Hindawi
Security and Communication Networks
Volume 2021, Article ID 8852901, 9 pages
https://doi.org/10.1155/2021/8852901

Research Article

Research on Lightweight Mutual Authentication for the Product Authorization Chain

Hanqing Ding,[1] Qing Zhang,[2] Yifeng Yin ,[1] Yong Gan,[3] and Weihua Liu[1]

[1]*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China*
[2]*China United Network Communications Group Co., Ltd., Shangqiu 476000, China*
[3]*Zhengzhou Institute of Technology, Zhengzhou 450044, China*

Correspondence should be addressed to Yifeng Yin; yinyifeng@zzuli.edu.cn

Received 1 September 2020; Revised 22 April 2021; Accepted 30 July 2021; Published 8 September 2021

Academic Editor: Tom Chen

With the development of the globalization economic integration in Internet of Things (IoT), it is very crucial to protect the wireless two-way authentication between users' intelligent terminals and servers in the product authorization chain. In order to ensure that legitimate users connect to the wireless network correctly, a lightweight wireless mutual authentication scheme for the product authorization chain was proposed contrapose to the security defect of Kaul and Awasthi's scheme, which easily suffered from offline password guessing attack. The improved scheme uses lightweight hash function and verifies the freshness of messages by using the send packet sequence number instead of timestamp, which can avoid strict clock synchronization between devices, and user passwords can be updated by themselves. Security analysis and cost and efficiency analysis show that the scheme presented in this paper has higher security, lower storage and communication costs, and lower computational complexity.

1. Introduction

With the progress of wireless technology, sensor network, and the explosion of intelligent terminals such as smart phones and smart watches, mobile users can enjoy a variety of overall service provided application, purchase products, and access to product information everywhere at any time through individual intelligent devices accessing the mobile Internet [1]. In the whole product authorization chain, from producer and agents to consumers, users want to use smart devices to purchase products at any time and place and obtain the product-related information. However, with the access of wireless network, intelligent terminals are vulnerable to unauthorized users to attack, intercept, steal, download, delete, or tamper with the private data [2].

Therefore, in the process of data transmission between the intelligent terminal and the remote server in the product authorization chain, it is particularly critical to accurately verify the identities of the wireless communication parties to ensure data security. User identity authentication can accurately identify legitimate users and assign them to server authorization to eliminate network security and malicious users. At present, a large number of user authentication schemes have been put forward.

Lamport initially designed a password-based authentication scheme in an insecure channel in 1981, but the scheme verifies the user's legitimacy by constructing a password table, which has large hash overhead and is not suitable for the practical application system [3]. Subsequently, many scholars began to study and improve the scheme that cannot achieve mutual authentication. In 2009, Wang et al. proposed an enhanced scheme to provide higher security [4]. Wen and Li demonstrated that Wang et al.'s scheme could not defend against impersonation attack in 2012 [5]. The legitimate users could initiate offline password guessing attack through obtaining the sensitive private information of other legitimate users in the system. In 2014, Chang et al. showed that Wang et al.'s scheme still used plaintext to transmit user identity in public channel [6]. Then, they designed an untraceable remote user authentication scheme on the basis of dynamic identity with a verifiable password update. In the same year, Kumari et al.

proved that Chang et al.'s scheme could not prevent offline password guessing attack, impersonation attack, and so on and proposed an improved scheme to overcome these security flaws [7]. In 2016, Kaul and Awasthi proved that Kumari et al.'s scheme is completely insecure because attackers can easily gain security parameters of the scheme and the public session key between the user and the server [8]. They proposed a new authentication scheme and proved its security. However, in 2017, Wang and Xu indicated that Kaul and Awasthi's scheme could not prevent offline password guessing attack and meet security requirements in the authentication system [9]. In view of the security level of remote user ID authentication, Liu et al. proposed an improved two-way security authentication scheme by dividing hash value into two parts in 2018 [10].

All above schemes use the timestamp to ensure whether the communication message is fresh. It is necessary to ensure strict clock synchronization between the intelligent terminal and the server. However, with the increase in intelligent terminal devices accessing wireless network, it is difficult to ensure that the clock of all intelligent terminals and servers is strictly synchronized. Many scholars have studied this problem and proposed corresponding authentication schemes. In 2016, Wang et al. improved the scheme of Wen and Li by using the send packet sequence number instead of timestamp, but the login password could not be changed freely [11, 12].

Given the above analysis, the scheme of the public key system proposed by Qiu et al. eliminates the long-standing problem of security and availability conflict in two-factor authentication mechanism by combining "honeyed words" with "fuzzy verifier" [13, 14]. This paper analyzes the security flaw of Kaul and Awasthi's scheme and proposes a lightweight wireless mutual authentication scheme for product authorization chain [15].

The sequence number PN of sending packet is constantly updated to verify the freshness of the message. The iterative operation composed of strong one-way hash function, and simple XOR is used to securely mutually authenticate the user and the server. The user can change his password and generate a secure session [16, 17]. The proposed scheme uses one-way hash function and bit exclusive or operation to realize the two-way authentication between users and servers and maintains the security advantage of the traditional scheme.

