Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 4, No. 2, 2023

Analysis and Improvement on a Three-Factor Authentication Scheme in IoT Environment

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

[1] School of Software, Tiangong University, Tianjin 300387, China
[2] School of Computer Science and Technology, Tiangong University, Tianjin 300387, China
*** Corresponding author:** Anqian Li (Email: lianqian98@163.com)

Abstract: With the development of IoT technology, more and more devices are connected to the Internet, which brings great convenience to people, but also security risks. As a result, IoT authentication scheme has become a research hotspot. In 2020, Lee et al. proposed a three-factor anonymous authentication scheme in IoT environment and claimed that their scheme can resist many known attacks. However, we find that their scheme not only has some drawbacks, but also has difficulty in resisting man-in-the-middle attack and impersonation attack. To overcome these drawbacks, we propose an improved scheme. Through security analysis and computational cost comparison, it is shown that the improved scheme is not only resistant to existing known attacks, but also has a smaller overhead in terms of computational cost and is suitable for resource-constrained IoT environment.

Keywords: Internet of Things (IoT); Authentication; Key Agreement.

1. Introduction

With the rapid development of sensing and communication technologies, the Internet of Things (IoT) connects real-world objects to the Internet. IoT is a collection of physical devices such as embedded sensors and actuators with computing power that can communicate between devices and systems0. IoT enables seamless communication and automatic management between heterogeneous devices, and can bring significant benefits to social production and human life through a fully intelligent and automated remote management system[2].

However, from a security perspective, the vast majority of IoT devices are exposed to an almost uncontrollable environment. At the same time, the proliferation of smart devices places their connection to the Internet at potential risk, such as unauthorized node tampering or impersonation attack[3]. To ensure the security of the IoT, people have been exploring authentication schemes and carrying out related research. Authentication is not only a key requirement for the development and progress of IoT, but also the "first line of defense" for security protection. Inadequate authentication or the existence of security vulnerabilities will lead to a security crisis for the entire IoT.

In recent years, scholars have done a lot of research on authentication schemes in IoT, but many existing schemes have security concerns[4]. In 2014, Chen et al.[4] proposed a smart-card-based password authentication and key negotiation scheme in order to solve the problem of remote user access. They claimed that their scheme is resistant to off-line password guessing attack even if an attacker extracts the information stored in smart card. In 2015, Jiang et al.[5] pointed out that the scheme of Chen et al.[4] is insecure and proposed an improved authentication scheme. In 2016, Das[6] found that Jiang et al.'s scheme[5] was not only vulnerable to internal privilege attack, but also failed to provide correct authentication during the login and authentication phase. Therefore, Das[6] presented a three-factor authentication scheme, especially the use of biometric-based password and smart card, which greatly enhanced the security of the scheme. In 2017, Dhillion et al.[7] proposed a lightweight multi-factor remote user authentication scheme in IoT environment and claimed that their scheme can resist a variety of known attacks such as man-in-the-middle attack, replay attack and so on. In 2018, Kumari et al.[8] presented an ECC-based authentication scheme for IoT and cloud servers. They claimed that their scheme is resistant to known attacks such as impersonation attack, offline password guessing attack, and meets various security objectives. In 2019, Gope et al.[9] proposed a lightweight and privacy-preserving two-factor authentication scheme. The scheme allows IoT devices to communicate anonymously with the server and effectively improves the security of the scheme by exploiting the inherent security properties of physical unclonable functions (PUF). Moreover, they claimed that their scheme is efficient and suitable for resource-constrained IoT devices. In 2020, Lee et al.[10] pointed out that Dhillion et al.'s scheme[7] fails to resist stolen mobile device attack, user impersonation attack, and has no provision for revocation. As a result, they proposed an improved three-factor user authentication scheme to address these security issues. However, we find that there are some security vulnerabilities in Lee et al.'s scheme[10]. On the one hand, Lee et al.'s scheme[10] has some drawbacks in the registration of user, registration of IoT node as well as login and authentication phase. On the other hand, Lee et al.'s scheme[10] cannot resist man-in-the-middle (MITM) attack and user impersonation attack.

Section II briefly reviews Lee et al.'s authentication scheme. The analysis of the shortcomings of the Lee et al.'s scheme is in Section III. In Section IV, we propose an improved scheme. The security of the improved scheme is analyzed in Section V. Section VI compares the improved scheme with similar schemes. Section VII concludes the paper.

2. Review of Lee et al.'s Scheme

This section briefly reviews Lee et al.'s scheme[10], which mainly consists of the following five phases: (1) registration of user; (2) registration of IoT node; (3) login and authentication phase; (4) password change phase; (5) revocation phase. Since the shortcomings of the scheme of

Lee et al.[10] are mainly focused on the first three phases, for the sake of brevity, this section reviews only the registration of user, the registration of IoT node, and the login and authentication phase. Table 1 denotes the notations and descriptions in Lee et al.'s scheme[10].

Table 1. Notations and descriptions used in Lee et al.'s scheme

Notations	Definition
MN_i,N_j,GW	The i^{th} mobile user, gateway and the j^{th} sensor node
ID_i, PW_i, BIO_i	Identity, password, biometrics of MN_i
ID_j	Identity of N_j
PID_i, NID_j	Pseudo-identity of MN_i, N_j
K_G	Private key only known to GW
$E_k(\cdot)$/$D_k(\cdot)$	Symmetric encryption/decryption algorithm with key
n_x,r_x	Random numbers
SK, SK_{ij} (= SK_{ji})	Session key agreed between MN_i and N_j
Gen(·)/Rep(·)	Fuzzy biometric generator/reproduction
T_x	Timestamps
ΔT	Maximum transmission delay
$\oplus$	Exclusive-OR operator
$\|\|$	Concatenation operator
$h(\cdot)$	Hush function
$H(\cdot)$	Bio-Hash function

2.1. Registration of User

In this phase, a user registers at the gateway by performing the following steps.

