Fog Server Registration Phase

Fog server FS_j selects a unique identity ID_j and sends identity ID_j to CS via a secure channel. On receiving ID_j, CS generates a pseudonym TID_j, and computes

$$B_j = h(\mathrm{ID}_j \| K_{cf}),\ h(\mathrm{MK} \| K_{cf}).$$

Then CS sends $\{\mathrm{TID}_j, B_j, h(\mathrm{MK}\|K_{cf})\}$ to FS_j and maintains $\{\mathrm{TID}_j, \mathrm{ID}_j, B_j\}$ in a protected verifier table of FS_j. Finally, FS_j stores TID_j, B_j, and $h(\mathrm{MK}\|K_{cf})$ in its memory.

1.3 Edge User Registration Phase

Edge user EU_i selects a unique identity ID_i and imprints his/her biometric BIO_i into smart device MD_i. MD_i generates a 160-bit random secret number n_u and computes $A_i = h(\mathrm{ID}_i \| \mathrm{BIO}_i \| n_u)$. Then MD_i sends $\{A_i, \mathrm{ID}_i\}$ to CS through a secure channel. After receiving ID_i and A_i, CS generates a pseudonym TID_i, and computes

$$B_i = h(\mathrm{ID}_i \| \mathrm{MK}),$$

$$C_i = h(\mathrm{ID}_i \| A_i \| h(K_{cu})),$$

$$D_i = B_i \oplus h(\mathrm{MK} \| K_{cu}) \oplus A_i.$$

Then CS sends $\{\mathrm{TID}_i, C_i, D_i, h(\cdot), h(K_{cu})\}$ to EU_i via a secure channel and maintains $\{\mathrm{TID}_i, B_i\}$ in a protected verifier table of EU_i. Finally, EU_i's smart device stores $\{\mathrm{TID}_i, C_i, D_i, h(\cdot), h(K_{cu}), n_u\}$ in its memory.

1.4 Authentication and Key Agreement Phase

In this phase, an edge user EU_i wants to access a fog server FS_j through a public channel, the cloud server CS can help EU_i and FS_j to authenticate each other and achieve a session key SK_{ij}. The specific steps are as follows.

Step 1: EU_i first inputs ID_i' and BIO_i' into his/her smart device. Then, MD_i retrieves n_u and $h(K_{cu})$ to compute

$$A_i' = h(\mathrm{ID}_i' \| \mathrm{BIO}_i' \| n_u),\ C_i' = h(\mathrm{ID}_i' \| A_i' \| h(K_{cu}))$$

and checks whether $C_i' = C_i$. If it is true, it means EU_i is a legal user, then smart device MD_i selects a 128-bit random number r_u and computes

$$E_i = D_i \oplus A_i \oplus r_u,\ F_i = h(h(K_{cu}) \| \mathrm{ID}_j \| r_u).$$

Finally, MD_i sends the message $M_{u1} = \{\mathrm{TID}_i, E_i, F_i\}$ to FS_j through a public channel.

Step 2: On receiving M_{u1}, FS_j selects a 128-bit random number r_f and retrieves TID_j, B_j, and $h(\mathrm{MK}\|K_{cf})$ to compute

$$O_j = B_j \oplus r_f,\ P_j = h(h(\mathrm{MK}\|K_{cf}) \| \mathrm{TID}_j \| r_f).$$

Finally, FS_j sends messages M_{u1} and $M_{u2} = \{\mathrm{TID}_j, O_j, P_j\}$ to CS through a public channel.

Step 3: On receiving M_{u1} and M_{u2}, CS inspects M_{u1} and searches the verifier table of EU_i in its database to find entry that match TID_i. If there is no matching entry, CS rejects the request and terminates the session. Otherwise, CS retrieves B_i and $h(\mathrm{MK}\|K_{cu})$ to compute

$$r_u' = E_i \oplus B_i \oplus h(\mathrm{MK}\|K_{cu}),\ F_i' = h(h(K_{cu}) \| \mathrm{ID}_j \| r_u').$$

Then CS checks whether $F_i' = F_i$. If it is not true, CS terminates the session. Otherwise, the legitimacy of EU_i is authenticated by CS.

Step 4: CS further inspects M_{u2} and searches the verifier table of FS_j in its database to find entry that match TID_j. If there is no matching entry, CS rejects the request and terminates the session. Otherwise, CS retrieves B_j and $h(\mathrm{MK}\|K_{cf})$ to compute

$$r_f' = O_j \oplus B_j,\ P_j' = h(h(\mathrm{MK}\|K_{cf}) \| \mathrm{TID}_j \| r_f').$$

Then CS checks whether $P_j' = P_j$.

Step 5: After verifying the validity of EU_i and FS_j, CS refreshes pseudonyms for EU_i and FS_j by computing

$$\mathrm{TID}_i^{\mathrm{new}} = \mathrm{TID}_i \oplus r_u',\ \mathrm{TID}_j^{\mathrm{new}} = \mathrm{TID}_j \oplus r_f'.$$

And CS replaces TID_i with $\mathrm{TID}_i^{\mathrm{new}}$ in the verifier table of EU_i, replaces TID_j with $\mathrm{TID}_j^{\mathrm{new}}$ in the verifier table of FS_j. CS further selects a 128-bit random number r_c and computes

$$Q_j = r_u' \oplus r_c \oplus h(\mathrm{ID}_j \| \mathrm{TID}_j^{\mathrm{new}}),\ \mathrm{SK}_{ij} = h(r_u' \oplus r_f' \oplus r_c),$$

$$R_j = h(B_j \oplus h(\mathrm{TID}_j^{\mathrm{new}}) \oplus \mathrm{SK}_{ij}),\ S_j = r_f' \oplus r_c \oplus h(\mathrm{ID}_j \| \mathrm{TID}_i^{\mathrm{new}}),$$

$$T_j = h(B_i \oplus h(\mathrm{MK}\|K_{cu}) \oplus h(\mathrm{TID}_i^{\mathrm{new}}) \oplus \mathrm{SK}_{ij}).$$

Finally, CS sends $M_{u3} = \{Q_j, R_j\}$ and $M_{u4} = \{S_j, T_j\}$ to FS_j.

Step 6: On receiving M_{u3} and M_{u4}, FS_j computes

$$\mathrm{TID}_j^{\mathrm{new}} = \mathrm{TID}_j \oplus r_f,\ r_u' \oplus r_c' = Q_j \oplus h(\mathrm{ID}_j \| \mathrm{TID}_j^{\mathrm{new}}),$$

$$\mathrm{SK}_{ij} = h(r_u' \oplus r_c' \oplus r_f),\ R_j' = h(\mathrm{B}_j \oplus h(\mathrm{TID}_j^{\mathrm{new}}) \oplus \mathrm{SK}_{ij}).$$

FS_j checks if $R_j' = R_j$. Finally, FS_j sends M_{u4} to EU_i.