2. Problem Statement and Motivation

Due to space limitations, Kaul and Awasthi's specific implementation steps are detailed in literature [8]. Table 1 presents the notations of this study.

It is discovered that Kaul and Awasthi's scheme cannot resist offline password guessing attack. The analysis is as follows.

Suppose that an adversary $\mathscr{A}$ can gain the smart card of A_i by stealing and other means and obtain $\left\{\mathrm{SPI}_{\beta_i}, \mathrm{SPI}_{\gamma_i}, \mathrm{SPI}_{\chi_i}, \eta_i, h(\cdot)\right\}$ stored in the smart card through some technology. Due to ID_i being transmitted in plaintext on the public channel, ID_i can be gained illegally by the adversary $\mathscr{A}$ in the authentication phase [18, 19]. And then, $\mathscr{A}$ performs the following operations to realize offline password guessing attack:

(1) From the password space D_PW, $\mathscr{A}$ selects a PW_i^* to guess as PW_i value, in which $\mathrm{PW}_i \in D_{\mathrm{PW}}$

(2) It compute $b = \eta_i \oplus h(\mathrm{ID}_i^* \oplus \mathrm{PW}_i^*)$, $\mathrm{RPW}_i^* = h(b \oplus \mathrm{PW}_i^*)$, $\mathrm{SPI}_{\alpha_i}^* = \mathrm{SPI}_{\beta_i} \oplus h(\mathrm{ID}_i^* \oplus \mathrm{RPW}_i^*)$, $R_{si}^* = \mathrm{SPI}_{\gamma_i} \oplus h(\mathrm{SPI}_{\alpha_i}^* \oplus \mathrm{RPW}_i^*)$, and $\mathrm{SPI}_{\chi_i}^* = h(\mathrm{ID}_i^* \| \mathrm{RPW}_i^* \| R_{si}^* \| \mathrm{SPI}_{\alpha_i}^*)$

(3) It checks $\mathrm{SPI}_{\chi_i}^* = \mathrm{SPI}_{\chi_i}$ to verify the correctness of PW_i^*; if not, repeat (1), (2), and (3) until PW_i is found

From the above analysis, the time complexity of this attack process is $O(\|D_{\mathrm{PW}}\|^* (5T_h + 5T_{\mathrm{XOR}}))$, T_h is the runtime of the hash function operation, T_XOR is the runtime of XOR operation, and $|D_PW|$ expresses the number of passwords in D_PW. $|D_PW|$ is very limited in practice, usually $D_{\mathrm{PW}} \leq 10^6$ [12]. Therefore, the above attack is very effective.

The scheme needs to include the encryption information of the product corresponding to the current authorization and the authorization information of all or part of the earlier authorized products. The product authorization chain is also important in tracking product flow through the logistics pipelines. Through the after-sales service tracking of after-sales certification authorized products or service behaviors, a trusted platform based on consumer information binding product information is formed to realize product and user information feedback and transmission under big data. It ensures consumers' understanding, exquisite manufacturing, and sincere service so as to realize the healthy cycle of social consumption circle.

3. Our Proposed Scheme

3.1. The Model of Product Authorization Chain. Contrapose to the existing problem in Kaul and Awasthi's authentication scheme, this paper proposes a model of product authorization chain, which consists of three flows: data flow, product flow, and product information flow. Product information flow flows the most frequently and the most sensitively responses to authorization chain, and it affects the product flow and subordinate agents information, and it is the main basis of authorization chain decision-making. In view of attacks data communication between both parties, the polymorphic authentication service protocol, due to the role of the built-in self-compiling system of the security subsystem, makes the active attacker face the improved virtual iterative function polytropic function set [20]. By using multiagent technology, the model of product authorization chain typically includes three roles: producer, consumer, and n-level agent. Producers with root access rights can access all information of products from the cloud server [21]. Other roles are authorized by the superior and can access the corresponding product information.

Figure 1 illustrates the authorization process of the product authorization chain. The producer owns the complete product information such as product composition,

Table 1: Notations.

Symbols	Meaning
A_i	ith level agent user
ID_i	Unique identity of A_i
PW_i	Login password of A_i
PW_i^N	The login password to update for A_i
S	Server
SPI_{α_i}, SPI_{β_i}, $\mathrm{SPI}_{\gamma_i}\mathrm{SPI}_{\chi_i}$	The ith values of four security parameters index
x, y	Secret key and number of S
R_{si}	Unique random number for A_i distributed by S
PN_i	A_i saving the last send packet sequence number in login phase
PN_s	S saving the last send packet sequence number in authentication phase
$\mathrm{PN}_{(i+1)}$	A_i saving the last send packet sequence number in authentication phase
$h(\cdot)$	Strong one-way hash function

processing technology, and production cost. Each level of agent can obtain the corresponding amount of information through registration, paging, authorization, and encoding. An agent in authorization chain distributes the information it receives from a superior agent only if it is authorized and not solely based on the discrepancy among superior and subordinate agents information, which might be caused by legitimate changes and not attacks. The product information flow received by primary agent, secondary agent, n-level agent, and consumer is the same or different, in which P_{a1}, P_{a2}, P_{a1n+i}, and P_{a2n+j} may be equal or different.