1) MN_i selects its own identity ID_i, password PW_i and enters the biometric BIO_i. MN_i's mobile device calculates

$$PWB_i = h(PW_i \| H(BIO_i)),$$
$$MID_i = h(ID_i \| H(BIO_i)),$$

and sends a registration message $\{ID_i, PWB_i, MID_i\}$ to GW over a secure channel.

2) GW chooses two random numbers r_D, r_{GU}, and computes

$$RID_i = E_{K_G}(ID_i),$$
$$PID_i = E_{K_G}(ID_i \| r_D),$$
$$x_i = h(ID_i \| PWB_i),$$
$$y_i = h(ID_i \| PWB_i \| r_{GU}) \oplus h(K_{GU} \| ID_i),$$

where K_{GU} is the secret value chosen by GW for MN_i. GW stores $\{RID_i, MID_i\}$ in its own database and sends $\{PID_i, x_i, y_i, r_{GU}\}$ to MN_i.

3) MN_i stores $\{PID_i, x_i, y_i, r_{GU}\}$ in its own mobile device.

2.2. Registration of IoT Node

The registration process of IoT node is as follows.

1) N_j chooses ID_j, r_j, and computes

$$MP_j = h(K_{GN} \| r_j \| ID_j),$$
$$MI_j = r_j \oplus h(ID_j \| K_{GN}),$$

where K_{GU} is the shared key between GW and N_j. N_j sends registration information $\{ID_j, MP_j, MI_j\}$ to GW over the public channel.

2) After receiving the registration information, GW discloses N_j's identity ID_j and computes $r_j^* = MI_j \oplus h(ID_j \| K_{GN})$, $MP_j^* = h(K_{GN} \| r_j^* \| ID_j)$. GW checks whether $MP_j^* = MP_j$ holds or not. If it fails, the current phase is terminated immediately; otherwise, GW computes

$$x_j = h(ID_j \| K_{GN}),$$
$$y_j = x_j \oplus MP_j^*,$$

and sends $\{y_j\}$ to N_j.

3) N_j receives the message from GW and stores $\{y_j\}$ in its own memory.

2.3. Login and Authentication Phase

In this stage, MN_i and N_j agree on a session key with the help of GW. The specific steps are as follows.

1) MN_i enters ID_i, PW_i and BIO_i. MN_i's mobile device calculates

$$PWB_i = h(PW_i \| H(BIO_i)),$$
$$x_i^* = h(ID_i \| PWB_i),$$

and checks the equation $x_i^* = x_i$. If the equation is invalid, the current phase will be terminated immediately. Otherwise, mobile device generates n_i, T_1, and computes

$$A_i = y_i \oplus h(ID_i \| PWB_i \| r_{GU}),$$
$$UN_i = h(A_i \| PID_i \| n_i),$$
$$UZ_i = n_i \oplus A_i.$$

Then, MN_i sends a message $M_1 = \{PID_i, UN_i, UZ_i, T_1\}$ to N_j.

2) N_j verifies whether $| T_{fresh} - T_1 | \le \Delta T$ is correct or not, and if not, it will immediately terminate the current phase. Otherwise, N_j generates a random number n_j and calculates

$$x_j = y_j \oplus h(K_{GN} \| r_j \| ID_j),$$
$$A_j = h(x_j) \oplus n_j,$$
$$B_j = h(x_j \| n_j).$$

N_j sends $M_2 = \{M_1, ID_j, A_j, B_j\}$ to GW.

3) GW firstly checks the correction of $| T_{fresh} - T_1 | \le \Delta T$. If not, GW will reject N_j's request. Otherwise, GW computes

$$x_j^* = h(ID_j \| K_{GN}),$$
$$n_j^* = h(x_j^*) \oplus A_j,$$
$$B_j^* = h(x_j^* \| n_j^*).$$

Then, GW verifies whether $B_j^* = B_j$ is true or not. If fails, the current session will be terminated. Otherwise, GW

decrypts PID_i using its private key K_G to obtain a pair of values $< ID_i, r_D >= D_{K_G}(PID_i)$, and computes

$$A_i^* = h(K_{GU} \| ID_i),$$
$$n_i^* = UZ_i \oplus A_i^*,$$
$$UN_i^* = h(A_i^* \| PID_i \| n_i^*),$$

GW checks whether $UN_i^* = UN_i$ holds or not. If it's not equal, GW will reject MN_i's request. Otherwise, GW generates r_D^{new}, and calculates

$$F_j = h(ID_i \| n_i^*),$$
$$G_j = F_j \oplus x_j^*,$$
$$R_{ij} = n_j^* \oplus n_i^*,$$
$$H_j = h(x_j^* \| n_j^* \| n_i^* \| F_j),$$
$$PID_i^{new} = E_{K_G}(ID_i, r_D^{new}).$$

GW sends $M_3 = \{PID_i^{new}, G_j, R_{ij}, H_j\}$ to N_j.

4) N_j computes

$$F_j^* = G_j \oplus x_j,$$
$$n_i^* = R_{ij} \oplus n_j,$$
$$H_j^* = h(x_j \| n_j \| n_i^* \| F_j^*),$$

and verifies whether $H_j^* = H_j$ is correct or not. If fails, N_j terminates the current session immediately. Otherwise, N_j selects m_j, T_2, and computes

$$L_j = m_j \oplus h(ID_j \| n_i^*),$$
$$SK_{ji} = h(F_j^* \| n_i^* \| m_j),$$
$$SV_j = h(SK_{ji} \| T_1 \| T_2).$$

Then, N_j sends $M_4 = \{PID_i^{new}, L_j, SV_j, T_2\}$ to MN_i.

5) MN_i checks whether $| T_{fresh} - T_2 | \leq \Delta T$. If it's not equal, MN_i will terminate current phase. Otherwise, MN_i computes

$$m_j^* = L_j \oplus h(ID_j \| n_i),$$
$$SK_{ij} = h(h(ID_i \| n_i) \| n_i \| m_j^*),$$
$$SV_i = h(SK_{ij} \| T_1 \| T_2).$$

MN_i verifies whether $SV_i = SV_j$ holds or not. If it holds, then MN_i and N_j have successfully agreed on the same session key.