Step 7: On receiving M_{u4}, EU_i computes

$$\mathrm{TID}_i^{\mathrm{new}} = \mathrm{TID}_i \oplus r_u,\ r_f' \oplus r_c' = S_j \oplus h(\mathrm{ID}_j \| \mathrm{TID}_i^{\mathrm{new}}),$$

$$\mathrm{SK}_{ij} = h(r_f' \oplus r_c' \oplus r_u),$$

$$T_j' = h(B_i \oplus h(\mathrm{MK}\|K_{cu}) \oplus h(\mathrm{TID}_i^{\mathrm{new}}) \oplus \mathrm{SK}_{ij}).$$

If $T_j' = T_j$, EU_i believes that CS and FS_j are legal parties and stores the shared session key SK_{ij} for future secure communication.

2 Attacks on Weng *et al*'s Scheme

This section will show that Weng *et al*'s scheme [15] is vulnerable to user traceability and impersonation attacks. Further details are provided in the following subsections.

2.1 User Traceability Attack

Weng *et al*'s scheme [15] cannot resist user traceability attack. The following steps show the process of user traceability attack.

Step 1: In the first authentication, if a user EU_i sends a message M_{U1}={TID_i, E_i, F_i} trying to contact a fog server FS_j, one attacker may intercept the message M_{U1}, and then he/she saves {E_i, TID_i}. At the end of this authentication, the cloud server CS and the user will update the pseudonym by (1). Lastly, the first authentication ends.

$$\mathrm{TID}_i^{\mathrm{new}}=\mathrm{TID}_i\oplus r_u \tag{1}$$

$$E_i=D_i\oplus A_i\oplus r_u \tag{2}$$

But through (1) and (2), the attacker can know

$$\mathrm{TID}_i\oplus E_i\oplus\mathrm{TID}_i^{\mathrm{new}}=D_i\oplus A_i \tag{3}$$

Step 2: In the second authentication, assume that three users EU_A, EU_B, EU_C, and the same user EU_i all send messages trying to contact a fog server FS_g. EU_A sends message M_{U1a}={$\mathrm{TID}_a^{\mathrm{new}}$, E_a^{new}, F_a^{new}}, EU_B sends message M_{U1b}={$\mathrm{TID}_b^{\mathrm{new}}$, E_b^{new}, F_b^{new}}, EU_C sends message $M_{\mathrm{U1}c}$ ={$\mathrm{TID}_c^{\mathrm{new}}$, E_c^{new}, F_c^{new}} and EU_i sends message $M_{\mathrm{U1}i}$= {$\mathrm{TID}_i^{\mathrm{new}}$, E_i^{new}, F_i^{new}}, the attacker intercepts four messages $M_{\mathrm{U1}a}$, $\mathrm{M}_{\mathrm{U1}b}$, $\mathrm{M}_{\mathrm{U1c}}$, $\mathrm{M}_{\mathrm{U1}i}$ and uses the previous intercepted E_i and TID_i to calculate

$$\mathrm{TID}_i\oplus E_i\oplus\mathrm{TID}_a^{\mathrm{new}}=N_a,\ \mathrm{TID}_i\oplus E_i\oplus\mathrm{TID}_b^{\mathrm{new}}=N_b,$$
$$\mathrm{TID}_i\oplus E_i\oplus\mathrm{TID}_c^{\mathrm{new}}=N_c,\ \mathrm{TID}_i\oplus E_i\oplus\mathrm{TID}_i^{\mathrm{new}}=N_d. \tag{4}$$

The attacker also saves the intercepted messages M_{U1a}, M_{U1b}, M_{U1c} and $M_{\mathrm{U1}i}$. At the end of the second authentication, the new pseudonym of user EU_i has been updated to $\mathrm{TID}_i^{\mathrm{new2}}$.

Step 3: In the third authentication, assume that three users EU_J, EU_K, EU_L and the same user EU_i send messages trying to contact a fog server FS_l. The four users send messages $M_{\mathrm{U1}j}$={$\mathrm{TID}_j^{\mathrm{new\,2}}$, E_j^{new2}, F_j^{new2}}, $M_{\mathrm{U1}k}$= {$\mathrm{TID}_k^{\mathrm{new2}}$, E_k^{new2}, F_k^{new2}}, $M_{\mathrm{U1}l}${$\mathrm{TID}_l^{\mathrm{new2}}$, E_l^{new2}, F_l^{new2}}, $M_{\mathrm{U1}i}^{\mathrm{new}}$ = {$\mathrm{TID}_i^{\mathrm{new2}}$, E_i^{new2}, F_i^{new2}}, respectively. The attacker intercepts all four messages $M_{\mathrm{U1}j}$, $M_{\mathrm{U1}k}$, $M_{\mathrm{U1}l}$, and $M_{\mathrm{U1}i}^{\mathrm{new}}$ and according to the second intercepted messages $M_{\mathrm{U1}a}$, $M_{\mathrm{U1}b}$, $M_{\mathrm{U1}c}$ and $M_{\mathrm{U1}i}$ to calculate

