**Frontiers in Computing and Intelligent Systems**
ISSN: 2832-6024 | Vol. 4, No. 3, 2023

# Analysis and Improvement of Authentication Schemes for Industrial Wireless Sensor Networks with Fog Computing

**Yuyan Huo [1], Baoyuan Kang [1], Xinyu Zuo [1], Shufang Niu [2], Anqian Li [1]**

[1] School of Software, Tiangong University, Tianjin 300387, China
[2] School of Computer Science and Technology, Tiangong University, Tianjin, China

**Abstract:** Cloud computing enables access to needed resources from a shared pool of configurable computing resources anytime, anywhere. Cloud computing offers many benefits, such as security and reliability of data and convenience of resource sharing. But as more and more devices are accessed, the demand for network bandwidth increases and it is because cloud computing centralizes all the resources that risks are also centralized. To overcome the shortcomings of cloud computing, the concept of fog computing has been introduced. Fog computing supports high mobility and has a wide geographical distribution, it also delivers data with very low latency. However, as an extension of cloud computing, fog computing inherits the security and privacy issues of cloud computing. Therefore, identity authentication in a fog computing environment is essential. In 2022, Sahoo et al. proposed an authentication scheme for industrial wireless sensor networks with fog computing. However, we analyze the security of Sahoo et al.'s scheme and find that many places in the data distribution phase are not clearly explained and their scheme is also not resistant to user impersonation attack, tracking attack, denial of service attack and replay attack. In order to overcome the weaknesses of Sahoo et al.'s scheme, this paper proposes an improved scheme by making full use of random numbers and timestamps. After security analysis and comparison with some similar schemes, it is shown that the improved scheme can resist various known attacks and has smaller computational cost.

**Keywords:** Authentication Schemes; Fog Computing; Cloud Computing.

## 1. Introduction

Cloud computing integrates various dispersed computing, storage and application resources, so cloud computing has powerful computing and massive data storage capacity, effectively provides all kinds of resources to users in the form of services [1]. Cloud computing mainly includes the advantages of low cost, high computing efficiency, saving time and high availability. However, cloud computing certainly involves a lot of interactions between the cloud and the edge and this requires a wide enough network bandwidth as well as fast enough network transmission speed, but this will cause many problems, such as network congestion, high latency, low quality of service and large bandwidth usage. In order to address the shortcomings of cloud computing, the concept of fog computing was introduced. Fog computing is an extension and supplement of cloud computing. In fog computing, fog servers are deployed at the edge of the network and close to geographically terminal facilities [2], so some data that is not needed temporarily or urgently can be processed and stored directly in the fog layer. Therefore, fog computing can assist cloud servers to complete data storage and analysis in the locality [3] and can effectively improve the transmission rate and decrease the latency, greatly reducing pressure on the cloud. However, fog nodes are often deployed in remote and unprotected locations [4] and the data transmissions between users and fog servers, fog servers and fog servers, fog servers and cloud servers rely on common channels. Therefore, authentication in fog computing environments is critical.

In 1981, Lampot et al. first proposed a remote authentication scheme in insecure environments [5]. Then, many two-factor authentication schemes based on hashes, smart cards, etc. have been proposed [6-9]. However, researches have shown that the majority of two-factor authentication schemes are insecure. As a result, three factor authentication schemes based on identity, password and biometrics [4,10-16] were proposed. In 2019, Jia et al. [4] proposed an authenticated key agreement scheme for fog-driven IoT healthcare system and claimed that their scheme is resistant to various known attacks and has low communication and computational costs. However, Wu et al. [10] pointed out that Jia et al.'s scheme[4] is not resistant to tracking attack, impersonation attack and does not provide user and fog node anonymity. In 2020, Chen et al. [11] proposed a secure authentication and key exchange scheme for fog computing and claimed that their scheme is more effective in performance and security compared to previous schemes. However, after analysis, Rana et al. [12] found that Chen et al.'s scheme 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 2022, Sahoo et al. [16] proposed an authentication scheme for industrial wireless sensor networks with fog computing and claimed that the proposed scheme ensures the confidentiality of the session key during communication and compared it with the existing schemes in terms of communication cost and storage cost, and results shows that the scheme has good practicality. However, this paper finds that many places in the data distribution phase are not clearly explained and Sahoo et al.'s scheme is also not resistant to user impersonation attack, tracking attack, denial of service attack and replay attack. To address the security issues of Sahoo et al.'s scheme, this paper proposes an improved scheme and performs security analysis on the improved scheme. We also compare the improved scheme with authentication schemes in the same direction in security and computational cost, and the result shows that the improved scheme is effective in ensuring security of the scheme under the conditions of low computational cost.

In this paper, Section 2 reviews Sahoo et al.'s scheme, Section 3 analyses the shortcomings of Sahoo et al.'s scheme, Section 4 proposes an improved scheme, Section 5 analyses security of the improved scheme, Section 6 compares the improved scheme with similar directional schemes in safety and computational cost and Section 7 draws conclusions.

# 2. Review of Sahoo et al.'s Scheme

The main problem of Sahoo et al.'s scheme is in the data distribution phase, for the sake of brevity, in this section, we will only review user and fog server registration phase and data distribution phase. The symbols and meanings used in this paper are listed in Table 1.

First, CS chooses a base point P on the elliptic curve E(GF($p$)) and the order of P is $n$. Then, CS chooses master keys $x$, $y$ and computes the public key $P_{ub}= y\cdot P$. CS keeps {x, y} as private key, making {P, $P_{ub}$} as public.

**Table 1.** Notations table

| Notation | Description |
|---|---|
| $U_i$, $ID_i$ | The $i$th user and his identity |
| $PW_i$, $BM_i$ | the $i$th user's password and biometric |
| Gen(·),Rep(·) | Fuzzy generation and reproduction function |
| $\theta_i$, $\omega_i$ | Biometric secret key and Public reproduction parameter of $EU_i$ |
| $FS_j$, $BID_j$ | The $j$th fog server and its identity |
| $SN_k$,$IDS_k$ | The $k$th sensor node and its identity |
| CS | cloud server |
| $x$, y | master keys chosen by CS |
| $SK_{uc\text{-}cs}$ | shared key between user $U_i$ and cloud server CS |
| $SK_{bs\text{-}cs}$ | shared key between fog server $FS_j$ and cloud server CS |
| $E_k(\cdot)$,$D_k(\cdot)$ | Symmetric encryption functions and symmetric decryption functions |
| $N_2$,$N_3$,$r_f$,$r_u$,$r_j$ | random numbers |
| ΔT | allowed maximum delay |

## 2.1. Registration Phase

In this section, users and fog servers register with the cloud server CS via a secure channel.