3. Analysis of Lee et al.'s Scheme

This section provides a security analysis of Lee et al.'s[10] scheme and identifies the following drawbacks of their scheme.

3.1. Drawbacks

Lee et al.'s scheme [10] has some drawbacks in the registration of user, registration of IoT node and login and authentication phase.

3.1.1. Drawbacks of the User Registration Phase

1) In the user registration phase of the Lee et al.'s scheme [10], GW selects a secret value K_{GU} for MN_i. However, GW does not store K_{GU} in its own database, which will result in GW not being able to retrieve K_{GU} during the login and authentication phase, and thus not being able to authenticate MN_i by equation (1).

$$\begin{aligned} &A_i^* = h(K_{GU} \| ID_i) \\ &n_i^* = UZ_i \oplus A_i^* \\ &UN_i^* = h(A_i^* \| PID_i \| n_i^*) \\ &\text{Checks } UN_i^* = UN_i \end{aligned} \quad (1)$$

2) GW calculates RID_i, PID_i after receiving the registration information $\{ID_i, PWB_i, MID_i\}$ sent by MN_i.

$$RID_i = E_{K_G}(ID_i),$$
$$PID_i = E_{K_G}(ID_i \| r_D),$$

GW stores $\{RID_i, MID_i\}$ in its own database. During the login and authentication phase, the identity information transmitted by MN_i in the public channel is the pseudo-identity PID_i. GW computes $< ID_i, r_D >= D_{K_G}(PID_i)$ and authenticates the MN_i by equation (1), without using the stored values $\{RID_i, MID_i\}$.

Therefore, the second drawback is that the user information stored by GW in the registration phase is not fully utilized. On the one hand, storing $\{RID_i, MID_i\}$ takes up the GW's storage space. If too much user information is stored or stored for too long, it will put pressure on the storage resources of GW. On the other hand, there is a risk of information leakage. For example, attackers can obtain user information by attacking GW, which can lead to security problems.

3.1.2. Drawbacks of the IoT Node Registration Phase

In the registration phase of the IoT node, GW calculates a secret value for N_j

$$x_j = h(ID_j \| K_{GN}).$$

During the login and authentication phase, GW checks whether N_j is a registered IoT node according to equation (2), where x_j is the key value to authenticate N_j. However, since K_{GN} is a shared key between GW and N_j, and N_j knows its own identity ID_j, then N_j can directly calculate x_j, which means that N_j can register itself without registering with GW for the subsequent login and authentication phase.

$$\begin{aligned} &B_j^* = h(x_j^* \| n_j^*) \\ &\text{Checks } B_j^* = B_j \end{aligned} \quad (2)$$

3.1.3. Drawbacks of the Login and Authentication Phase

There are numerous nodes in IoT and each IoT node has a unique identity to identify itself and authenticate with other entities. During the login and authentication phase of Lee et al.'s scheme [10], as shown in equation (3), each value contained in the message $M_1 = \{PID_i, UN_i, UZ_i, T_1\}$ sent by MN_i to N_j does not incorporate N_j's identity ID_j, i.e., MN_i does not indicate which IoT node it wants to contact.

$$PID_i = E_{K_G}(ID_i \| r_D)$$
$$UN_i = h(A_i \| PID_i \| n_i) \quad (3)$$
$$UZ_i = n_i \oplus A_i$$

3.2. Man-in-the-Middle (MITM) Attack

After N_j has contacted MN_i once, N_j obtains MN_i's updated pseudo-identity PID_i^{new}. Then, N_j can track MN_i when MN_i contact another IoT node N_s with PID_i^{new}. Assuming that N_j knows the identity ID_i of MN_i, and since N_j has obtained $\{n_i, UZ_i\}$ while contacting with MN_i. In that case, N_j can launch the MITM attack based on the information $\{UZ_i^*, L_s\}$ intercepted from this contact between MN_i and N_s. The specific steps of the attack are as follows.

1) In the login and authentication phase where MN_i and N_j are interconnected, N_j receives the message $M_1 = \{PID_i, UN_i, UZ_i, T_1\}$ sent by MN_i and obtains UZ_i. N_j computes n_i by

$$n_i = R_{ij} \oplus n_j$$

according to the message

$$M_3 = \{PID_i^{new}, G_j, R_{ij}, H_j\}$$

sent by GW.

2) The random number generated by MN_i during the current round of contact with N_s is noted as n_i^{**}, then

$$UZ_i^* = n_i^{**} \oplus A_i.$$

N_j intercepts the message $M_1 = \{PID_i, UN_i^*, UZ_i^*, T_1\}$ sent by MN_i to GW, and thus obtains UZ_i^*. N_j can get $n_i \oplus n_i^{**}$ by dissimilarizing UZ_i^* with the UZ_i obtained by N_j in the previous round of contact with MN_i.

3) Since N_j obtained n_i in the last contact between N_j and MN_i, N_j can calculate n_i^{**} based on $n_i \oplus n_i^{**}$ and n_i, then N_j can compute $F_s = h(ID_i \| n_i^{**})$ for the current contact between MN_i and N_s.

4) At this time when MN_i contacts N_s, N_j intercepts the message

$$M_4 = \{PID_i^{new}, L_s, SV_s, T_2\}$$

sent by N_s to MN_i, thus obtaining L_s. Due to $L_s = m_s \oplus h(ID_s \| n_i^{**})$, and the identity ID_s of N_s are public, N_j can calculate m_s.

5) In summary, N_j can calculate the session key

$$SK_{is} = SK_{si} = h(F_s \| n_i^{**} \| m_s)$$

reached between MN_i and N_s in the current round of MN_i-N_s contact.