This paper proposes a lightweight wireless two-way authentication scheme for product authentication chain, which improves the security performance and efficiency of the authentication system by the improvement based on Kaul and Awasthi's scheme. Figure 2 illustrates the authentication process of registration, login, authentication, and password changing phase of the proposed scheme.

The validity of the improved scheme can be verified by logical reasoning of the security model. Burrows–Abadi–Needham (BAN) logic is used to prove the security of the proposed scheme in this paper [22]. The two authentication entities in the scheme are the ith level agent user A_i and the server S. The transmission plaintext irrelevant to the security attribute to be demonstrated in the authentication process is eliminated.

Only the security attributes and logical parts related to mutual authentication are retained. The ideal goal formula of the authorization chain model is as follows:

$$A_i \longrightarrow S\colon \left(\langle y_i, \mathrm{SPI}_{\alpha_i}, \mathrm{PN}_i\rangle_{\mathrm{ID}_i}, \mathrm{ID}_i, y_i, \mathrm{SPI}_{\alpha_i}, \mathrm{PN}_i\right), \tag{1}$$

$$S \longrightarrow A_i\colon \langle y_i, \mathrm{SPI}_{\alpha_i}, \mathrm{PN}_s\rangle_{\mathrm{ID}_i}. \tag{2}$$

ID_i is the secret shared by both A_i and the server S. The ultimate objective of mutual authentication scheme is as follows:

$$A_i \,|\!\equiv S \overset{\mathrm{SK}}{\longleftrightarrow} A_i, \tag{3}$$

$$S \,|\!\equiv A_i \overset{\mathrm{SK}}{\longleftrightarrow} S. \tag{4}$$

The scheme is initialized and assumed that

$$A_i| \equiv S \overset{\mathrm{ID}_i}{\leftrightarrow} A_i, \tag{5}$$

$$S| \equiv A_i \overset{\mathrm{ID}_i}{\leftrightarrow} S. \tag{6}$$

In order to achieve the final goal (4) of the scheme, we will prove the following main logical conclusion.

It can be deduced from the idealized model (1).

$$S \triangleleft \left(\langle y_i, \mathrm{SPI}_{\alpha_i}, \mathrm{PN}_i\rangle_{\mathrm{ID}_i}, \mathrm{ID}_i, y_i, \mathrm{SPI}_{\alpha_i}, \mathrm{PN}_i\right). \tag{7}$$

Based on assumptions (6), the result can be obtained by substituting (7) into R1: $(P| \equiv P \overset{k}{\leftrightarrow} Q, P \triangleleft \{X\}_k / P| \equiv Q| \sim X)$.

$$S| \equiv A_i| \sim \left(y_i, \mathrm{SPI}_{\alpha_i}, \mathrm{PN}_i\right). \tag{8}$$

The inference rules of the freshness in BAN logic can be applied to infer the formula.

$$S| \equiv \#\left(y_i, \mathrm{SPI}_{\alpha_i}, \mathrm{PN}_i\right). \tag{9}$$

Result (10) can be obtained by combining (8) and (9) with fresh value validation rules $R4$.

$$S| \equiv A_i| \equiv \left(y_i, \mathrm{SPI}_{\alpha_i}, \mathrm{PN}_i\right). \tag{10}$$

According to the improved scheme proposed in this paper, the session key is $\mathrm{SK} = h(\mathrm{ID}_i \| R_{si} \| \mathrm{SPI}_{\alpha_i} \| \mathrm{PN}_i \| \mathrm{PN}_S)$. The result can be derived by combining the results of (6), (9), and (10).

$$S| \equiv A_i| \equiv A_i \overset{\mathrm{SK}}{\leftrightarrow} S. \tag{11}$$

Based on assumptions (6), the result can be obtained by substituting (11) into jurisdiction rule R5: $(P| \equiv Q| \Longrightarrow X,\ P| \equiv Q| \equiv X / P| \equiv X)$.

$$S| \equiv A_i \overset{\mathrm{SK}}{\leftrightarrow} S. \tag{12}$$

The safety target (4) is achieved, and the proof is completed.