### 2.1.1. Uesr Registration Phase

The user registers with the cloud server by the following steps:

1) The user $U_i$ selects his identity $ID_i$, password $PW_i$ and extracts biometric $BM_i$. Using fuzzy extractor, $U_i$ calculates $\text{Gen}(BM_i) = (\omega_i, \theta_i)$, $PW_{i1} = h(PW_i \,||\, \omega_i)$ and then sends {$ID_i$, $PW_{i1}$} to the cloud server CS via a secure channel.

2) Upon receiving the message {$ID_i$, $PW_{i1}$}, CS calculates

$$A_i = h(ID_i \,||\, x),$$

$$B_i = h(ID_i \,||\, PW_{i1} \,||\, A_i),$$

$$C_i = h(ID_i \,||\, PW_{i1} \,||\, SK_{uc\text{-}cs}),$$

$$D_i = SK_{uc\text{-}cs} \oplus h(x).$$

Then, CS stores {$B_i$,$C_i$,$D_i$,$SK_{uc\text{-}cs}$,h(·)} into a smart card (SC) and sends the SC to user $U_i$. At the same time, CS stores {$B_i$,$C_i$,$SK_{uc\text{-}cs}$} into its database.

3) After receiving the SC from CS, $U_i$ computes

$$SK'_{uc\text{-}cs} = SK_{uc\text{-}cs} \oplus PW_{i1},$$

$$E'_i = B_i \oplus h(ID_i \,||\, PW_i),$$

and replace $B_i$ and $SK_{uc\text{-}cs}$ with $E'_i$ and $SK'_{uc\text{-}cs}$. Finally, the SC contains { $C_i, D_i, E'_i, SK'_{uc\text{-}cs}, \theta_i$, Rep(·), $h$(·) }.

### 2.1.2. Fog Server Registration Phase

In this phase, $FS_j$ registers with the cloud server via a secure channel. $FS_j$ chooses its identity $BID_j$ and sends the sensor identity $IDS_k$ and $BID_j$ to the CS. After receiving {$BID_j$,$IDS_k$}, the cloud server CS computes $R_1 = h(BID_j \,||\, IDS_k \,||\, y \,||\, SK_{bs\text{-}cs})$ , $R_2 = h(y) \oplus R_1$ , CS stores {$R_1$,$SK_{bs\text{-}cs}$} in its database and sends {$R_1$,$R_2$,$SK_{bs\text{-}cs}$} to the fog server. Then, $FS_j$ stores {$BID_j$, $R_1$,$R_2$,$SK_{bs\text{-}cs}$} in its own database.

### 2.1.3. Data Distribution Phase

In this subsection, user $U_i$, fog server $FS_j$ and cloud server CS authenticate each other and agree on a session key, then user $U_i$ can communicate securely with fog server $FS_j$. The specific steps in this subsection are shown below:

1) $U_i$ inserts his SC into card reader and inputs his identity $ID_i$, password $PW_i$ and imprints biometric $BM_i$, SC retrieves $\omega_i = Rep(BM_i, \theta_i)$ and calculates

$$PW^*_{i1} = h(PW_i \,||\, \omega_i),$$

$$SK^*_{uc\text{-}cs} = SK'_{uc\text{-}cs} \oplus PW^*_{i1}.$$

Further, SC verifies whether $C_i \stackrel{?}{=} h(ID_i \,||\, PW^*_{i1} \,||\, SK^*_{uc\text{-}cs})$ holds or not. If the condition fails, the session is aborted. Otherwise, SC generates a random number $N_2$ and current timestamp $T_u$ and calculates

$$B^*_i = E'_i \oplus h(ID_i \,||\, PW_i),$$

$$L_3 = h(\text{ID}_i \,||\, SK^*_{uc\text{-}cs} \,||\, B^*_i \,||\, N_2 \,||\, T_u),$$

$$L_4 = \text{ID}_i \oplus h(C_i \,||\, h(x)),$$

$$L_5 = N_2 \oplus h(\text{ID}_i \,||\, SK^*_{uc\text{-}cs}).$$

Then, SC transmits request message{$L_3$,$L_4$,$L_5$,$C_i$,$T_u$}to the CS through secure channel.

2) After receiving the message {$L_3$,$L_4$,$L_5$,$C_i$,$T_u$}, the CS verifies the freshness of $T_u$ by checking $|T_s - T_u| \le \Delta T$ ,where $T_s$ is receiving time of the request message. If the verification holds, CS extracts the corresponding $B_i$ and $SK_{uc\text{-}cs}$ from database based on $C_i$ and calculates

$$\text{ID}^*_i = L_4 \oplus \text{h}(C_i \,||\, h(x)),$$

$$N^*_2 = L_5 \oplus \text{h}(\text{ID}^*_i \,||\, \text{SK}_{uc\text{-}cs}),$$

$$L^*_3 = \text{h}(\text{ID}^*_i \,||\, SK_{uc\text{-}cs} \,||\, B_i \,||\, N^*_2 \,||\, T_u).$$

Comparing $L^*_3$ with $L_3$ sent from $U_i$ to see if they are equal. If they are equal, the CS successfully authenticates user $U_i$. Then, the CS calculates

$$S_2 = N_2^* \cdot \mathrm{P},$$

$$S_3 = E_{SK_{bs\text{-}cs}}(ID_i, SK_{uc\text{-}cs}, S_2, N_2, B_i),$$

and sends message $\{S_3, T_s\}$ to fog server $FS_j$.