3.3. Impersonation Attack

According to Section 3.2, N_j records the obtained values $\{n_i, UZ_i, PID_i^{new}\}$ when contacting MN_i. Therefore, N_j can then act as an attacker to impersonate MN_i to N_k when MN_i wants to contact another IoT node N_k,

1) Since N_j knows MNi's updated pseudo-identity PID_i^{new}, N_j can track MN_i based on PID_i^{new} in $M_1 = \{PID_i^{new}, UN_i', UZ_i', T_1\}$ and tamper with M_1 when MN_i contacts N_k using PID_i^{new}. N_j chooses a random number n_i'. when MN_i contacts N_k using PID_i^{new}. Then, N_j can calculate A_i by

$$A_i = UZ_i \oplus n_i$$

based one $\{n_i, UZ_i\}$ obtained in the previous round of contacting MN_i. From $A_i = h(K_G \| ID_i)$, A_i is constant in each contact, so N_j can continue to use A_i for the following calculations in the current round when MN_i contacts N_k

$$UN_i' = h(A_i \| PID_i^{new} \| n_i'),$$
$$UZ_i' = n_i' \oplus A_i.$$

N_j sends $M_1' = \{PID_i^{new}, UN_i', UZ_i', T_1\}$ to N_k.

2) N_k firstly checks the validity of timestamp T_1 based on the receipt of message M_1'. If T_1 is invalid, N_k terminates the current phase. Otherwise, N_k generates a random number n_k, calculates

$$x_k = y_k \oplus h(K_{GN} \| r_k \| ID_k),$$
$$A_k = h(x_k) \oplus n_k,$$
$$B_k = h(x_k \| n_k),$$

and send $M_2' = \{M_1', NID_k, A_k, B_k\}$ to GW.

3) When GW receives M_2', it first checks if $|T_{fresh} - T_1| \le \Delta T$ is valid. GW will terminate the current phase, if it fails. Otherwise, GW calculates

$$x_k^* = h(ID_k \| K_{GN}),$$
$$n_k^* = h(x_k) \oplus A_k,$$
$$B_k^* = h(x_k^* \| n_k^*).$$

GW verifies equation $B_k^* = B_k$ and finds that equations hold, i.e., GW can authenticate N_k.

4) GW computes

$$< ID_i, r_D^{new} >= D_{K_G}(PID_i^{new}),$$
$$A_i^* = h(K_G \| ID_i),$$
$$n_i'^* = UZ_i' \oplus A_i^*,$$
$$UN_i'^* = h(A_i^* \| PID_i^{new} \| n_i'^*).$$

After GW validates equation $UN_i'^* = UN_i'$ and finds that the equation is correct, then it means that GW authenticates the user impersonated by N_j. Then, both GW and N_k will proceed normally with the rest of the login and authentication phase.

5) Eventually, N_k and attacker N_j agree on a session key, i.e., attacker N_j successfully impersonates the user.

4. Proposed Scheme

In order to overcome the shortcomings of Lee et al.'s scheme [10], an improved scheme is proposed in this section. Firstly, the improved scheme protects the key values stored in the mobile device against the stolen mobile device attack during the user registration phase. Secondly, in the

registration phase of IoT node, the improved scheme transmits information through a secure channel and simplifies the registration process. Thirdly, messages transmitted over the public channel are added with timestamps to resist replay attacks during the login and authentication phase of the improved scheme. In addition, the updated user pseudo-identity PID_i^{new} will not be transmitted to the user through the public channel, thus ensuring the untrace ability of the scheme.

For the sake of brevity, this section only describes the registration for user, the registration for IoT node, the login and the key agreement phase. The specific steps are as follows.

4.1. Registration for User

As soon as a new user wants to access an IoT service and contact one of the IoT nodes, he/she must first register with GW. GW stores the user's registration information in order to verify his/her identity during the login phase. As shown in Figure 1, this phase is divided into three steps.

1) MN_i enters his/her identity ID_i, password PW_i and biometric BIO_i. The mobile device selects a random number $n_m \in Z_q^*$. The mobile device extracts two strings $\mathrm{Gen}(BIO_i) = (\sigma_i, \rho_i)$ from the BIO_i using Gen (·) of the fuzzy extractor and computes $PWB_i = h(PW_i \| \sigma_i)$. Then, MN_i sends the registration information $\{ID_i, PWB_i, n_m\}$ to GW through a secure channel.

2) Once GW receives the registration information from MN_i, it generates the pseudo-identity PID_i of MN_i and chooses a random number $n_G \in Z_q^*$. GW calculates some values by equation (4) and then stores $\{PID_i, ID_i, n_G, n_m\}$ in its own database, which will be sent to MN_i through a secure channel.

$$\begin{aligned} x_i &= h(ID_i \| PWB_i \| n_m) \\ A_i &= h(K_G \| ID_i \| n_G) \\ y_i &= h(ID_i \| n_m) \oplus A_i \end{aligned} \tag{4}$$

3) MN_i protects the random number n_m of its choice by

$$n_m' = n_m \oplus \sigma_i$$

and stores $\{\mathrm{Gen}(\cdot), \mathrm{Rep}(\cdot), PID_i, \rho_i, x_i, y_i, n_m'\}$ in the mobile device.

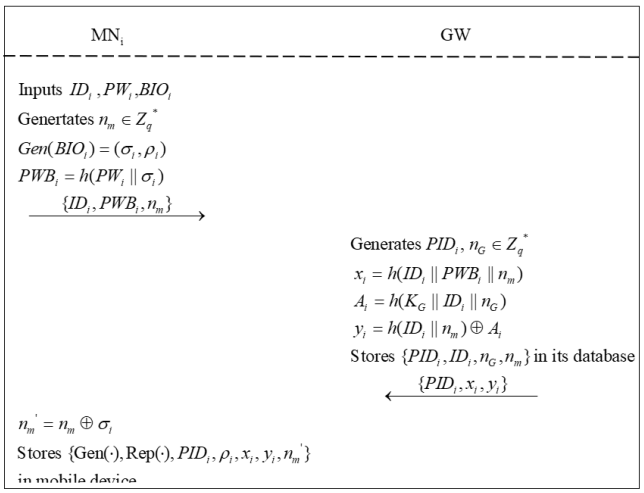

Fig 1. Registration for user

4.2. Registration for IoT Node

As shown in Figure 2, the registration for IoT node is divided into three steps as follows.