3.2. Registration Phase. So as to acquire services from S, new user A_i must register as follows:

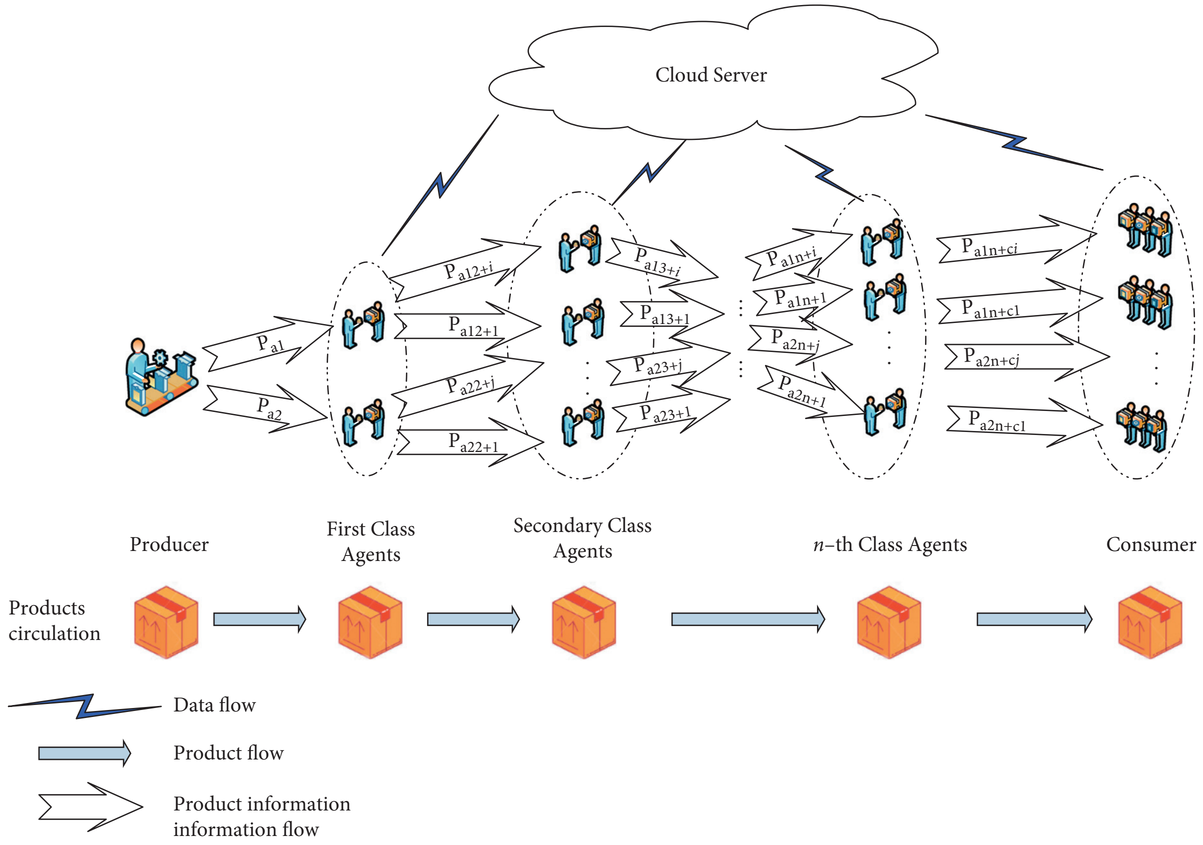

Figure 1: The model of product authorization chain.

Step 1. A_i chooses his own ID_i, PW_i, and a random number R_a, calculates $\mathrm{RPW}_i = h(R_a \oplus \mathrm{PW}_i)$, and transmits $\{\mathrm{ID}_i, \mathrm{RPW}_i\}$ to remote server S by the secure channel.

Step 2. S selects randomly an unique random number R_{si} and computes the values of four security parameter indexes (SPI) SPI_{α_i}, SPI_{β_i}, SPI_{γ_i}, and SPI_{χ_i}:

$$\begin{aligned}
\mathrm{SPI}_{\alpha_i} &= h((\mathrm{ID}_i \oplus R_{si}) \| x \| y),\\
\mathrm{SPI}_{\beta_i} &= \mathrm{SPI}_{\alpha_i} \oplus h(\mathrm{ID}_i \oplus \mathrm{RPW}_i),\\
\mathrm{SPI}_{\gamma_i} &= R_{si} \oplus h(\mathrm{SPI}_{\alpha_i} \oplus \mathrm{RPW}_i),\\
\mathrm{SPI}_{\chi_i} &= h(\mathrm{ID}_i \| \mathrm{RPW}_i \| R_{si} \| \mathrm{SPI}_{\alpha_i}),
\end{aligned}$$

It stores $\{\mathrm{SPI}_{\beta_i}, \mathrm{SPI}_{\gamma_i}, \mathrm{SPI}_{\chi_i}, h(\cdot)\}$ in the smart card and delivers it to A_i.

Step 3. A_i computes $\eta_i = b \oplus h(\mathrm{ID}_i \oplus \mathrm{PW}_i)$ and stores it in the smart card.

3.3. Login Phase

3.3. Login Phase. If A_i wants to log in S, it must insert the smart card into the card reader, and then does as follows:

Step 1. A_i inputs ID_i^* and PW_i^* and computes the following:

$$\begin{aligned}
&b = \eta_i \oplus h(\mathrm{ID}_i^* \oplus \mathrm{PW}_i^*),\ \mathrm{RPW}_i^* = h(b \oplus \mathrm{PW}_i^*),\\
&\mathrm{SPI}_{\alpha_i}^* = \mathrm{SPI}_{\beta_i} \oplus h(\mathrm{ID}_i^* \oplus \mathrm{RPW}_i^*),\\
&R_{si}^* = \mathrm{SPI}_{\gamma_i} \oplus h(\mathrm{SPI}_{\alpha_i}^* \oplus \mathrm{RPW}_i^*),\\
&\mathrm{SPI}_{\chi_i}^* = h(\mathrm{ID}_i^* \| \mathrm{RPW}_i^* \| R_{si}^* \| \mathrm{SPI}_{\alpha_i}^*).
\end{aligned}$$

If the calculated $\mathrm{SPI}_{\chi_i}^*$ is equivalent to SPI_{χ_i} saved in the smart card, the reader accepts the login request of A_i; else, exits. In addition, in order to prevent online password guessing attack, if wrong passwords are entered more than a preset number of times, the card is locked in a predefined limited period of time.