3) $FS_j$ receives the message $\{S_3, T_s\}$ at time $T_f$ and verifies if $|T_f - T_s| \le \triangle T$. If the verification holds, $FS_j$ computes $D_{SK_{bs\text{-}cs}}(S_3) = (ID_i, SK_{uc\text{-}cs}, S_2, N_2^*, B_i)$. Then, $FS_j$ generates a random number $N_3$ and computes

$$M_1 = SK_{uc-cs} \cdot P_{ub} = (M_x, M_y),$$

$$M_2 = h(ID_i || M_x || T_f),$$

$$M_3 = N_3 \oplus M_1,$$

$$M_4 = N_3 \cdot P,$$

$$ACK = N_3 \cdot S_2,$$

$$SK = h(ACK || B_i || N_2^* || N_3).$$

$FS_j$ sends message $\{M_2, M_3, M_4, T_f\}$ to user $U_i$ over public channel.

4) After receiving message $\{M_2, M_3, M_4, T_f\}$, user $U_i$ checks the freshness of the message by verifying $|T_u^* - T_f| \le \triangle T$, where $T_u^*$ is current timestamp. If the verification holds, $U_i$ calculates

$$M_1^* = SK_{uc-cs}^* P_{ub} = (M_x^*, M_y^*),$$

$$M_2^* = h(ID_i || M_x^* || T_f),$$

and checks whether $M_2^* \stackrel{?}{=} M_2$ or not. If equal, $U_i$ successfully authenticates $FS_j$. Then, $U_i$ computes

$$N_3^* = M_3 \oplus M_1^*,$$

$$ACK^* = N_2 \cdot M_4,$$

$$SK^* = h(ACK || B_i || N_2 || N_3^*),$$

$$M_5 = h(ID_i || B_i || SK^* || T_u^*),$$

and sends $\{M_5, T_u^*\}$ to $FS_j$ to complete mutual authentication.

5) After receiving message $\{M_5, T_u^*\}$, $FS_j$ checks the authenticity of the user by computing $M_5^* \stackrel{?}{=} h(ID_i || B_i || SK || T_u^*)$. If the condition fails, $FS_j$ terminates the session. Otherwise, $FS_j$ stores the session key SK and sends information to the user $U_i$.

# 3. Shortcomings of Sahoo et al.'s Scheme

In this section, the drawbacks of Sahoo et al.'s scheme [16] are demonstrated, the following subsections provide a detailed description of drawbacks of Sahoo et al.'s scheme.

## 3.1. Shortcoming 1

The data distribution phase of Sahoo et al.'s scheme is not well explained in many places. In the data distribution phase, it is mentioned that the SC sends message $\{L_3, L_4, L_5, C_i, T_u\}$ to the cloud server CS via secure channel and fog server $FS_j$ sends message $\{M_2, M_3, M_4, T_f\}$ to user $U_i$ via public channel. However, it is not clarified whether CS sends message $\{S_3, T_s\}$ and $U_i$ sends message $\{M_5, T_u^*\}$ via public channel or secure channel. Secondly, SC sends message $\{L_3, L_4, L_5, C_i, T_u\}$ to cloud server CS over secure channel is not in accordance with the general practice. In general, the process of user and fog server authenticating with each other and agreeing on a session key is performed over public channel.

## 3.2. Shortcoming 2

User $U_k$ is a registered legitimate user and the cloud server CS saves $U_k$'s value $\{B_k, C_k, SK_{uc-cs}^k\}$. If $U_k$ knows his friend $U_i$'s identity $ID_i$, then $U_k$ can impersonate user $U_i$ by following steps:

1) User $U_k$ chooses a random number $N_k$ and current timestamp $T_k$ and by $U_k$'s value of $\{B_k, C_k, SK_{uc-cs}^k\}$ to compute

$$L_{3k} = h(ID_i || SK_{uc\text{-}cs}^k || B_k || N_k || T_k),$$

$$L_{4k} = ID_i \oplus h(C_k || h(x)),$$

$$L_{5k} = N_k \oplus h(ID_i || SK_{uc\text{-}cs}^k).$$

Then, $U_k$ sends the message $\{L_{3k}, L_{4k}, L_{5k}, C_k, T_k\}$ to the cloud server CS over a secure channel.

2) After receiving message $\{L_{3k}, L_{4k}, L_{5k}, C_k, T_k\}$, CS retrieves the corresponding $\{B_k, SK_{uc-cs}^k\}$ by $C_k$ and calculates

$$ID_i^* = L_{4k} \oplus h(C_k || h(x)),$$

$$N_k^* = L_{5k} \oplus h(ID_i^* || SK_{uc\text{-}cs}^k),$$

$$L_{3k}^* = h(ID_i^* || SK_{uc\text{-}cs}^k || B_k || N_k^* || T_k).$$

CS obtains $U_i$'s identity $ID_i$ by $L_{4k}$ and compares $L_{3k}^*$ with $L_{3k}$ sent by $U_k$. Obviously, the verification holds and CS thinks that it is user $U_i$ who sends the message, then $U_k$ impersonates user $U_i$ successfully. Finally, CS computes

$$S_{2k} = N_k \cdot P,$$

$$S_{3k} = E_{SK_{bs-cs'}}(ID_i, SK_{uc\text{-}cs}^k, S_{2k}, N_k, B_k),$$

and sends message $\{S_{3k}, T_{sk}\}$ to fog server $FS_g$.

3) Upon receving message $\{S_{3k}, T_{sk}\}$, $FS_g$ decrypts $S_{3k}$ to obtain the values of $\{ID_i, SK_{uc\text{-}cs}^k, S_{2k}, N_k, B_k\}$ by $SK_{bs\text{-}cs'}$. Then, $FS_g$ generates a random number $N_3$ and calculates

$$M_1 = SK_{uc-cs}^k \cdot P_{ub} = (M_x, M_y),$$

$$M_3 = N_3 \oplus M_1,$$

$$M_4 = N_3 \cdot P,$$

$$ACK = N_3 \cdot S_{2k},$$

$$SK = h(ACK || B_k || N_k^* || N_3).$$

Finally, $FS_g$ sends message $\{M_3, M_4, T_g\}$ to user $U_k$ via public channel.

4) After receiving $\{M_3, M_4, T_g\}$, $U_k$ computes

$$M_1^* = SK_{uc-cs}^k \cdot P_{ub} = (M_x^*, M_y^*),$$

$$N_3^* = M_3 \oplus \mathrm{M}_1^*,$$

$$ACK = N_k \cdot M_4,$$

$$SK^* = h(ACK \,||\, \mathrm{B_k} \,||\, N_k \,||\, N_3^*) .$$

Thus, $U_k$ impersonates $U_i$ and agrees on a session key with fog server $FS_g$.