$$\mathrm{TID}_a^{\mathrm{new}}\oplus E_a^{\mathrm{new}}\oplus\mathrm{TID}_j^{\mathrm{new2}}=N_{aj},$$
$$\mathrm{TID}_a^{\mathrm{new}}\oplus E_a^{\mathrm{new}}\oplus\mathrm{TID}_k^{\mathrm{new2}}=N_{ak},$$
$$\mathrm{TID}_a^{\mathrm{new}}\oplus E_a^{\mathrm{new}}\oplus\mathrm{TID}_l^{\mathrm{new2}}=N_{al},$$
$$\mathrm{TID}_a^{\mathrm{new}}\oplus E_a^{\mathrm{new}}\oplus\mathrm{TID}_i^{\mathrm{new2}}=N_{ai},$$
$$\mathrm{TID}_b^{\mathrm{new}}\oplus E_b^{\mathrm{new}}\oplus\mathrm{TID}_j^{\mathrm{new2}}=N_{bj},$$
$$\mathrm{TID}_b^{\mathrm{new}}\oplus E_b^{\mathrm{new}}\oplus\mathrm{TID}_k^{\mathrm{new2}}=N_{bk},$$
$$\mathrm{TID}_b^{\mathrm{new}}\oplus E_b^{\mathrm{new}}\oplus\mathrm{TID}_l^{\mathrm{new2}}=N_{bl},$$
$$\mathrm{TID}_b^{\mathrm{new}}\oplus E_b^{\mathrm{new}}\oplus\mathrm{TID}_i^{\mathrm{new2}}=N_{bi},$$
$$\mathrm{TID}_c^{\mathrm{new}}\oplus E_c^{\mathrm{new}}\oplus\mathrm{TID}_j^{\mathrm{new2}}=N_{cj},$$
$$\mathrm{TID}_c^{\mathrm{new}}\oplus E_c^{\mathrm{new}}\oplus\mathrm{TID}_k^{\mathrm{new2}}=N_{ck},$$
$$\mathrm{TID}_c^{\mathrm{new}}\oplus E_c^{\mathrm{new}}\oplus\mathrm{TID}_l^{\mathrm{new2}}=N_{cl},$$
$$\mathrm{TID}_c^{\mathrm{new}}\oplus E_c^{\mathrm{new}}\oplus\mathrm{TID}_i^{\mathrm{new2}}=N_{ci},$$
$$\mathrm{TID}_i^{\mathrm{new}}\oplus E_i^{\mathrm{new}}\oplus\mathrm{TID}_j^{\mathrm{new2}}=N_{ij},$$
$$\mathrm{TID}_i^{\mathrm{new}}\oplus E_i^{\mathrm{new}}\oplus\mathrm{TID}_k^{\mathrm{new2}}=N_{ik},$$
$$\mathrm{TID}_i^{\mathrm{new}}\oplus E_i^{\mathrm{new}}\oplus\mathrm{TID}_l^{\mathrm{new2}}=N_{il},$$
$$\mathrm{TID}_i^{\mathrm{new}}\oplus E_i^{\mathrm{new}}\oplus\mathrm{TID}_i^{\mathrm{new2}}=N_{ii}. \tag{5}$$

After calculating (4) and (5), the attacker will find that N_d = N_{ii}. This is because $\mathrm{TID}_i\oplus E_i\oplus\mathrm{TID}_i^{\mathrm{new}}=\mathrm{TID}_i^{\mathrm{new}}\oplus E_i^{\mathrm{new}}\oplus\mathrm{TID}_i^{\mathrm{new2}}=D_i\oplus A_i$, where A_i and D_i are constant, then the attacker can determine the messages M_{U1}, $M_{\mathrm{U1}i}$ and $M_{\mathrm{U1}i}^{\mathrm{new}}$ are from the same user EU_i. When a user EU_k contacts a fog server FS_n with the new pseudonym $\mathrm{TID}_k^{\mathrm{new3}}$ next time, the attacker can judge whether EU_k is EU_i by calculating $\mathrm{TID}_i^{\mathrm{new2}}\oplus E_i^{\mathrm{new2}}\oplus\mathrm{TID}_k^{\mathrm{new3}}\overset{?}{=}N_d=N_{ii}$. Thus, user traceability attack can be successful.

2.2 User Impersonation Attack

Weng *et al*'s scheme [15] cannot resist user impersonation attack and the attack can be simulated as follows.

Step 1: According to the Section 2.1, the attacker can obtain the value of $A_i\oplus D_i$ of edge user EU_i. Based on user traceability attack, the attacker intercepts the message $M_{\mathrm{U1}\,i}$={$\mathrm{TID}_i^{\mathrm{new3}}$, E_i^{new3}, F_i^{new3}} sent by EU_i to the fog server. According to (6), the attacker can calculate the random number r_u' chosen by edge user EU_i this time; then, the attacker can calculate the new pseudonym updated by the user and cloud server CS at the end of this authentication by (7).

$$r_u'=D_i\oplus A_i\oplus E_i^{\mathrm{new3}} \tag{6}$$

$$\mathrm{TID}_i^{\mathrm{new4}}=\mathrm{TID}_i^{\mathrm{new3}}\oplus r_u' \tag{7}$$

Step 2: Assume that the attacker is a legitimately registered user, then, based on the analysis in Step 1, the attacker can know $A_i\oplus D_i$, $\mathrm{TID}_i^{\mathrm{new4}}$, $h(K_{cu})$. In the login and authentication phase, the attacker selects a fog server FS_j that he/she wants to contact and picks a random number r_A, calculates

$$E_i^{\mathrm{new4}}=D_i\oplus A_i\oplus r_A,\ F_i^{\mathrm{new4}}=h(h(K_{cu})\|\mathrm{ID}_j\|r_A).$$

Then, the attacker sends the message M_{u1}^*= {$\mathrm{TID}_i^{\mathrm{new4}}$, E_i^{new4}, F_i^{new4}} to the fog server FS_j.

Step 4: Upon receiving the message M_{u1}^*, FS_j

chooses a random number r_{f1}, and calculates

$$O_{j1}=B_j\oplus r_{f1},\ P_{j1}=h(h(\mathrm{MK}\|K_{cf})\|\mathrm{TID}_j^{\mathrm{new}}\|r_{f1}).$$

FS_j sends the message $M_{\mathrm{U}1}^*, M_{\mathrm{U}2}^*=\{\mathrm{TID}_j^{\mathrm{new}}, O_{j1}, P_{j1}\}$ to the cloud server CS, where $\mathrm{TID}_j^{\mathrm{new}}$ is the fog server FS_j's pseudonym.

Step 5: On receiving messages $M_{\mathrm{U}1}^*, M_{\mathrm{U}2}^*$, CS first checks the validity of pseudonyms; it is evident that $\mathrm{TID}_i^{\mathrm{new4}}$, $\mathrm{TID}_j^{\mathrm{new}}$ is valid, and then CS computes

$$r'_A=E_i^{\mathrm{new4}}\oplus B_i\oplus h(\mathrm{MK}\|K_{cu}),\ F_i^{\mathrm{new4}'}=h(h(K_{cu})\|\mathrm{ID}_j\|r'_A),$$

$$F_i^{\mathrm{new4}'}\stackrel{?}{=}F_i^{\mathrm{new4}}.$$

This verification can pass, and the attacker can successfully impersonate user EU_i. Then, cloud server CS chooses a random number r_{c1}, and computes

$$\mathrm{TID}_i^{\mathrm{new5}}=\mathrm{TID}_i^{\mathrm{new4}}\oplus r'_A,\ \mathrm{TID}_j^{\mathrm{new2}}=\mathrm{TID}_j^{\mathrm{new}}\oplus r'_{f1},$$

$$Q_{j1}=r'_A\oplus r_{c1}\oplus h(\mathrm{ID}_j\|\mathrm{TID}_j^{\mathrm{new2}}),\ \mathrm{SK}'_{ij}=h(r'_A\oplus r_{c1}\oplus r'_{f1}),$$

$$R_{j1}=h(B_j\oplus h(\mathrm{TID}_j^{\mathrm{new2}})\oplus \mathrm{SK}'_{ij}), S_{j1}=r'_{f1}\oplus r_{c1}\oplus h(\mathrm{ID}_j\|\mathrm{TID}_i^{\mathrm{new5}}).$$

Then, the cloud server CS sends $M_{\mathrm{U}3}=\{Q_{j1}, R_{j1}\}$, $M_{\mathrm{U}4}=\{S_{j1}, T_{j1}\}$ to the fog server FS_j.