1) IoT node N_j selects its own identity ID_j and sends the registration information $\{ID_j\}$ to GW through a secure channel.

2) After receiving the registration information from N_j, GW selects a random number $n_w \in Z_q^*$, calculates N_j's pseudo-identity NID_j and a secret value x_j by equation (5), and stores

$$\{ID_j, NID_j, n_w\}$$

in its own database. Then, GW sends $\{NID_j, x_j\}$ to N_j through a secure channel and discloses N_j's pseudo-identity NID_j.

$$\begin{aligned} NID_j &= h(ID_j \| n_w) \\ x_j &= h(NID_j \| K_G \| n_w) \end{aligned} \tag{5}$$

3) After receiving $\{NID_j, x_j\}$ from GW, N_j stores $\{NID_j, x_j\}$ into its own memory.

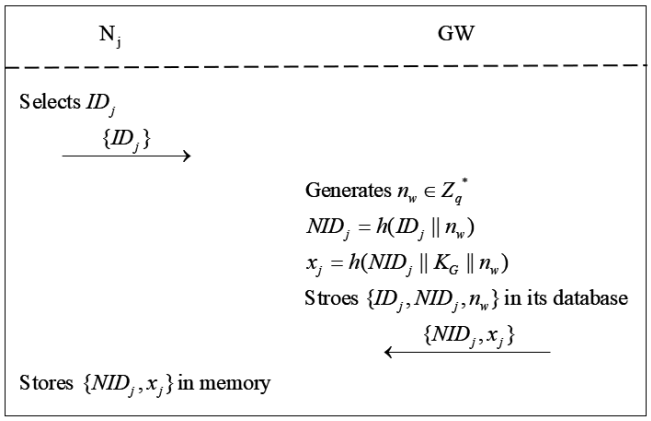

Fig 2. Registration for IoT node

4.3. Login and Key Agreement Phase

As shown in Figure 3, MN_i and N_j authenticate with each other with the help of GW and establish a session key. The detailed steps are as follows.

1) MN_i inputs his/her identity ID_i, password PW_i and biometric BIO_i. MN_i's mobile device uses the stored values ρ_i, BIO_i and Rep (·) algorithm of the fuzzy extractor to retrieve the string

$$\sigma_i = \mathrm{Rep}(BIO_i, \rho_i)$$

and compute PWB_i, n_m, x_i^* by equation (6).

$$\begin{aligned} PWB_i &= h(PW_i \| \sigma_i) \\ n_m &= n_m' \oplus \sigma_i \\ x_i^* &= h(ID_i \| PWB_i \| n_m) \end{aligned} \tag{6}$$

The mobile device verifies whether MN_i is a registered user by checking whether $x_i^* = x_i$ holds. If fails, the mobile device rejects MN_i's login request. Otherwise, the mobile device generates a random number $r_i \in Z_q^*$ and the current timestamp T_1, and calculates A_i, F_1, V_1 by equation (7).

$$A_i = y_i \oplus h(ID_i \| n_m)$$
$$F_1 = r_i \oplus h(A_i \| T_1) \quad (7)$$
$$V_1 = h(ID_i \| A_i \| NID_j \| r_i \| T_1)$$

MN_i sends $M_1 = \{PID_i, F_1, V_1, T_1\}$ through a public channel to the IoT node N_j.

2) N_j checks the validity of T_1 by $| T_{fresh} - T_1 | \leq \Delta T$, where T_{fresh} is the current timestamp. If T_1 is invalid, N_j terminates the current phase. Otherwise, N_j generates a random number $r_j \in Z_q^*$ and the current timestamp T_2, and computes F_2, V_2 by equation (8)

$$F_2 = h(x_j \| T_2) \oplus r_j$$
$$V_2 = h(ID_j \| x_j \| NID_j \| r_j \| T_2) \quad (8)$$

N_j transmits $M_2 = \{M_1, NID_j, F_2, V_2, T_2\}$ over a public channel.

3) GW checks the validity of T_1 and T_2 by $| T_{fresh} - T_1 | \leq \Delta T, | T_{fresh} - T_2 | \leq \Delta T$. If T_1 and T_2 are invalid, GW terminates the current phase. Otherwise, GW retrieves $\{ID_j, n_w\}$ from its own database using NID_j and calculates x_j^*, r_j^*, V_2^* by equation (9)

$$x_j^* = h(NID_j \| K_G \| n_w)$$
$$r_j^* = h(x_j^* \| T_2) \oplus F_2 \quad (9)$$
$$V_2^* = h(ID_j \| x_j^* \| NID_j \| r_j^* \| T_2)$$

GW verifies whether the equation $V_2^* = V_2$ holds. If not, GW will terminate the current session; otherwise, GW extracts $\{ID_i, n_G, n_m\}$ from its own database according to PID_i and computes A_i^*, r_i^*, V_1^* by equation (10).

$$A_i^* = h(K_G \| ID_i \| n_G)$$
$$r_i^* = F_1 \oplus h(A_i^* \| T_1) \quad (10)$$
$$V_1^* = h(ID_i \| A_i^* \| NID_j \| r_i^* \| T_1)$$

GW authenticates MN_i by checking whether V_1^* and V_1 are equal. If not, GW will terminate the current session. Otherwise, GW generates the current timestamp T_3, calculates F_3, F_4, V_3 and the updated user's pseudo-identity PID_i^{new} by equation (11). After that, GW will save PID_i^{new} to its own database and replace PID_i.

$$F_3 = h(ID_i \| r_i^* \| T_3)$$
$$F_4 = (F_3 \| r_i^*) \oplus h(x_j^* \| r_j^* \| T_3)$$
$$V_3 = h(x_j^* \| r_j^* \| r_i^* \| F_3 \| T_3) \quad (11)$$
$$PID_i^{new} = PID_i \oplus h(n_m \| T_3)$$
Stores PID_i^{new} and replaces PID_i

Finally, GW delivers $M_3 = \{F_4, V_3, T_3\}$ over a public channel.