### 3.3. Shortcoming 3

After user $U_i$ and fog server $FS_j$ complete a mutual authentication process, fog server $FS_j$ can obtain user $U_i$'s $ID_i$, $SK_{uc\text{-}cs}$ and $B_i$ by decrypting $S_3$. Then, for the next mutual authentication, when users $U_a$, $U_b$ and $U_i$ contact other fog servers, users $U_a$, $U_b$ and $U_i$ respectively send messages $\{L'_{3a}, L'_{4a}, C'_{ia}, T'_{ua}\}$, $\{L'_{3b}, L'_{4b}, C'_{ib}, T'_{ub}\}$, $\{L'_3, L'_4, C'_i, T'_u\}$ to cloud server CS. Upon receiving the messages, CS authenticates the identities of $U_a$, $U_b$ and $U_i$. After calculation, CS sends $\{S'_{3a}, T'_{sa}\}$, $\{S'_{3b}, T'_{sb}\}$, $\{S'_3, T'_s\}$ to corresponding fog servers.

After receiving message and computing, the corresponding fog servers send messages $\{M'_{2a}, M'_{3a}, M'_{4a}, T'_{fa}\}$, $\{M'_{2b}, M'_{3b}, M'_{4b}, T'_{fb}\}$, $\{M'_2, M'_3, M'_4, T'_f\}$ to users $U_a$, $U_b$ and $U_i$, respectively. $FS_j$ then intercepts messages $\{M'_{2a}, T'_{fa}\}$, $\{M'_{2b}, T'_{fb}\}$, $\{M'_2, T'_f\}$ to compute

$$\mathrm{M}_1 = \mathrm{SK}_{\mathrm{uc-cs}} \cdot \mathrm{P}_{\mathrm{ub}} = (\mathrm{M}_x, \mathrm{M}_y),$$

$$M_{2a}^* = \mathrm{h}(\mathrm{ID}_i || \mathrm{M}_x || \mathrm{T}'_{fa}),$$

$$M_{2b}^* = \mathrm{h}(\mathrm{ID}_i || \mathrm{M}_x || \mathrm{T}'_{fb}),$$

$$M_2^* = \mathrm{h}(\mathrm{ID}_i || \mathrm{M}_x || \mathrm{T}'_f),$$

$FS_j$ verifies $M_{2a}^* \overset{?}{=} M'_{2a}$, $M_{2b}^* \overset{?}{=} M'_{2b}$, $M_2^* \overset{?}{=} M'_2$. Obviously, $M_2^* = M'_2$. Thus, $FS_j$ can track user $U_i$ through $M_2$.

### 3.4. Shortcoming 4

1) According to subsection 3.3 above, user $U_i$ can be traced. When it is confirmed that the user who sends the message is $U_i$, fog server $FS_j$ intercepts the message $\{M'_2, M'_3, M'_4, T'_f\}$ sent by fog server $FS_g$ to $U_i$ and chooses a random number $N_j$ to calculate

$$\mathrm{M}_1 = \mathrm{SK}_{\mathrm{uc-cs}} \cdot \mathrm{P}_{\mathrm{ub}} = (\mathrm{M}_x, \mathrm{M}_y),$$

$$M_2^* = \mathrm{h}(\mathrm{ID}_i || \mathrm{M}_x || \mathrm{T}_f),$$

$$M_3^* = N_j \oplus \mathrm{M}_1,$$

$$M_4^* = N_j \cdot \mathrm{P} .$$

$FS_j$ sends message $\{M_2^*, M_3^*, M_4^*, T'_f\}$ to user $U_i$.

2) After receiving message $\{M_2^*, M_3^*, M_4^*, T'_f\}$, $U_i$ computes

$$\mathrm{M}_1^* = \mathrm{SK}_{\mathrm{uc-cs}}^* \cdot \mathrm{P}_{\mathrm{ub}} = (\mathrm{M}_x^*, \mathrm{M}_y^*),$$

$$M_2^{**} = \mathrm{h}(\mathrm{ID}_i || \mathrm{M}_x^* || \mathrm{T}_f),$$

$U_i$ checks whether $M_2^{**}$ is equal to the passed $M_2^*$. Obviously, $\mathrm{M}_2^{**} = \mathrm{M}_2^*$, $U_i$ thinks that it is fog server $FS_g$ that sends the message. Then, $U_i$ calculates

$$N_j^* = M_3 \oplus \mathrm{M}_1^*,$$

$$ACK = N_2 \cdot M_4,$$

$$SK^* = h(ACK \,||\, \mathrm{B}_i \,||\, N_2 \,||\, N_j^*),$$

$$M_5 = \mathrm{h}(\mathrm{ID}_i \,||\, \mathrm{B}_i \,||\, SK^* \,||\, T_u^*),$$

and sends message $\{M_5, T_u^*\}$ to fog server $FS_g$.

3) After receiving $\{M_5, T_u^*\}$, $FS_g$ verifies $M_5^* \overset{?}{=} h(\mathrm{ID}_i || \mathrm{B}_i || SK || \mathrm{T}_u^*)$, the verification cannot pass, the session key $SK^*$ calculated by $U_i$ is different from the session key SK of fog server $FS_g$, thus $U_i$ and $FS_g$ cannot agree on a same session key, and it also leads to a waste of network resources.

### 3.5. Shortcoming 5

1) The attacker intercepts message $\{S_3, T_s\}$ sent from cloud server CS to fog server $FS_j$. After a period of time, the attacker selects the current timestamp $T_g$ and sends message $\{S_3, T_g\}$ to fog server $FS_j$.