Step 6: After receiving $M_{\mathrm{U}3}$ and $M_{\mathrm{U}4}$ from CS, the fog server FS_j computes

$$\mathrm{TID}_j^{\mathrm{new2}}=\mathrm{TID}_j^{\mathrm{new}}\oplus r_{f1},\ r'_A\oplus r'_{c1}=Q_{j1}\oplus h(\mathrm{ID}_j\|\mathrm{TID}_j^{\mathrm{new2}}),$$

$$\mathrm{SK}_{ij}=h(r'_A\oplus r'_{c1}\oplus r_{f1}).$$

Step 7: The attacker can also compute the session key by

$$\mathrm{TID}_i^{\mathrm{new5}}=\mathrm{TID}_i^{\mathrm{new4}}\oplus r_A,\ r'_{f1}\oplus r'_{c1}=S_{j1}\oplus h(\mathrm{ID}_j\|\mathrm{TID}_i^{\mathrm{new5}}),$$

$$\mathrm{SK}_{ij}=h(r'_{f1}\oplus r'_{c1}\oplus r_A).$$

Eventually, the attacker and the fog server FS_j agree on a session key SK_{ij} with the help of the cloud server CS.

3 The Improved Scheme

To overcome the shortcomings of Weng *et al*'s scheme[15], we propose an improved scheme in this section. In the registration phase, we add a password to make the improved scheme more complete, and we also avoid sending $h(k_{cu})$ and $h(\mathrm{MK}\|K_{cf})$ directly, this can prevent attackers from performing impersonation attack. In the login and authentication phase, to ensure that the improved scheme is resistant to user traceability attack, we also modify the calculation of E_i and add timestamps. The following is a detailed description of the improved scheme.

3.1 Deployment Phase

The cloud server CS generates a master secret key MK and two long-term secret keys K_{cf} and K_{cu} and keeps them secret. CS chooses a one-way Hash function $h(\cdot)$.

3.2 Fog Server Registration Phase

Fog server FS_j selects a unique identity ID_j and registers itself with CS by sending identity ID_j to CS via a secure channel. On receiving ID_j, CS generates a pseudonym TID_j and a random secret number n_f and computes $B_j=h(\mathrm{ID}_j\|K_{cf})$ and $h(\mathrm{MK}\|K_{cf}\|n_f)$. CS publicizes fog server FS_j's identity ID_j. Then CS sends $\{\mathrm{TID}_j, B_j, h(\mathrm{MK}\|K_{cf}\|n_f)\}$ to FS_j and maintains $\{\mathrm{TID}_j, \mathrm{ID}_j, B_j, n_f\}$ in a protected verifier table. Finally, FS_j stores TID_j, B_j, and $h(\mathrm{MK}\|K_{cf}\|n_f)$ in its memory. The fog server registration phase is illustrated in Fig. 1.

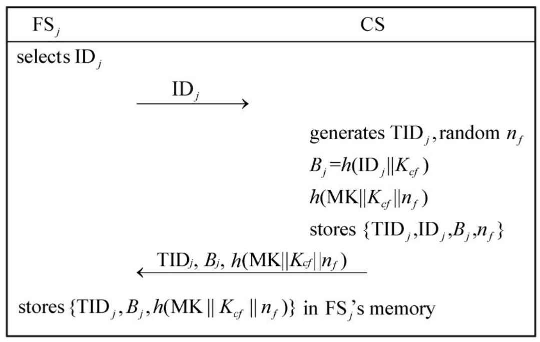

Fig. 1 Fog server registration

3.3 Edge User Registration Phase

Edge user EU_i selects a unique identity ID_i, PW_i, and imprints his/her biometric BIO_i into a smart device. EU_i's smart device MD_i generates a random secret number n_u and computes

$$\mathrm{Gen}(\mathrm{BIO}_i)=(\sigma_i,\tau_i),\ \mathrm{PPW}_i=h(\mathrm{PW}_i\|n_u),$$

$$A_i=h(\mathrm{ID}_i\|\mathrm{PPW}_i\|\sigma_i\|n_u).$$

Then MD_i sends $\{A_i, \mathrm{ID}_i, \mathrm{PPW}_i\}$ to CS through a secure channel. After receiving ID_i, A_i and PPW_i, CS generates a psedonym TID_i and computes

$$B_i=h(\mathrm{ID}_i\|\mathrm{MK}),\ C_i=h(\mathrm{ID}_i\|A_i\|h(K_{cu}\|\mathrm{PPW}_i)),$$

$$D_i=B_i\oplus h(\mathrm{MK}\|K_{cu})\oplus A_i,\ H_i=\mathrm{PPW}_i\oplus h(\mathrm{MK}\|K_{cu}).$$

Then CS sends $\{\mathrm{TID}_i, C_i, D_i, h(\cdot), h(K_{cu}\|\mathrm{PPW}_i)\}$ to EU_i via a secure channel and maintains $\{\mathrm{TID}_i, B_i, H_i\}$ in a protected verifier table. Finally, MD_i computes

$$K=h(K_{cu}\|\mathrm{PPW}_i)\oplus h(\mathrm{PPW}_i\|\sigma_i),\ \mathrm{TID}'_i=\mathrm{TID}_i\oplus h(\mathrm{ID}_i\|\sigma_i).$$

Then, MD_i stores $\{\mathrm{TID}'_i, C_i, D_i, h(\cdot), K, n_u, \mathrm{Rep}(\cdot), \tau_i\}$ in its memory. Edge user registration phase is explained in Fig. 2.