Step 2. After verifying the legality of the login request, A_i calculates λ_i, ω_i, and ϑ_i as follows:

$$\begin{aligned}
\lambda_i &= \mathrm{ID}_i \oplus R_{si},\\
\omega_i &= R_{si} \oplus h(\lambda_i \oplus \mathrm{SPI}_{\alpha_i} \oplus \mathrm{PN}_i),\\
\vartheta_i &= h(\mathrm{ID}_i \| \lambda_i \oplus \mathrm{SPI}_{\alpha_i} \oplus \mathrm{PN}_i).
\end{aligned}$$

A_i updates the sending package sequence number $\mathrm{PN}_i = \mathrm{PN}_i + 1$ and then sends the login request information $\{\lambda_i, \omega_i, \vartheta_i, \mathrm{PN}_i\}$ to the server S.

3.4. Authentication Phase. At this phase, A_i and S complete two-way authentication and establish a secure session key:

Step 1. Firstly, S verifies the freshness of the message through $\mathrm{PN}_i \overset{?}{=} \mathrm{PN}_S + 1$; if it is satisfied, S receives the login request $\{\lambda i, wi, \vartheta i, \mathrm{PN}_i\}$; else, rejects the login request.

Step 2. S computes $\mathrm{SPI}_{\alpha_i}^*$, R_{si}^*, ID_i^*, and ϑ_i^* as follows:

$$\begin{aligned}
\mathrm{SPI}_{\alpha_i}^* &= h(\lambda_i^* \| x \| y),\\
R_{si}^* &= \omega_i^* \oplus h(\lambda_i^* \oplus \mathrm{SPI}_{\alpha_i}^* \oplus \mathrm{PN}_s),\\
\mathrm{ID}_i^* &= R_{si}^* \oplus \lambda_i^*,
\end{aligned}$$