2) Upon receiving message $\{S_3, T_g\}$, $FS_j$ checks $|T_j - T_g| \leq \Delta T$, where timestamp $T_g$ is valid. $FS_j$ then decrypts $S_3$ using shared key $SK_{bs\text{-}cs}$ to obtain $\{ID_i, SK_{uc\text{-}cs}, S_2, N_2^*, B_i\}$ and selects a random number $N_f$ to calculate

$$\mathrm{M}_1 = \mathrm{SK}_{\mathrm{uc-cs}} \cdot \mathrm{P}_{\mathrm{ub}} = (\mathrm{M}_x, \mathrm{M}_y),$$

$$M_2 = \mathrm{h}(\mathrm{ID}_i || \mathrm{M}_x || \mathrm{T}_j),$$

$$M_3 = N_3 \oplus \mathrm{M}_1,$$

$$M_4 = N_f \cdot \mathrm{P}.$$

As a result, replay attack is successfully performed and also wastes network resources of fog server $FS_j$.

## 4. Improved Scheme

To overcome Sahoo et al.'s scheme [16]'s security problems, an improved scheme is proposed in this paper. The improved scheme includes three participants: user ($U_i$), cloud server (CS) and fog server ($FS_j$). User and fog server authenticate with each other and agree on a session key with the help of cloud server. The following is a detailed description of the improved scheme.

### 4.1. Initialization Phase

(1) Cloud server CS chooses a base point P on the elliptic curve E(GF($p$)) and the order of P is $n$. Then, CS chooses master keys $x$, $y$ and computes the public key $P_{ub} = y \cdot P$.

(2) CS selects two hash functions:

$$\mathrm{h}(\cdot): \{0,1\}^* \longrightarrow \{0,1\}^k$$

$$\mathrm{h}_1(\cdot): \{0,1\}^* \longrightarrow Z_p^*$$

and keeps {x, y} as private key, making $\{P, P_{ub}, h(\cdot), h_1(\cdot)\}$ as public.

### 4.2. User Registration Phase

User $U_i$ registers with the cloud server CS by the following steps:

1) $U_i$ selects his identity $ID_i$, password $PW_i$ and extracts his biometric $BM_i$. Using fuzzy extractor, $U_i$ calculates $\mathrm{Gen}(BM_i)=(\omega_i,\theta_i)$, $RPW_i = h(PW_i \| \omega_i)$ . Then, user sends $\{ID_i,RPW_i\}$ to cloud server CS via a secure channel.

2) After receiving message $\{ID_i,RPW_i\}$, the cloud server CS generates a pseudo- identity $TID_i$ for user $U_i$ and calculates

$$A_i = h(ID_i \| x),$$

$$B_i = h(ID_i \| RPW_i \| A_i),$$

$$C_i = h(ID_i \| RPW_i \| SK_{uc\text{-}cs}).$$

Then, CS stores $\{TID_i,B_i,C_i,SK_{uc\text{-}cs},h(\cdot)\}$ into a smart card(SC) and sends the SC to user $U_i$, meanwhile CS stores $\{TID_i, ID_i, B_i, SK_{uc\text{-}cs}\}$ into its database.

3) After receiving the SC, user calculates

$$SK'_{uc\text{-}cs}=SK_{uc\text{-}cs} \oplus h(RPW_i),$$

$$E_i = B_i \oplus h(ID_i \| RPW_i).$$

user $U_i$ then replaces $B_i$ and $SK_{uc\text{-}cs}$ in the SC with $E_i, SK'_{uc\text{-}cs}$ . At this point, the SC contains $\{ TID_i, C_i, E_i, SK'_{uc\text{-}cs}, \theta_i, \mathrm{Rep}(\cdot), h(\cdot) \}$. The process of user registration is shown in Figure 1.

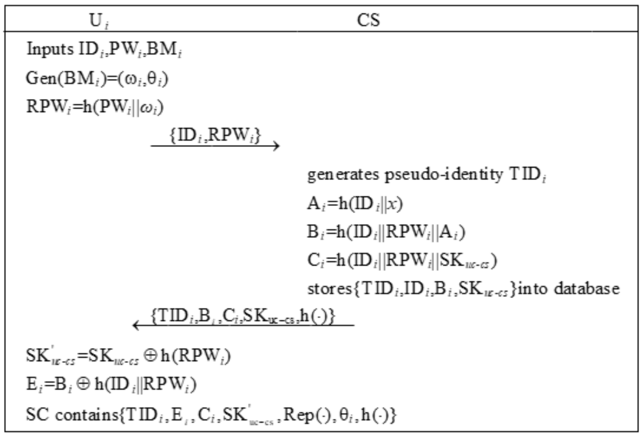


**Figure 1.** User registration phase

### 4.3. Fog Server Registration Phase

Fog server registers with the cloud server through the following steps:

1) Fog server $FS_j$ selects its own identity $BID_j$ and sends $BID_j$ to cloud server CS via a secure channel.

2) After receiving $BID_j$, CS selects a random number $r_f$ and calculates

$$R_1=h(BID_j \| y \| SK_{bs\text{-}cs} \| r_f),$$

$$R_2=R_1 \oplus h(y \| r_f).$$

CS then saves $\{BID_j,r_f,SK_{bs\text{-}cs}\}$ into its own database and sends $\{R_1,R_2,SK_{bs\text{-}cs}\}$ to fog server $FS_j$.

3) After receiving message $\{R_1, R_2, SK_{bs\text{-}cs}\}$, $FS_j$ saves the message into its own database. The fog server registration process is shown in Figure 2

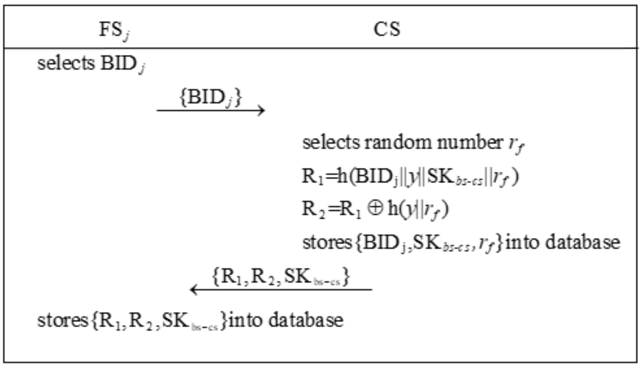


**Figure 2.** Fog server registration phase

### 4.4. Login and Authentication Phase

In this phase, if user wants to access fog servers via public channel, cloud server can help user and fog server to authenticate each other and agree on session key. The steps are demonstrated as follows; the detailed process is shown in Figure 3.