4) N_j checks the validity of T_3 by $| T_{fresh} - T_3 | \leq \Delta T$. If T_3 is fresh, N_j recovers the key value $(F_3^* \| r_i^{**}) = F_4 \oplus h(x_j \| r_j \| T_3)$ used in the subsequent computation and computes $V_3^* = h(x_j \| r_j \| r_i^{**} \| F_3^* \| T_3)$. N_j verifies that equation $V_3^* = V_3$ is correct. If it is incorrect, N_j will terminate the current phase. Otherwise, N_j generates a random number $r_n \in Z_q^*$ and the current timestamp T_4, and calculates F_5, session key SK_{ji} and the verification value V_4.

$$F_5 = r_n \oplus h(NID_j \| r_i^{**} \| T_4)$$
$$SK_{ji} = h(F_3^* \| r_i^{**} \| r_n \| T_4) \quad (12)$$
$$V_4 = h(SK_{ji} \| r_n \| r_i^{**} \| T_4)$$

N_j transmits $M_4 = \{F_5, V_4, T_3, T_4\}$ to MN_i via a public channel.

5) MN_i checks the validity of T_3 and T_4 according to

$$| T_{fresh} - T_3 | \leq \Delta T \text{ ,}$$
$$| T_{fresh} - T_4 | \leq \Delta T \text{ .}$$

If T_3 and T_4 are invalid, MN_i terminates the current phase. Otherwise, MN_i calculates r_n^*, V_4^* and session key SK_{ij} by equation (13).

$$r_n^* = F_5 \oplus h(NID_j \| r_i \| T_4)$$
$$SK_{ij} = h(h(ID_i \| r_i \| T_3) \| r_i \| r_n^* \| T_4) \quad (13)$$
$$V_4^* = h(SK_{ij} \| r_n^* \| r_i \| T_4)$$

MN_i checks if V_4^* is the same as V_4. If not, MN_i will terminate the current session. If the two values are equal, it proves that MN_i and N_j establish the same session key. Then, MN_i calculates the updated pseudo-identity

$$PID_i^{new} = PID_i \oplus h(n_m \| T_3)$$

and replaces PID_i.

5. Security Analysis

This section provides a security analysis of the improved scheme. As described below, the improved scheme satisfies user anonymity and untraceability, and is also resistant to known attacks.

5.1. User Anonymity

In the improved scheme, user MN_i's identity ID_i is neither stored in the mobile device nor transmitted in the public channel, but a pseudo-identity PID_i is transmitted. PID_i is generated by GW for each user during the user registration phase. Even if an attacker intercepts PID_i, he/she cannot obtain the user's real identity information. In summary, the improved scheme satisfies user anonymity.

5.2. User Untraceability

The attacker can intercept the message M_1. However, as shown in equation (14), the values $\{PID_i, F_1, V_1, T_1\}$ contained in M_1 are associated with random numbers n_m, r_i, respectively, and these random numbers vary from session to session. That is, all values in M_1 are not associated with a

specific user. Therefore, the attacker cannot trace the user's actions in the login and authentication phase.

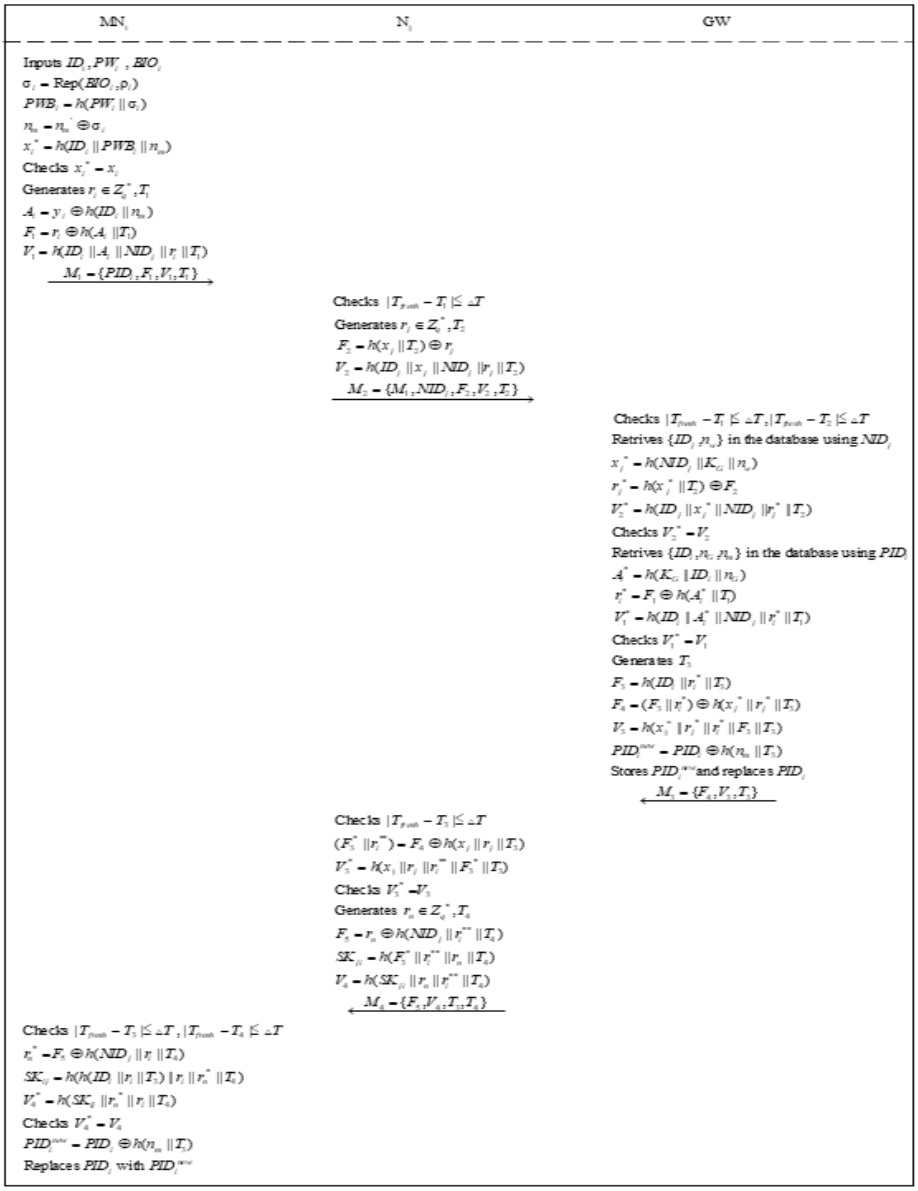

Fig 3. Login and key agreement phase

$$F_1 = r_i \oplus h(A_i \,||\, T_1)$$
$$V_1 = h(ID_i \,||\, A_i \,||\, NID_j \,||\, r_i \,||\, T_1) \quad (14)$$