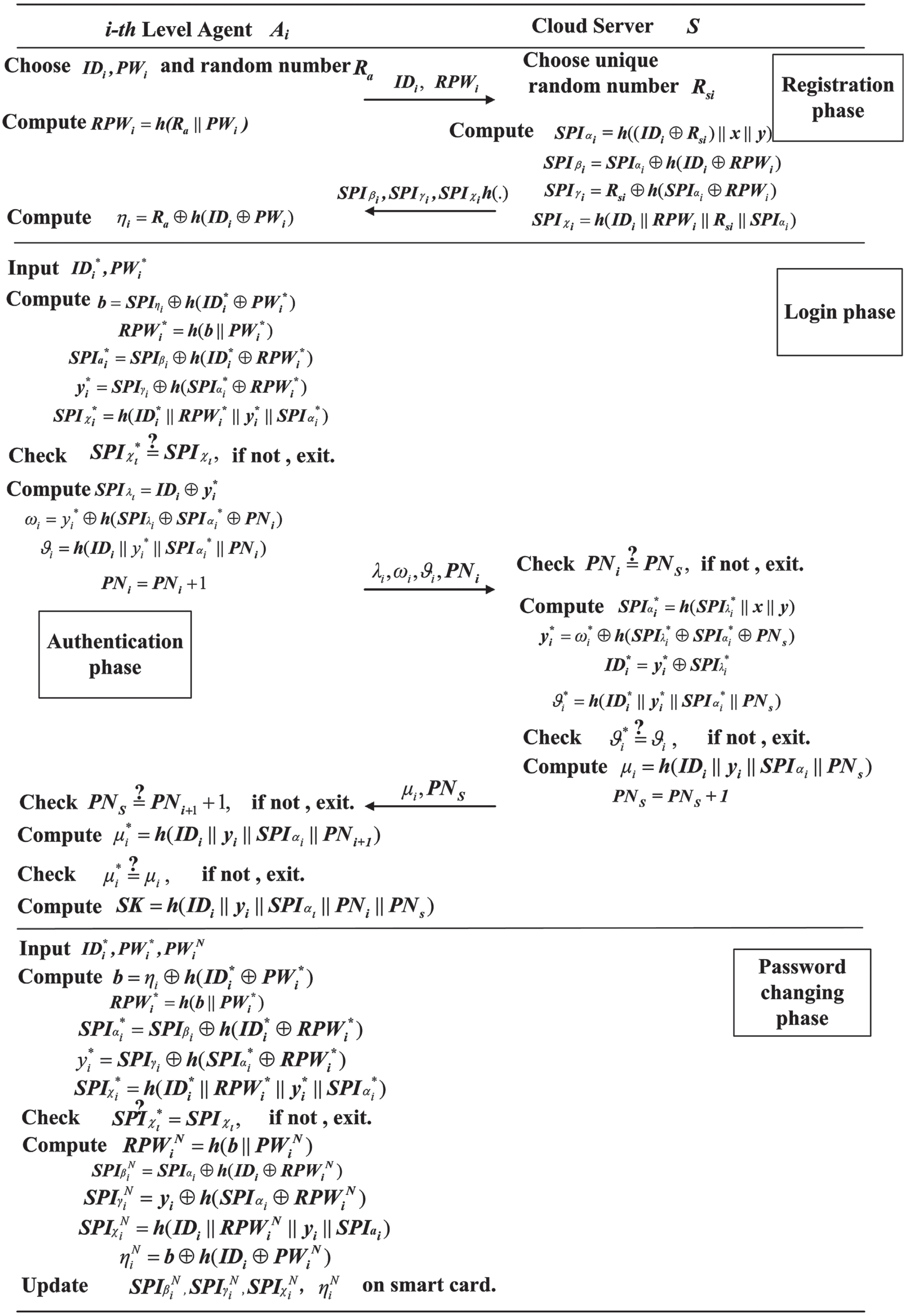
i-th Level Agent A_i
Cloud Server S
Choose ID_i, PW_i and random number R_a
ID_i, RPW_i
Choose unique random number R_{si}
Registration phase
Compute $RPW_i = h(R_a \| PW_i)$
Compute $SPI_{\alpha_i} = h((ID_i \oplus R_{si}) \| x \| y)$
$SPI_{\beta_i} = SPI_{\alpha_i} \oplus h(ID_i \oplus RPW_i)$
$SPI_{\gamma_i} = R_{si} \oplus h(SPI_{\alpha_i} \oplus RPW_i)$
$SPI_{\chi_i} = h(ID_i \| RPW_i \| R_{si} \| SPI_{\alpha_i})$
$SPI_{\beta_i}, SPI_{\gamma_i}, SPI_{\chi_i} h(.)$
Compute $\eta_i = R_a \oplus h(ID_i \oplus PW_i)$
Input ID_i^*, PW_i^*
Login phase
Compute $b = SPI_{\eta_i} \oplus h(ID_i^* \oplus PW_i^*)$
$RPW_i^* = h(b \| PW_i^*)$
$SPI_{\alpha_i}^* = SPI_{\beta_i} \oplus h(ID_i^* \oplus RPW_i^*)$
$y_i^* = SPI_{\gamma_i} \oplus h(SPI_{\alpha_i}^* \oplus RPW_i^*)$
$SPI_{\chi_i}^* = h(ID_i^* \| RPW_i^* \| y_i^* \| SPI_{\alpha_i}^*)$
Check $SPI_{\chi_t}^* \overset{?}{=} SPI_{\chi_t}$, if not , exit.
Compute $SPI_{\lambda_t} = ID_i \oplus y_i^*$
$\omega_i = y_i^* \oplus h(SPI_{\lambda_i} \oplus SPI_{\alpha_i}^* \oplus PN_i)$
$\vartheta_i = h(ID_i \| y_i^* \| SPI_{\alpha_i}^* \| PN_i)$
$PN_i = PN_i + 1$
$\lambda_i, \omega_i, \vartheta_i, PN_i$
Check $PN_i \overset{?}{=} PN_S$, if not , exit.
Authentication phase
Compute $SPI_{\alpha_i}^* = h(SPI_{\lambda_i}^* \| x \| y)$
$y_i^* = \omega_i^* \oplus h(SPI_{\lambda_i}^* \oplus SPI_{\alpha_i}^* \oplus PN_s)$
$ID_i^* = y_i^* \oplus SPI_{\lambda_i}^*$
$\vartheta_i^* = h(ID_i^* \| y_i^* \| SPI_{\alpha_i}^* \| PN_s)$
Check $\vartheta_i^* \overset{?}{=} \vartheta_i$, if not , exit.
Compute $\mu_i = h(ID_i \| y_i \| SPI_{\alpha_i} \| PN_s)$
$PN_S = PN_S + 1$
μ_i, PN_S
Check $PN_S \overset{?}{=} PN_{i+1} + 1$, if not , exit.
Compute $\mu_i^* = h(ID_i \| y_i \| SPI_{\alpha_i} \| PN_{i+1})$
Check $\mu_i^* \overset{?}{=} \mu_i$, if not , exit.
Compute $SK = h(ID_i \| y_i \| SPI_{\alpha_t} \| PN_i \| PN_s)$
Input ID_i^*, PW_i^*, PW_i^N
Password changing phase
Compute $b = \eta_i \oplus h(ID_i^* \oplus PW_i^*)$
$RPW_i^* = h(b \| PW_i^*)$
$SPI_{\alpha_i}^* = SPI_{\beta_i} \oplus h(ID_i^* \oplus RPW_i^*)$
$y_i^* = SPI_{\gamma_i} \oplus h(SPI_{\alpha_i}^* \oplus RPW_i^*)$
$SPI_{\chi_i}^* = h(ID_i^* \| RPW_i^* \| y_i^* \| SPI_{\alpha_i}^*)$
Check $SPI_{\chi_t}^* \overset{?}{=} SPI_{\chi_t}$, if not , exit.
Compute $RPW_i^N = h(b \| PW_i^N)$
$SPI_{\beta_i}^N = SPI_{\alpha_i} \oplus h(ID_i \oplus RPW_i^N)$
$SPI_{\gamma_i}^N = y_i \oplus h(SPI_{\alpha_i} \oplus RPW_i^N)$
$SPI_{\chi_i}^N = h(ID_i \| RPW_i^N \| y_i \| SPI_{\alpha_i})$
$\eta_i^N = b \oplus h(ID_i \oplus PW_i^N)$
Update $SPI_{\beta_i}^N, SPI_{\gamma_i}^N, SPI_{\chi_i}^N$, η_i^N on smart card.