1) User inserts his SC into card reader and inputs his identity $ID_i$, password $PW_i$, and biometrics $BM_i$, SC retrieves $\omega_i^*=\mathrm{Rep}(BM_i^*,\theta_i)$ and calculates

$$RPW_i^* = h(PW_i^* \| \omega_i^*),$$

$$SK_{uc\text{-}cs}^* = SK'_{uc\text{-}cs} \oplus h(RPW_i^*),$$

$$C_i^* = h(ID_i^* \| RPW_i^* \| SK_{uc\text{-}cs}^*).$$

SC verifies whether $C_i^* \overset{?}{=} C_i$ holds or not. If the verification holds, it indicates that the user is legitimate holder of SC. The SC then generates a random number $r_u$ and current timestamp $T_1$ to calculate

$$B_i^*=E_i \oplus h(ID_i \| RPW_i),$$

$$F_1 = r_u \oplus h(ID_i \| SK_{uc\text{-}cs} \| T_1),$$

$$F_2=BID_j \oplus h(r_u \| B_i^*),$$

$$F_3=h(BID_j \| SK_{uc\text{-}cs} \| B_i^* \| r_u \| T_1).$$

Finally, SC sends $\{TID_i, F_1, F_2, F_3, T_1\}$ to the cloud server CS via public channel.

2) After receiving message $\{TID_i, F_1, F_2, F_3, T_1\}$, CS first validates $|T_2 - T_1| \le \Delta T$ , where $T_2$ is current time of CS receiving the message. If the validation holds, CS searches for $\{ID_i,B_i,SK_{uc\text{-}cs}\}$ according to $TID_i$ and calculates

$$r_u^*=F_1 \oplus h(ID_i \| SK_{uc\text{-}cs} \| T_1),$$

$$BID_j^*=F_2 \oplus h(r_u^* \| B_i),$$

$$F_3^*=h(BID_j^* \| SK_{uc\text{-}cs} \| B_i \| r_u^* \| T_1).$$

CS checks wehther $F_3^* \overset{?}{=} F_3$ , if the condition fails, CS rejects the session. Otherwise, cloud server CS authenticates the identity of the user $U_i$. Then, CS retrieves $\{r_f,SK_{bs\text{-}cs}\}$ by $BID_j$ and calculates

$$R_c = r_u \cdot P,$$

$$R_1=h(BID_j \| y \| SK_{bs\text{-}cs} \| r_f),$$

$$G_1 = \mathrm{h}(\mathrm{ID}_i \| R_I \| R_c \| \mathrm{T}_2),$$

$$\mathrm{G}_2 = \mathrm{E}_{\mathrm{SK}_{bs\text{-}cs}}(\mathrm{ID}_i, \mathrm{h}_1(\mathrm{SK}_{uc\text{-}cs} \| r_u), R_c, \mathrm{G}_1),$$

$$\mathrm{TID}_i^* = \mathrm{TID}_i \oplus r_u.$$

CS replaces $\mathrm{TID}_i$ in its own database with $\mathrm{TID}_i^*$ and sends $\{\mathrm{G}_2, \mathrm{T}_2\}$ to $\mathrm{FS}_j$.

3) $\mathrm{FS}_j$ receives the message from CS at time $\mathrm{T}_3$ and verifies $|\mathrm{T}_3 - \mathrm{T}_2| \le \Delta\mathrm{T}$. If the verification holds, $\mathrm{FS}_j$ decrypts $\mathrm{G}_2$ by $\mathrm{SK}_{bs\text{-}cs}$ to obtain $\{\mathrm{ID}_i, \mathrm{h}_1(\mathrm{SK}_{uc\text{-}cs} \| r_u), \mathrm{R}_c, \mathrm{G}_1\}$, and computes $\mathrm{G}_1^* = \mathrm{h}(\mathrm{ID}_i \| \mathrm{R}_1 \| R_c \| \mathrm{T}_2)$. $\mathrm{FS}_j$ checks whether $\mathrm{G}_1^* \stackrel{?}{=} \mathrm{G}_1$ or not. If equal, fog server $\mathrm{FS}_j$ authenticates the cloud server CS. Then, $\mathrm{FS}_j$ generates a random number $r_j$ and computes

$$M_I = \mathrm{h}_1(\mathrm{SK}_{uc\text{-}cs} \| r_u) \cdot \mathrm{P}_{\mathrm{ub}} = (\mathrm{M}_x, \mathrm{M}_y),$$

$$\mathrm{P}_f = r_j \cdot \mathrm{P},$$

$$ACK = r_j \cdot R_c,$$

$$SK = \mathrm{h}(ACK \| \mathrm{ID}_i \| \mathrm{BID}_j),$$

$$N_j = \mathrm{h}(SK \| \mathrm{P}_f \| \mathrm{M}_x \| \mathrm{T}_3).$$

Now, $\mathrm{FS}_j$ sends the message $\{\mathrm{P}_f, N_j, \mathrm{T}_3\}$ to $\mathrm{U}_i$.

4) $\mathrm{U}_i$ receives the message and checks the freshness of the message by verifying $|\mathrm{T}_4 - \mathrm{T}_3| \le \Delta\mathrm{T}$. If the verification holds, $\mathrm{U}_i$ computes

$$\mathrm{M}_1^* = \mathrm{h}_1(\mathrm{SK}_{uc\text{-}cs} \| r_u) \cdot \mathrm{P}_{\mathrm{ub}} = (\mathrm{M}_x^*, \mathrm{M}_y^*),$$

$$\mathrm{ACK}^* = r_u \cdot \mathrm{P}_f,$$

$$\mathrm{SK}^* = \mathrm{h}(\mathrm{ACK}^* \| \mathrm{ID}_i \| \mathrm{BID}_j),$$

$$N_j^* = \mathrm{h}(SK^* \| \mathrm{P}_f \| \mathrm{M}_x^* \| \mathrm{T}_3).$$

$\mathrm{U}_i$ checks whether $\mathrm{N}_j^* \stackrel{?}{=} \mathrm{N}_j$ or not. If equal, then user $\mathrm{U}_i$ authenticates fog server $\mathrm{FS}_j$ and $\mathrm{U}_i$ saves the session key SK for future communication. Finally, $\mathrm{U}_i$ calculates $\mathrm{TID}_i^* = \mathrm{TID}_i \oplus r_u$ and replaces $\mathrm{TID}_i$ in the SC with $\mathrm{TID}_i^*$.