Meanwhile, as shown in equation (15), GW will update and store PID_i after each session. Moreover, the updated pseudo-identity PID_i^{new} is not directly transmitted over the public channel. MN_i can calculate the PID_i^{new} by itself to be used in the next session. In a word, the attacker cannot obtain the correlation between PID_i and PID_i^{new} by intercepting them. Therefore, the improved scheme has untraceability.

$$PID_i^{new} = PID_i \oplus h(n_m \,||\, T_3) \quad (15)$$

5.3. Resistance to Man-in-the-middle (MITM) Attack

If an attacker attempts to manipulate a message transmitted in the public channel, he/she will be caught by the mutual authentication mechanism of each of the entities involved in the authentication. GW authenticates N_j and MN_i through V_1 and V_2 in equation (16), respectively. N_j achieves authentication to the GW by verifying V_3 in equation (16). MN_i checks whether the same session key is reached with N_j by validating V_4 in equation (16). Therefore, malicious attempts by attackers will not succeed and the improved scheme can resist MITM attacks.

$$V_1 = h(ID_i \,||\, A_i \,||\, NID_j \,||\, r_i \,||\, T_1)$$
$$V_2 = h(ID_j \,||\, x_j \,||\, NID_j \,||r_j \,||\, T_2)$$
$$V_3 = h(x_j^* \,||\, r_j^* \,||\, r_i^* \,||\, F_3 \,||\, T_3) \quad (16)$$
$$V_4 = h(SK_{ji} \,||\, r_n \,||\, r_i^{**} \,||\, T_4)$$

5.4. Resistance to Impersonation Attack

In the improved scheme, MN_i uses a pseudo-identity PID_i to transmit identity information over the public channel, which is updated at the end of each session. Therefore, it is difficult for the attacker to use an outdated pseudo-identity to impersonate a legitimate user. Even if the attacker happens to guess the identity of MN_i, it is impossible for the attacker to send a valid message to GW for proving his/her identity. It is because that the attacker cannot obtain a secret value

$A_i = h(K_G \| ID_i \| n_G)$ calculated by GW for MN_i during the registration phase, and A_i contains GW's private key K_G, MN_i's identity ID_i, and a random number n_G chosen by GW. Therefore, the improved scheme can resist the impersonation user attack.

5.5. Resistance to Replay Attack

During the login and authentication phase of the improved scheme, each entity performs a series of calculations to generate the current timestamp T_x, and adds T_x to each value transmitted over the public channel to ensure the freshness of the message. After each entity receives the message, it first verifies the validity of T_x and then proceeds to subsequent operations. Therefore, an attacker cannot create a session key by intercepting the information transmitted in the public channel and send the message through the session key. In summary, the improved scheme can resist replay attack.

6. Performance Analysis

This section compares the improved scheme with Lee et al.'s scheme[10] and other schemes of the same type in terms of computational cost and security.

6.1. Implementation Setup

This section refers to the experimental results of Xie et al.[11]. For convenience, only four main cryptographic operations are considered in this section: (1) one-way hash function (2) point operation (3) symmetric encryption/ decryption (4) fuzzy extraction function. Their respective estimated times are shown in Table 2.

Table 2. Estimated time

Operation	Description	Estimated time/ms
T_h	Unit calculated cost of hash function	0.068
T_s	Unit calculated cost of encryption/decryption	0.56
T_e	Unit calculated cost of operation	2.501
T_f	Unit calculated cost of fuzzy extraction function	2.501

6.2. Computation Comparisons

The comparison between the improved scheme and the same type of scheme in terms of computational cost is shown in Table 3. Although the running time of the improved scheme is slightly longer than that of the Lee et al.'s scheme [10], the improved scheme solves the defects in the Lee et al.'s scheme[10] and significantly improves the safety. It can be more intuitively seen from Fig. 4 that the difference in computational cost between improved scheme and Lee et al.'s scheme[10] is small. Meanwhile, the computational cost of the improved scheme is significantly lower than that of the schemes of Rafique et al.[12], Sahoo et al.[13], Chaudhry et al.[14], and Srinivas et al.[15].

Table 3. Computational cost of the schemes

Schemes	User	IoT nodes	Gateway	Total	Estimated time/ms
Lee et al.[10]	$9T_h$	$7T_h$	$7T_h+2T_s$	$23T_h+2T_s$	2.684
Rafique et al.[12]	$10T_h+T_s+T_f$	$4T_h+2T_s$	$8T_h+4T_s$	$22T_h+7T_s+T_f$	7.917
Sahoo et al.[13]	$10T_h+4T_e+2T_s+T_f$	$5T_h+2T_e+2T_s$	$5T_h+2T_e+T_s$	$20T_h+8T_e+5T_s+T_f$	26.669
Chaudhry et al.[14]	$17T_h+T_f$	$13T_h$	$8T_h$	$38T_h+T_f$	5.085
Srinivas et al.[15]	$17T_h+6T_e+T_f$	$8T_h+4T_e$	$12T_h+3T_e$	$37T_h+13T_e+T_f$	37.53
Improved scheme	$10T_h+T_f$	$7T_h$	$10T_h$	$27T_h+T_f$	4.337