3.4 Authentication and Key Agreement Phase

In this phase, if an edge user EU_i wants to access a fog server FS_j through a public channel, EU_i and FS_j au-

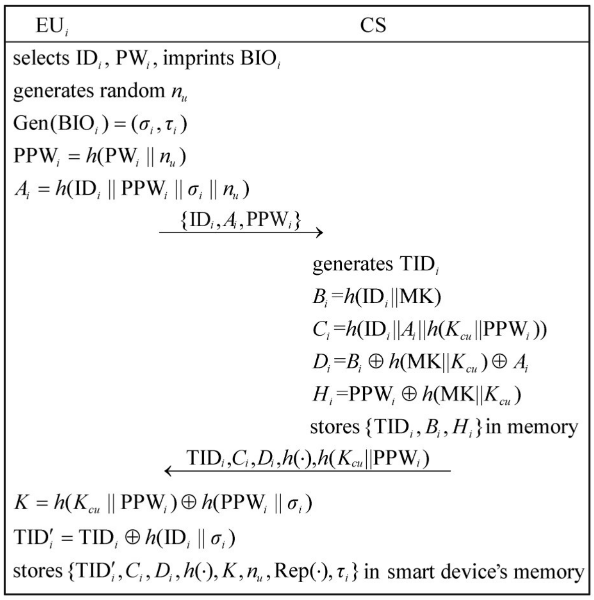

Fig. 2 Edge user registration

thenticate each other with the help of the cloud server CS and establish a session key SK_{ij}. The detailed steps are as follows and the detailed process is explicated in Fig. 3.

Step 1: EU_i first inputs $\mathrm{ID}_i', \mathrm{PW}_i'$ and imprints BIO_i' into his/her smart device MD_i. Then, MD_i computes

$$\sigma_i' = \mathrm{Rep}(\mathrm{BIO}_i', \tau_i),\ \mathrm{PPW}_i' = h(\mathrm{PW}_i' \| n_u),$$

$$A_i' = h(\mathrm{ID}_i' \| \mathrm{PPW}_i' \| \sigma_i' \| n_u),\ h(K_{cu} \| \mathrm{PPW}_i)' = K \oplus h(\mathrm{PPW}_i' \| \sigma_i'),$$

$$C_i' = h(\mathrm{ID}_i' \| A_i' \| h(K_{cu} \| \mathrm{PPW}_i)'),$$

checks whether $C_i' = C_i$. If it does not hold, the smart device MD_i rejects the request and terminates the session. Otherwise, it means that this user is a legitimate smart device holder. Then, user EU_i selects a fog server FS_j that he/she wants to contact, selects a random number r_u, and generates a current timestamp T_1. MD_i computes

$$\mathrm{TID}_i = \mathrm{TID}_i' \oplus h(\mathrm{ID}_i \| \sigma_i),\ M_i = r_u \oplus h(D_i \oplus A_i \oplus T_1),$$

$$V_i = h(h(K_{cu} \| \mathrm{PPW}_i) \| \mathrm{ID}_j \| r_u \| T_1).$$

Finally, MD_i sends the request message $M_{u1} = \{\mathrm{TID}_i, M_i, V_i, T_1\}$ to FS_j through a public channel.

Step 2: On receiving M_{u1}, FS_j first checks the freshness of the message. Then, FS_j selects a random number r_f and a current timestamp T_2, and retrieves TID_j, B_j, and $h(\mathrm{MK} \| K_{cf} \| n_f)$ to compute

$$M_j = r_f \oplus h(B_j \| T_2),\ V_j = h(h(\mathrm{MK} \| K_{cf} \| n_f) \| \mathrm{TID}_j \| r_f \| T_2).$$

Finally, FS_j sends messages M_{u1} and $M_{u2} = \{\mathrm{TID}_j, M_j, V_j, T_2\}$ to CS through a public channel.

Step 3: On receiving M_{u1} and M_{u2}, CS first checks the freshness of the messages. If true, CS retrieves B_i, H_i and B_j, ID_j, n_f according to TID_i and TID_j, respectively. Then CS computes

$$\mathrm{PPW}_i' = H_i \oplus h(\mathrm{MK} \| K_{cu}),\ r_u' = M_i \oplus h(B_i \oplus h(\mathrm{MK} \| K_{cu}) \oplus T_1),$$

$$V_i' = h(h(K_{cu} \| \mathrm{PPW}_i') \| \mathrm{ID}_j \| r_u' \| T_1).$$

CS checks whether $V_i' = V_i$. If it is not true, CS terminates the session. Otherwise, the legitimacy of EU_i is authenticated by CS. CS further computes

$$r_f' = M_j \oplus h(B_j \| T_2),\ V_j' = h(h(\mathrm{MK} \| K_{cf} \| n_f) \| \mathrm{TID}_j \| r_f' \| T_2).$$

Then CS checks whether $V_j' = V_j$.

Step 4: After verifying the validity of EU_i and FS_j, CS refreshes new pseudonyms for EU_i and FS_j by computing

$$\mathrm{TID}_i^{\mathrm{new}} = \mathrm{TID}_i \oplus r_u',\ \mathrm{TID}_j^{\mathrm{new}} = \mathrm{TID}_j \oplus r_f'$$

and replaces TID_i with $\mathrm{TID}_i^{\mathrm{new}}$ in the verifier table of EU_i, replaces TID_j with $\mathrm{TID}_j^{\mathrm{new}}$ in the verifier table of FS_j. CS further selects a random number r_c and a current timestamp T_3 to compute

$$H_j = r_u' \oplus r_c \oplus h(ID_j \| \mathrm{TID}_j^{\mathrm{new}}),\ \mathrm{SK}_{ij} = h(r_u' \oplus r_f' \oplus r_c),$$

$$I_j = h(B_j \oplus h(\mathrm{TID}_j^{\mathrm{new}}) \oplus \mathrm{SK}_{ij} \oplus T_3),$$

$$J_i = r_f' \oplus r_c \oplus h(\mathrm{ID}_j \| \mathrm{TID}_i^{\mathrm{new}}),$$

$$K_i = h(B_i \oplus h(\mathrm{MK} \| K_{cu}) \oplus h(\mathrm{TID}_i^{\mathrm{new}}) \oplus \mathrm{SK}_{ij} \oplus T_3).$$

Finally, CS sends $M_{u3} = \{H_j, I_j, T_3\}$ and $M_{u4} = \{J_i, K_i, T_3\}$ to FS_j.

Step 5: On receiving M_{u3} and M_{u4}, FS_j computes

$$\mathrm{TID}_j'^{\mathrm{new}} = \mathrm{TID}_j \oplus r_f,\ r_u' \oplus r_c' = H_j \oplus h(\mathrm{ID}_j \| \mathrm{TID}_j'^{\mathrm{new}}),$$

$$\mathrm{SK}_{ij} = h(r_u' \oplus r_c' \oplus r_f),\ I_j' = h(B_j \oplus h(\mathrm{TID}_j'^{\mathrm{new}}) \oplus \mathrm{SK}_{ij} \oplus T_3).$$

FS_j checks if $I_j' = I_j$. If it holds, FS_j sends M_{u4} to EU_i.