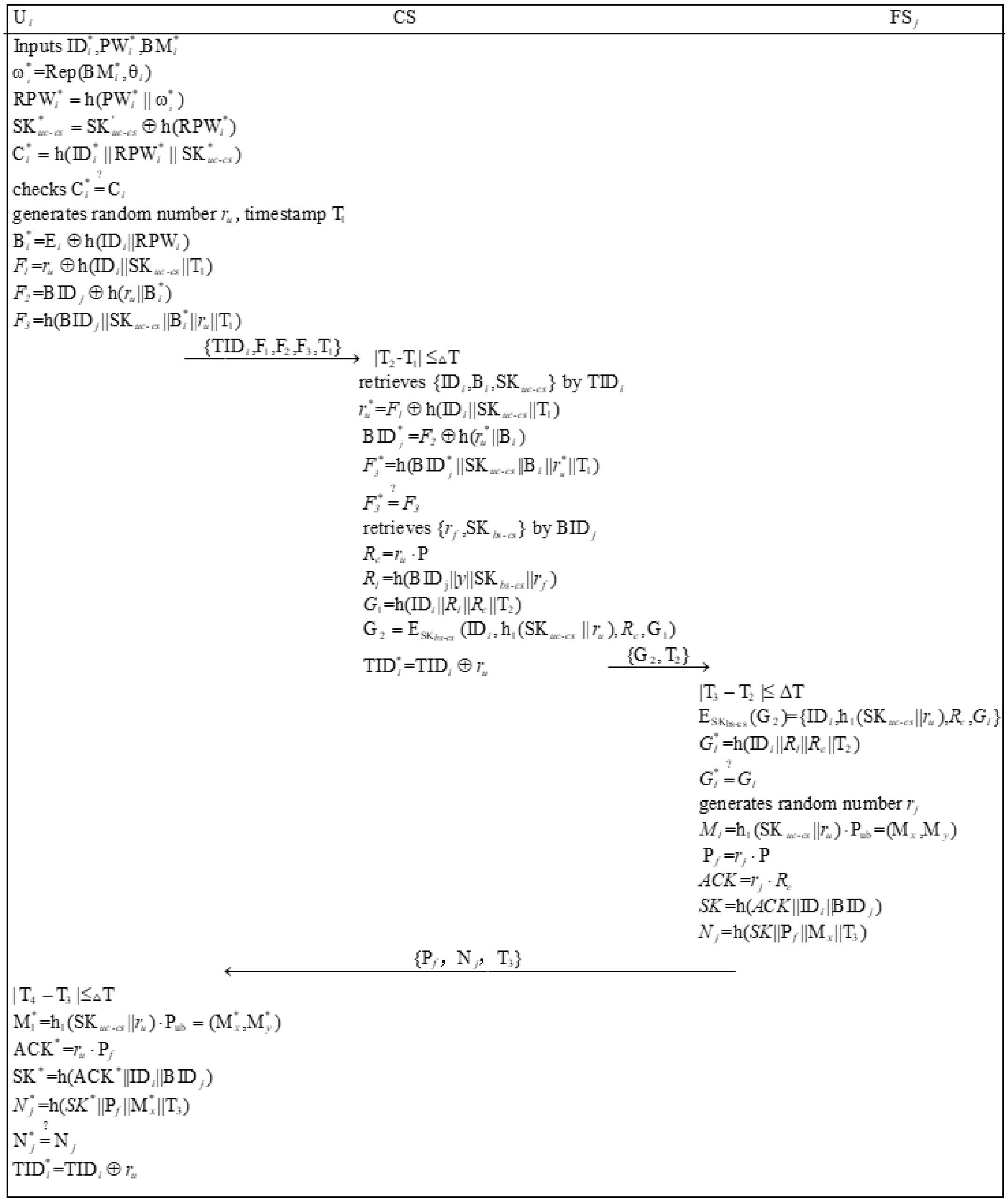


**Figure 3.** Login and authentication phase

# 5. Security Analysis of the Improved Scheme

## 5.1. Providing User Anonymity

In the login and authentication phase of the improved scheme, user does not send his identity directly on the public channel, but sends a pseudo-identity $TID_i$, the attacker cannot correspond to the user's identity $ID_i$ through the intercepted pseudo- identity $TID_i$. Therefore, the improved scheme can provide user anonymity.

## 5.2. Resist Tracking Attack

User hides his identity by sending a pseudo-identity $TID_i$ and the user's pseudo- identity $TID_i$ is updated by $TID_i^* = TID_i \oplus r_u$ at the end of each session. Since $TID_i^* = TID_i \oplus r_u$ contains a random number $r_u$, user's pseudo-identity is different each time. Therefore, an attacker cannot correspond to the user's identity through the intercepted pseudo-identity $TID_i$. Furthermore, the request message $\{TID_i, F_1, F_2, F_3, T_1\}$ sent by the user $U_i$ contains a random number $r_u$ or a timestamp $T_1$ in $F_1$, $F_2$, $F_3$. Therefore, when the user $U_i$ contacts the fog server again, the values of $F_1$, $F_2$, $F_3$ will be different and the attacker will not be able to trace user through the request message.

## 5.3. Resist User Impersonation Attack

If an attacker tries to impersonate user $U_i$ to cheat the cloud server, then the attacker must construct a correct request message, but according to the analysis in 5.1, the attacker cannot know $U_i$'s pseudo-identity $TID_i$. Secondly, attacker also does not know user's secret value $B_i$ and $SK_{uc\text{-}cs}$, so the attacker cannot construct a valid $F_3$ to pass the authentication of the cloud server. Even if fog server $FS_j$ acts as an attacker, $FS_j$ can only obtain $h_1(SK_{uc\text{-}cs}||r_u)$ by finishing one authentication, and $h_1(SK_{uc\text{-}cs}||r_u)$ is different for each session. Therefore, $FS_j$ is also unable to calculate valid $F_1$, $F_2$, $F_3$ without knowing $B_i$ and $SK_{uc\text{-}cs}$. So, the improved scheme can resist user impersonation attack.