Figure 2: The calculation process of the group key.

$$\vartheta_i^* = h(\mathrm{ID}_i^* \| R_{si}^* \| \mathrm{SPI}_{\alpha_i}^* \| \mathrm{PN}_s).$$

S checks $\vartheta_i^* \overset{?}{=} \vartheta_i$. If they are equal, the identity of A_i is legal; if not, S withdraws from the authentication phase.

Step 3. S computes $\mu_i = h(\mathrm{ID}_i \| R_{si} \| \mathrm{SPI}_{\alpha_i} \| \mathrm{PN}_S)$, updates the sending package serial number $\mathrm{PN}_i = \mathrm{PN}_i + 1$, and sends $\{\mu_i, \mathrm{PN}_S\}$ to A_i.

Step 4. A_i checks $\mathrm{PN}_S \overset{?}{=} \mathrm{PN}_{i+1} + 1$ to verify that the message is valid, then calculates $\mu_i^* = h(\mathrm{ID}_i \| R_{si} \| \mathrm{SPI}_{\alpha_i} \| \mathrm{PN}_{i+1})$, and judges $\mu_i^* \overset{?}{=} \mu_i$. If $\mu_i^* = \mu_i$, the identity of S is legal; if not, exits.

Step 5. After realizing two-way authentication successfully, A_i and S generate independently session key $\mathrm{SK} = h(\mathrm{ID}_i \| R_{si} \| \mathrm{SPI}_{\alpha_i} \| \mathrm{PN}_i \| \mathrm{PN}_S)$ for the future safety communication.

3.5. Password Changing Phase. At this phase, A_i can update password directly in the smart card without any help from the remote server. When the user A_i wants PW_i to be updated to PW_i^N, A_i only needs to make the smart card inserted into the smart card reader and then performs the following actions:

Step 1. A_i inputs ID_i^*, PW_i^*, and PW_i^N and sends a password update request.

Step 2. The card reader calculates b, RPW_i^*, $\mathrm{SPI}_{\alpha_i}^*$, R_{si}^*, and $\mathrm{SPI}_{\chi_i}^*$ by Step 1 in Section 3.2 and compares $\mathrm{SPI}_{\chi_i}^* \overset{?}{=} \mathrm{SPI}_{\chi_i}$; if the result is equal, it verifies that the requesting user is a legitimate user and updates their password; else, exits;

Step 3. The card reader computes RPW_i^N, $\mathrm{SPI}_{\beta_i}^N$, $\mathrm{SPI}_{\gamma_i}^N$, $\mathrm{SPI}_{\chi_i}^N$, and η_i^N as follows:

$$\begin{aligned}
\mathrm{RPW}_i^N &= h(b \| \mathrm{PW}_i^N),\\
\mathrm{SPI}_{\beta_i}^N &= \mathrm{SPI}_{\alpha_i} \oplus h(\mathrm{ID}_i \oplus \mathrm{RPW}_i^N),\\
\mathrm{SPI}_{\gamma_i}^N &= R_{si} \oplus h(\mathrm{SPI}_{\alpha_i} \oplus \mathrm{RPW}_i^N),\\
\mathrm{SPI}_{\chi_i}^N &= h(\mathrm{ID}_i \| \mathrm{RPW}_i^N \| R_{si} \| \mathrm{SPI}_{\alpha_i}),\\
\eta_i^N &= b \oplus h(\mathrm{ID}_i \oplus \mathrm{PW}_i^N),
\end{aligned}$$

It updates $\{\mathrm{SPI}_{\beta_i}, \mathrm{SPI}_{\gamma_i}, \mathrm{SPI}_{\chi_i}, \eta_i\}$ to $\{\mathrm{SPI}_{\beta_i}^N, \mathrm{SPI}_{\gamma_i}^N, \mathrm{SPI}_{\chi_i}^N, \eta_i^N\}$.

3.6. Information Inquiry and Information Authorization Phase

Step 1. Information inquiry phase: After the two-way authentication between the intelligent terminal and the server, if A_i wants to query the product information, the identity ID_i is encrypted by the secure session key generated after mutual authentication and then A_i transmits it and the query request to S. S inquires the information authorization of the authorization user granted by the upper level user, and then S sends relevant encrypted information to the user A_i through using the session key SK.

Step 2. Information authorization phase: After the user successfully logs in the system, the user sends an information authorization request to the server and uses SK to encrypt the identity of the next-level user and encrypted amount of information granted and sends it to S. The server saves it and waits for the query of the next-level user.

4. Security and Efficiency Analysis of Our Scheme

4.1. Security Analysis. The security of our scheme is investigated, which is the ability of a scheme to fend off some well-known attacks and is compared with the schemes in [4–8, 11]. Table 2 is the comparison result. "✓" indicates that it has resistance to some known attacks. "×" indicates that it cannot be defended or does not have the resistance. The result illustrates that even if attackers extract all the values saved in the smart card and intercept all communication information in public channel, the security of the proposed solution is not affected.