Step 6: On receiving M_{u4}, EU_i computes

$$\mathrm{TID}_i'^{\mathrm{new}} = \mathrm{TID}_i \oplus r_u,\ r_f' \oplus r_c' = J_i \oplus h(\mathrm{ID}_j \| \mathrm{TID}_i'^{\mathrm{new}}),$$

$$\mathrm{SK}_{ij} = h(r_f' \oplus r_c' \oplus r_u),\ K_i' = h(D_i \oplus A_i \oplus h(\mathrm{TID}_i^{\mathrm{new}}) \oplus \mathrm{SK}_{ij} \oplus T_3).$$

If $K_i' = K_i$, EU_i believes that CS and FS_j are legal parties and stores the shared session key SK_{ij} for future secure communication. Then, EU_i computes

$$\mathrm{TID}_i'^{\mathrm{new}} = \mathrm{TID}_i^{\mathrm{new}} \oplus h(\mathrm{ID}_i \| \sigma_i),$$

and replaces TID_i' with $\mathrm{TID}_i'^{\mathrm{new}}$ in smart device's memory.

4 Security Analysis and Comparisons

4.1 Security Analysis

This section analyzes security of the improved scheme. We demonstrate the improved scheme's security features and resilience against various attacks.

4.1.1 User anonymity

The improved scheme has strong anonymity. During the authentication and key agreement phase, the edge user's identity is never shared openly or sent over

EU_i	FS_j	CS
Inputs ID_i', PW_i' and imprints BIO_i' $\sigma_i' = \mathrm{Rep}(BIO_i', \tau_i)$ $PPW_i' = h(PW_i' \parallel n_u)$ $A_i' = h(ID_i' \parallel PPW_i' \parallel \sigma_i' \parallel n_u)$ $h(K_{cu} \parallel PPW_i)' = K \oplus h(PPW_i' \parallel \sigma_i')$ $C_i' = h(ID_i' \parallel A_i' \parallel h(K_{cu} \parallel PPW_i'))$ checks if $C_i' = C_i$ selects r_u, generates timestamp T_1 $TID_i = TID_i' \oplus h(ID_i \parallel \sigma_i)$ $M_i = r_u \oplus h(D_i \oplus A_i \oplus T_1)$ $V_i = h(h(K_{cu} \parallel PPW_i) \parallel ID_j \parallel r_u \parallel T_1)$ $\xrightarrow{M_{u1}=\{TID_i, M_i, V_i, T_1\}}$		
	Verifies the validity of T_1 selects r_f, generates timestamp T_2 retrieves $TID_j, B_j, h(MK \parallel K_{cf} \parallel n_f)$ $M_j = r_f \oplus h(B_j \parallel T_2)$ $V_j = h(h(MK \parallel K_{cf} \parallel n_f) \parallel TID_j \parallel r_f \parallel T_2)$ $\xrightarrow{M_{u1}, M_{u2}=\{TID_j, M_j, V_j, T_2\}}$	
		Verifies the validity of T_1, T_2 checks validity of TID_i, TID_j $PPW_i' = H_i \oplus h(MK \parallel K_{cu})$ $r_u' = M_i \oplus h(B_i \oplus h(MK \parallel K_{cu}) \oplus T_1)$ $V_i' = h(h(K_{cu} \parallel PPW_i') \parallel ID_j \parallel r_u' \parallel T_1)$ checks if $V_i' = V_i$ $r_f' = M_j \oplus h(B_j \parallel T_2)$ $V_j' = h(h(MK \parallel K_{cf} \parallel n_f) \parallel TID_j \parallel r_f' \parallel T_2)$ checks if $V_j' = V_j$ $TID_i^{new} = TID_i \oplus r_u'$ $TID_j^{new} = TID_j \oplus r_f'$ selects r_c, generates timestamp T_3 $H_j = r_u' \oplus r_c \oplus h(ID_j \parallel TID_j^{new})$ $SK_{ij} = h(r_u' \oplus r_f' \oplus r_c)$ $I_j = h(B_j \oplus h(TID_j^{new}) \oplus SK_{ij} \oplus T_3)$ $J_i = r_f' \oplus r_c \oplus h(ID_j \parallel TID_i^{new})$ $K_i = h(B_i \oplus h(MK \parallel K_{cu}) \oplus h(TID_i^{new}) \oplus SK_{ij} \oplus T_3)$ $\xleftarrow{M_{u3} = \{H_j, I_j, T_3\}, M_{u4} = \{J_i, K_i, T_3\}}$
	$TID_j'^{new} = TID_j \oplus r_f$ $r_u' \oplus r_c' = H_j \oplus h(ID_j \parallel TID_j'^{new})$ $SK_{ij} = h(r_u' \oplus r_c' \oplus r_f)$ $I_j' = h(B_j \oplus h(TID_j'^{new}) \oplus SK_{ij} \oplus T_3)$ checks if $I_j' = I_j$ $\xleftarrow{M_{u4} = \{J_i, K_i, T_3\}}$	
$TID_i'^{new} = TID_i \oplus r_u$ $r_f' \oplus r_c' = J_i \oplus h(ID_j \parallel TID_i'^{new})$ $SK_{ij} = h(r_f' \oplus r_c' \oplus r_u)$ $K_i' = h(D_i \oplus A_i \oplus h(TID_i^{new}) \oplus SK_{ij} \oplus T_3)$ $TID_i'^{new} = TID_i^{new} \oplus h(ID_i \parallel \sigma_i)$		

Fig. 3 Authentication and key agreement phase

the public channel; instead, the edge user sends his/her pseudonym TID_i. Even if an attacker intercepts the edge user's pseudonym TID_i, he/she cannot know the user's identity ID_i.

4.1.2 Password guessing attack

Assuming that an attacker gets all information $\{\mathrm{TID}_i', C_i, D_i, h(\cdot), K, n_u, \mathrm{Rep}(\cdot), \tau_i\}$ stored in the mobile device, the attacker performs password guessing attack based on $\mathrm{PPW}_i = h(n_u \| \mathrm{PW}_i)$. However, the attacker must know the values of PPW_i and n_u. Although the attacker knows the value of n_u, he/she cannot know the value of PPW_i. So, the attacker cannot guess the password.