## 5.4. Resist Replay Attacks

An attacker may attempt to perform replay attack by resending an intercepted message to the cloud server after a period of time. However, the improved scheme uses timestamp verification and the past message contains a timestamp that is not in the acceptable range, so resending the past message will result in session termination. Even though the attacker can modify timestamp $T_1$, the attacker does not know $B_i$ and $SK_{uc\text{-}cs}$, then the corresponding $F_1$, $F_2$ and $F_3$ cannot be modified according to the new timestamp. Therefore, the value of $F_3$ sent by the attacker is different from $F_3$ calculated by the cloud server CS and the verification cannot pass. Therefore, the improved scheme is resistant to replay attack.

# 6. Comparison of the Improved Scheme

In this section, the improved scheme is compared with Sahoo et al.'s scheme [16] and two other schemes [11][17] in terms of security features and computational cost. In order to achieve reliable and straightforward evaluation, Table 3 shows the symbols and execution times required for various operations, and these enable us to accurately compare the improved scheme with three similar direction schemes in the computational costs. The comparison results about safety features and computational costs are shown in Tables 2 and 4, respectively.

**Table 2.** Comparison of safety features

| Schemes | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ | $F_8$ | $F_9$ | $F_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| Chen et al.'s scheme | × | √ | √ | √ | √ | √ | √ | √ | × | √ |
| Ma et al.'s scheme [17] | √ | √ | × | × | √ | √ | √ | √ | √ | √ |
| Sahoo et al.'s scheme[16] | × | √ | × | × | √ | √ | √ | √ | √ | √ |
| improved scheme | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |

$F_1$:user anonymity; $F_2$: replay attacks; $F_3$: user untrace ability; $F_4$: impersonation attacks; $F_5$: reach session key; $F_6$: mutual authentication; $F_7$: offline password guessing attacks; $F_8$: denial of service attacks; $F_9$: perfect forward security; $F_{10}$: man-in-the-middle attacks;

**Table 3.** Symbols represented by various operations and approximately required time

| Symbols | Description | ≈Calculation time (ms) |
|---|---|---|
| H | Hash function | 0.0023 |
| X | XOR operation | 0.001 |
| F | Fuzzy extractor function | 2.226 |
| E | Symmetric key encryption/decryption | 0.0046 |
| P | ECC point multiplication | 2.226 |

**Table 4.** Comparison of calculation costs

| scheme | P1 | P2 | P3 | P4 |
|---|---|---|---|---|
| Chen et al.'s scheme | 8H+4X+F | 2H+F | 22H+24X+5P | 32H+28X+5P+2F |
| Ma et al.'s scheme [17] | 2H+3P | ---- | 17H+4X+15P | 19H+4X+18P |
| Sahoo et al.'s scheme[16] | 8H+4X+F | 2H+X+F | 14H+8X+2E+6P | 24H+13X+2F+2E+6P |
| improved scheme | 8H+3X+F | 3H+X+F | 13H+7x+2E+5P | 24H+11X+2F+2E+5P |

P1: registration phase; P2: login phase; P3: authentication and key agreement phase; P4: total computational cost;

From table 2, it is shown that the security of the improved scheme is better than Sahoo et al.'s scheme [16] and other schemes [11][17] and the improved scheme can resist various known attacks. From table 4, the total computational cost of Sahoo et al.'s scheme [16], Chen et al.'s scheme [11], Ma et al.'s scheme [17] and the improved scheme are 17.8854, 15.6836, 40.1157, 15.6574, respectively. The computational cost of the improved scheme is lower than the other three schemes.

# 7. Conclusion

In this paper, we analyse Sahoo et al.'s scheme and find that many areas of their scheme are not clearly explained in the data distribution phase and it also cannot resist user impersonation attack, tracking attack, denial of service attack and replay attack. In order to overcome the shortcomings of Sahoo et al.'s scheme, this paper makes full use of random numbers and timestamps to propose an improved scheme, we compare the improved scheme with existing similar schemes in terms of security features and computational cost and result shows that the improved scheme has higher security, can resist various known attacks and has lightweight features with lower computational cost. The improved scheme is therefore more suitable for applications in fog computing environments.

# References


[1] Yang Xiaodong, An Faying, Yang Ping, et al. Cross-domain identity authentication scheme based on proxy re-signature in cloud environment [J]. Journal of Computers,2019,42(4):756-771. (in Chinese)

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

[3] Alrawais A, Alhothaily A, Hu CQ, Cheng XZ. Fog computing for the Internet of Things: Security and privacy issues[J], IEEE Internet Computing, 2017, 21(2): 34–42.

[4] 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.

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

[6] 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.

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

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

[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] 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: 1-14.

[11] Chen, C.M.; Huang, Y.; Wang, K.H.; Kumari, S.; Wu, M.E.A secure authenticated and key exchange scheme for fog computing[J]. Enterprise Information Systems, 2020;15(1):1-16.

[12] Rana M, Mahmood K, Saleem M A, et al. Towards a Provably Secure Authentication Protocol for Fog-Driven IoT-Based Systems[J]. Applied Sciences, 2023, 13(3): 1424.

[13] Guo Yimin, Zhang Zhenfeng, Xiong Ping, et al. Lightweight fog assisted iot Authentication Protocol based on PUF [J]. Chinese Journal of Computers,2022,45(7):1412-1430. (in Chinese).

[14] Minahil Rana, Khalid Mahmood, Muhammad Asad Saleem, et al. Towards a Provably Secure Authentication Protocol for Fog-Driven IoT-Based Systems[J]. applied sciences,2023, 13(3):1424

[15] Yijian Lin, Xiaoming Wang, Qingqing Gan, and Mengting Yao. A secure cross-domain authentication scheme with perfect forward security and complete anonymity in fog computing[J]. Journal of Information Security and Applications, 2021; 63 (9): 103022.

[16] Sahoo S S, Mohanty S , Majhi B. An efficient three-factor user authentication scheme for industrial wireless sensor network with fog computing[J]. International journal of communication systems, 2022(3):35.

[17] Ma Mimi, He Debiao, Wang Huaqun, Kumar Neeraj, Choo Kim-Kwang Raymond. 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.