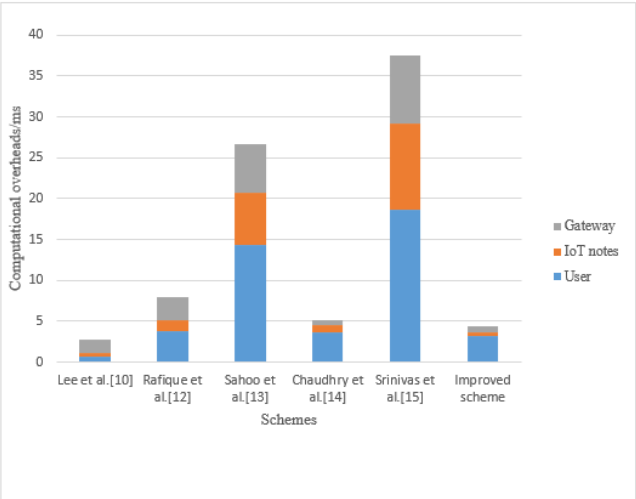

Fig 4. Comparison of computation cost

6.3. Comparison of Safety Features and Functions

In this section, the improved scheme is compared with Lee et al.'s [10] scheme and similar schemes in recent years for safety. As shown in Table 4, the existing schemes are not resistant to some attacks and their security needs to be enhanced. Compared with these schemes, the improved scheme has great security advantages and is suitable for resource-constrained IoT devices.

R1: User anonymity; R2: User untrace ability; R3: Mutual authentication; R4: Resistance to reply attack; R5: Resistance to MITM attack; R6: Resistance to DOS attack; R7: Forward security; R8: Resistance to impersonation attack; R9: Resistance to stolen smart card attack; R10: Resistance to known session key attack; R11: Resistance to off-line password guessing attack;

√: denotes the scheme can provide the corresponding attribute; ×: denotes the scheme cannot provide the corresponding attribute.

Table 4. Security comparison among relevant schemes

Schemes	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
Lee et al.[10]	√	×	√	×	×	×	√	√	×	√	√
Rafique et al.[12]	×	√	√	√	√	√	×	√	×	√	√
Sahoo et al.[13]	√	√	×	√	√	×	√	√	√	×	√
Chaudhry et al.[14]	√	√	√	√	√	√	×	√	×	√	√
Srinivas et al.[15]	×	√	×	×	×	√	×	√	×	√	×
Improved scheme	√	√	√	√	√	√	√	√	√	√	√

7. Conclusion

The paper first reviews a three-factor anonymous authentication scheme proposed by Lee et al. in an IoT environment and points out that their scheme not only has shortcomings in the user registration phase, the IoT node registration phase, and the login and authentication phases, but also fails to resist man-in-the-middle attacks and impersonation attacks. In order to overcome these shortcomings, this paper proposes an improved scheme to address the security vulnerabilities of Lee et al.'s scheme. Through security analysis, it proves that the improved scheme can resist various known attacks and meet all security requirements. In addition, this paper also compares and analyzes the security and computational effort of the proposed scheme with similar schemes in recent years. The analysis results show that the improved scheme achieves the expected efficiency and is suitable for the IoT environment.

References

[1] Rao P M, Deebak B D. A Comprehensive Survey on Authentication and Secure Key Management in Internet of Things: Challenges, Countermeasures, and Future Directions. Ad Hoc Networks, 2023: 103159.

[2] Nguyen D C, Ding M, Pathirana P N, et al. 6G Internet of Things: A comprehensive survey. IEEE Internet of Things Journal, 2021, 9(1): 359-383.

[3] Chen Z, Jiang Y, Song X, et al. A Survey on Zero-Knowledge Authentication for Internet of Things. Electronics. 2023, 12 (5): 1145.

[4] Chen B, Kuo W , Wuu L . Robust smart-card-based remote user password authentication scheme. International Journal of Communication Systems. 2014, 27 (2): 377-389.

[5] Jiang Q, Ma J, Li G, et al. Improvement of robust smart-card-based password authentication scheme. International Journal of Communication Systems. 2015, 28 (2): 383-393.

[6] Das A K. A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks. Peer-to-Peer Networking and Applications. 2016,9(1): 223-244.

[7] Dhillon PK , Kalra S . Secure multi-factor remote user authentication scheme for internet of things environments. International Journal of Communication Systems. 2017, 30 (16): e3323.

[8] Kumari S, Karuppiah M , Das A K , et al. A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers. Journal of Supercomputing, 2018, 74 (12): 6428-6453.

[9] Gope, Prosanta, Sikdar, et al. Lightweight and Privacy-Preserving Two-Factor Authentication Scheme for IoT Devices. IEEE Internet of Things Journal, 2019, 6(1):580-589.

[10] Lee H , Kang D , Ryu J, et al. A three-factor anonymous user authentication scheme for Internet of Things environments. Journal of Information Security and Applications, 2020, 52: 2214-2126.

[11] Xie Q, Wong D S, Wang G, et al. Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model. IEEE Transactions on Information Forensics and Security, 2017, 12(6): 1382-1392.

[12] F. Rafique, M. Obaidat, K. Mahmood et al. An efficient and p rovably secure certificateless protocol for industrial internet of things, IEEE Transaction Industrial Informatics, 2022,18(11): 8039-8046.

[13] Sahoo S S, Mohanty S, Majhi B. A secure three factor based authentication scheme for health care systems using IoT enabled devices. Journal of Ambient Intelligence and Humanized Computing, 2021, 12: 1419-1434.

[14] Chaudhry S A, Irshad A, Yahya K , et al. Rotating behind Privacy: An Improved Lightweight Authentication Scheme for Cloud-based IoT Environment. ACM Transactions on Internet Technology (TOIT), 2021,21(3): 1–19.

[15] Srinivas J, Das A K, Wazid M, et al. Designing secure user authentication protocol for big data collection in IoT-based intelligent transportation system. IEEE Internet of Things Journal, 2020, 8(9): 7727-7744.