4.1.3 Replay attack

An attacker may resend previous messages to cloud server for replay attack. However, this will not succeed because the improved scheme contains timestamp verifications, and the timestamps in past messages are not within an acceptable range. Resending past messages will cause the session to terminate. Even if an attacker can modify the timestamp T_1, the attacker does not know D_i, A_i, and $h(K_{cu} \| \mathrm{PPW}_i)$, then the corresponding values M_i and V_i cannot be modified based on the new timestamp. Finally, the value of V_i sent by the attacker differs from V_i' calculated by the cloud server CS by (8), and verification $V_i' \stackrel{?}{=} V_i$ cannot pass. Therefore, the improved scheme can resist replay attack.

$$V_i' = h(h(K_{cu} \| \mathrm{PPW}_i) \| \mathrm{ID}_j \| r_u \| T_1) \tag{8}$$

4.1.4 User untraceability

In the mutual authentication and key agreement phase, the user's identity is hidden by sending the pseudonym TID_i, and the user's pseudonym TID_i is updated every time by (9). Due to the inclusion of random number r_u in (9), the user's pseudonym will be different each time. Therefore, the attacker does not know what the user's next pseudonym $\mathrm{TID}_i^{\mathrm{new}}$ will be.

$$\mathrm{TID}_i^{\mathrm{new}} = \mathrm{TID}_i \oplus r_u \tag{9}$$

When the attacker intercepts EU_i's request message $M_{u1} = \{\mathrm{TID}_i, M_i, V_i, T_1\}$, and stores $\{\mathrm{TID}_i, M_i\}$ as described in Section 2.1.

In the second authentication, even if the attacker intercepts EU_i's message $M_{u1}' = \{\mathrm{TID}_i^{\mathrm{new}}, M_i', V_i', T_1'\}$, the attacker computes

$$M_i \oplus \mathrm{TID}_i^{\mathrm{new}} \oplus \mathrm{TID}_i = h(D_i \oplus A_i \oplus T_1).$$

In the third authentication, assuming that the attacker intercepts EU_i's request message $M_{u1}^2 = \{\mathrm{TID}_i^{\mathrm{new2}}, M_i^2, V_i^2, T_1^2\}$ once again, the attacker computes

$$M_i' \oplus \mathrm{TID}_i^{\mathrm{new2}} \oplus \mathrm{TID}_i^{\mathrm{new}} = h(D_i \oplus A_i \oplus T_1').$$

Because $h(D_i \oplus A_i \oplus T_1)$ and $h(D_i \oplus A_i \oplus T_1')$ contain timestamps T_1, T_1', which makes the value of $h(D_i \oplus A_i \oplus T_1)$ is not equal to the value of $h(D_i \oplus A_i \oplus T_1')$. Therefore, even if the attacker intercepts messages from the same user EU_i, the result of $M_i \oplus \mathrm{TID}_i^{\mathrm{new}} \oplus \mathrm{TID}_i$ will be different every time, and the user traceability attack cannot be performed.

4.1.5 User impersonation attack

If an attacker intercepts $M_{u1} = \{\mathrm{TID}_i, M_i, V_i, T_1\}$ sent by edge user EU_i on the public channel and assumes that the attacker obtains the user's data stored in the mobile device's memory $\{\mathrm{TID}_i', C_i, D_i, h(\cdot), K, n_u, \mathrm{Rep}(\cdot), \tau_i\}$, if an attacker wants to impersonate the user to spoof the cloud server successfully, he/she must construct new M_i and V_i based on (10) and (11), so that the verification of the V_i can pass. Still, the attacker cannot construct a correct M_i and V_i, because the attacker cannot obtain the values of D_i, A_i, and $h(K_{cu} \| \mathrm{PPW}_i)$. Therefore, the attacker cannot impersonate an edge user.

$$M_i = r_u \oplus h(D_i \oplus A_i \oplus T_1), \tag{10}$$

$$V_i = h(h(K_{cu} \| \mathrm{PPW}_i) \| \mathrm{ID}_j \| r_u \| T_1). \tag{11}$$

4.1.6 Session key agreement

The improved scheme can achieve secure session key agreement. Edge user, fog server, and the cloud server CS can compute a shared session key $\mathrm{SK}_{ij} = h(r_u \oplus r_f \oplus r_c)$. And the attacker cannot compute the correct session key because the random numbers r_u, r_f and r_c chosen by the three entities are protected by (12), (13), (14), (15); the attacker does not know the value of the random number r_u, r_f and r_c in the session key, so security of the session key in the improved scheme is guaranteed.

$$M_i = r_u \oplus h(D_i \oplus A_i \oplus T_1) \tag{12}$$

$$M_j = r_f \oplus h(B_j \| T_2) \tag{13}$$

$$H_j = r_u' \oplus r_c \oplus h(\mathrm{ID}_j \| \mathrm{TID}_j^{\mathrm{new}}) \tag{14}$$

$$J_i = r_f' \oplus r_c \oplus h(\mathrm{ID}_j \| \mathrm{TID}_i^{\mathrm{new}}) \tag{15}$$

4.1.7 Mutual authentication

In the improved scheme, edge users, fog servers, and the cloud server can authenticate each other's identity. For example, the cloud server can authenticate the edge user's identity through the verification of V_i in (16), because the calculation of V_i includes $h(K_{cu} \| \mathrm{PPW}_i)$, only edge user EU_i and the cloud server CS know the value of $h(K_{cu} \| \mathrm{PPW}_i)$. Cloud server CS authenticates the fog server's identity via V_j in (17), because only fog server FS_j and CS know the value of $h(\mathrm{MK} \| K_{cf} \| n_f)$. Therefore, the attacker cannot impersonate the edge user or fog server.

$$V_i = h(h(K_{cu}||\mathrm{PPW}_i)||\mathrm{ID}_j||r_u||T_1) \quad (16)$$

$$V_j = h(h(\mathrm{MK}||K_{cf}||n_f)||\mathrm{TID}_j||r_f||T_2) \quad (17)$$

4.1.8 Denial of service attack

The improved scheme can resist denial of service attack. If an attacker wants to waste network resources for DOS attack, his fake messages must pass a verification. However, according to our analysis above, if the attacker impersonates user EU_i and sends messages to CS, this fake messages cannot pass the verification of V_i. Even if the attacker changes the timestamp, he/she cannot calculate the correct M_i, V_i, and V_j based on the changed timestamp. Therefore, V_i and V_j constructed by the attacker fail to pass the verification of the cloud server.

4.1.9 Perfect forward security

EU_i and FS_j negotiate the session key $\mathrm{SK}_{ij} = h(r_u \oplus r_f \oplus r_c)$, where r_u, r_f, and r_c are all random numbers. The attacker cannot compute the correct session key because he/she does not know the values of the random numbers r_u, r_f, and r_c in the session key. The attacker cannot compute the following session key based on present random numbers, and if this session key is compromised, it does not affect the security of past and future session keys.

4.2 Comparison

This section compares the improved scheme with Weng *et al*'s scheme[15] and the other three schemes[3,16,17]. According to Weng *et al*'s scheme[15], some symbols (H, X, Y, F) are defined and the execution times of these operations are given in Table 2. The results of the comparison of security features and computation costs are shown in Table 3 and Table 4, respectively.

Table 2 Approximate time required for various operations

Notation	Description	Computation time / ms
H	Hash function	0.5
X	XOR operation	0.1
Y	ECC point multiplication	63.075
F	Fuzzy extractor function	63.075

As can be seen from Table 3, the improved scheme can resist various known attacks and has higher security than Weng *et al*'s scheme[15] and the other three schemes[3,16,17]. From Table 4, it can be seen that our improved scheme requires less computation than Jia *et al*' scheme[3], Chen *et al*'s scheme[17],and Ma *et al*'s scheme[16].

Table 3 Security comparison of different schemes

Attack	Ma *et al* [16]	Jia *et al* [3]	Chen *et al* [17]	Weng *et al* [15]	Our scheme
User anonymity	×	√	√	√	√
Replay attack	√	√	√	√	√
User untraceability	√	√	√	×	√
Impersonation attack	√	√	√	×	√
Session key agreement	√	√	×	√	√
Insider attack	√	×	√	√	√
Possword guessing attack	√	√	×	√	√
Stolen smart card attack	√	×	√	√	√
Perfect forward security	×	√	√	√	√
Man-in-the-middle attack	√	√	√	√	√

Table 4 Comparison of computation costs

Scheme	Registration phase	Login phase	Authentication phase	Total computation cost	Total/ms
Ma *et al* [16]	2H+3Y	—	17H+4X+15Y	19H+4X+18Y	1 145.25
Jia *et al* [3]	Y+3H+3X	2Y	18H+5X+8Y	21H+8X+11Y	705.125
Chen *et al* [17]	8H+4X+F	2H+F	22H+24X+5Y	32H+28X+5Y+2F	460.325
Weng *et al* [15]	7H+2X	3H	25H+32X	35H+34X	20.9
Our scheme	10H+4X+F	3H+X+F	30H+38X	43H+43X+2F	151.95

Because of the use of fuzzy extractor, the improved scheme requires more computation than Weng *et al*'s scheme[15]. However, the improved scheme can resist various known attacks and is more secure. It is worthwhile to add some necessary computations to ensure communication security.

5 Conclusion

In this paper, we find that Weng *et al*'s authentication scheme is not resistant to user traceability attack and user impersonation attack. We propose an improved scheme to overcome the weaknesses of Weng *et al*'s scheme. We compare the improved scheme with several existing authentication schemes in terms of security features and calculation costs. The improved scheme not only completely overcomes the drawbacks of Weng *et al*'s scheme but also meets the lightweight feature. Therefore, the improved scheme is suitable for use in fog computing.

References

[1] Hamada M, Salem S A, Salem F M. LAMAS: Lightweight anonymous mutual authentication scheme for securing fog computing environments[J]. *Ain Shams Engineering Journal*, 2022, **13**(6): 101752-101766.

[2] Wazid M, Das A K, Kumar N, *et al*. Design of secure key management and user authentication scheme for fog computing services[J]. *Future Generation Computer Systems*, 2019, **91**(2): 475-492.

[3] Jia X, He D, Kumar N, *et al*. Authenticated key agreement scheme for fog-driven IoT healthcare system[J]. *Wireless Networks*, 2019, **25**(8): 4737-4750.

[4] Lamport L. Password authentication with insecure communication[J]. *Communications of the ACM*, 1981, **24**(11): 770-772.

[5] Hwang M S, Li L H. A new remote user authentication scheme using smartcards[J]. *IEEE Transactions on Consumer Electronics*, 2000, **46**(1): 28-30.

[6] Juang W S. Efficient multi-server password authenticated key agreement using smart cards[J]. *IEEE Transactions on Consumer Electronics*, 2004, **50**(4): 251-255.

[7] Das M L. Two-factor user authentication in wireless sensor networks[J]. *IEEE Transactions on Wireless Communications*, 2009, **8**(3): 1086-1090.

[8] Kang B Y. Cryptanalysis on an E-voting scheme over computer network[J]. 2008 *International Conference on Computer Science and Software Engineering*, 2008, **3**: 826-829.

[9] Xue K, Ma C, Hong P, *et al*. A temporal-credential-based mutual authenticationand key agreement scheme for wireless sensor networks[J]. *Journal of Network and Computer Applications*, 2013, **36**(1): 316-323.

[10] Kalra S, Sood S K. Advanced password based authentication scheme for wireless sensor networks[J]. *Journal of Information Security and Applications*, 2015, **20**: 37-46.

[11] Arij B A, Abid M, Meddeb A. A privacy-preserving authentication scheme in an edge-fog environment[J]. *ACS International Conference on Computer Systems and Applications*, 2018, **2017**(10): 1225-1231.

[12] Wu T Y, Yang L, Meng Q, *et al*. Fog-driven secure authentication and key exchange scheme for wearable health monitoring system[J]. *Security and Communication Networks*, 2021, **2021**(4): 1-14.

[13] Guo Y M, Zhang Z F, Guo Y J. Fog-centric authenticated key agreement scheme without trusted parties[J]. *IEEE Systems Journal*, 2021, **15**(4): 5057-5066.

[14] Ali Z, Chaudhry S A, Mahmood K, *et al*. A clogging resistant secure authentication scheme for fog computing services [J]. *Computer Networks*, 2021, **185**: 107731.

[15] Weng C Y, Li C T, Chen C L, *et al*. A lightweight anonymous authentication and secure communication scheme for fog computing services[J]. *IEEE Access*, 2021, **9**(4): 145522-145537.

[16] Ma M M, He D B, Wang H Q, *et al*. An efficient and provably secure authenticated key agreement protocol for fog-based vehicular ad-hoc networks[J]. *IEEE Internet Things*, 2019, **6** (5): 8065-8075.

[17] Chen C M, Huang Y, Wang K H, *et al*. A secure authenticated and key exchange scheme for fog computing[J]. *Enterprise Information Systems*, 2021, **15**(1): 1-16.

[18] Rana M, Mahmood K, Saleem MA, *et al*. Towards a provably secure authentication protocol for fog-driven IoT-based systems[J]. *Applied Sciences*, 2023, **13**(3): 1424-1424.

□