4.1.1. Resist Offline Password Guessing Attack. Assume an attacker $\mathscr{A}$ obtains the user's smart card and extracts all values $\{\mathrm{SPI}_{\beta_i}, \mathrm{SPI}_{\gamma_i}, \mathrm{SPI}_{\chi_i}, \eta_i\}$ saved in the smart card by taking some actions. At the same time, the security parameters $\{\lambda_i, \omega_i, \vartheta_i, \mathrm{PN}_i, \mu_i, \mathrm{PN}_S\}$ transmitted in public channel can also be obtained. Because $\lambda_i = \mathrm{ID}_i \oplus R_{si} = \mathrm{ID}_i^* \oplus R_{si}^*$, $R_{si}^* = \mathrm{SPI}_{\gamma_i} \oplus h(\mathrm{SPI}_{\alpha_i}^* \oplus RPW_i^*)$, and $\mathrm{SPI}_{\alpha_i}^* = \mathrm{SPI}_{\beta_i} \oplus h(\mathrm{ID}_i^* \oplus \mathrm{RPW}_i^*)$, in which ID_i^* and RPW_i^* are unknown, $\mathscr{A}$ needs to correctly surmise the values of two unknowns at least each time, which is impossible in polynomial time [13]. Therefore, $\mathscr{A}$ cannot guess the values ID_i and PW_i from all the gained information. Therefore, the scheme proposed in this article can defend against offline password guessing attack.

4.1.2. Resist Impersonation Attack

(1) *Impersonating Legitimate User A_i.* For counterfeiting A_i, $\mathscr{A}$ must be able to calculate the correct user login request $\{\lambda_i, \omega_i, \vartheta_i, \mathrm{PN}_i\}$; as can be seen from the model of product authorization chain, it is impossible for $\mathscr{A}$ to get useful information from the obtained information from the missing or stolen smart card to infer ID_i, PW_i, b, and R_{si}. And $\mathscr{A}$ cannot gain the server's private key x and random number R_{si} to calculate $\mathrm{SPI}_{\alpha_i} = h((\mathrm{ID}_i \oplus R_{si}) \| x \| y)$. So, $\mathscr{A}$ cannot successfully disguise as a legitimate user A_i.

(2) *Impersonating Server S.* The server does not reveal the unique key x and number y. Hash function has the strong unidirectionality; hence, $\mathscr{A}$ cannot calculate $\mathrm{SPI}_{\alpha_i}^* = h(\lambda_i^* \| x \| y)$ accurately and then calculates $\vartheta_i^* = h(\mathrm{ID}_i^* \| R_{si}^* \| \mathrm{SPI}_{\alpha_i}^* \| \mathrm{PN}_S)$ to make $\vartheta_i^* = \vartheta_i$. So, $\mathscr{A}$ is not able to successfully dress up as a server.

4.1.3. Resist DoS Attack. When A_i logs in the server, the smart card first checks whether the input ID_i and PW_i are correct or not. Only with the correct input ID_i and PW_i, the smart card would accept the login request. Therefore, it does not exist that A_i inputs error login request ID_i and PW_i to

Table 2: Security comparisons with other schemes.

Attack testing	Wang et al.'s [4]	Wen and Li's [5]	Chang et al.'s [6]	Kumari et al.'s [7]	Wang et al.'s [11]	Kaul and Awasthi's [8]	Ours
Resist offline password guessing attack	×	×	×	×	✓	×	✓
Resist replay attack	✓	✓	✓	✓	✓	✓	✓
Resist impersonation attack	×	×	×	✓	✓	✓	✓
Resist DoS attack	×	×	×	✓	×	✓	✓
Resist insider attack	×	✓	×	×	✓	✓	✓
Resist man in middle attack	✓	×	✓	×	✓	✓	✓
Secure session key establishment	×	✓	×	✓	×	✓	✓

Table 3: The comparison of memory and communication cost.

Cost testing	Wang et al.'s [4]	Wen and Li's [5]	Chang et al.'s [6]	Kumari et al.'s [7]	Wang et al.'s [11]	Kaul and Awasthi's [8]	Ours
Memory cost	$3L$	$3L$	$3L$	$6L$	$2L$	$5L$	$5L$
Communication cost	$6L$	$9L$	$6L$	$7L$	$6L$	$6L$	$6L$

cause login request calculated incorrectly. The user is exposed to denial-of-service vulnerabilities after the server verification test failure. Similarly, it is impossible for $\mathscr{A}$ to update the values $\left\{\mathrm{SPI}_{\beta_i}, \mathrm{SPI}_{\gamma_i}, \mathrm{SPI}_{\chi_i}, \eta_i\right\}$ in the smart card with any password so that it cannot be used anymore. In addition, due to a fake login request or a legitimate user's wrong operation or malicious attacks from $\mathscr{A}$, when the number of failed logins exceeds a predetermined value, the card would be locked during a certain period of time, which economizes on server time, cost, and computing resources. Therefore, the scheme can avoid the situation of the DoS caused by a wrong operation of a legitimate user or a malicious attack by an attacker.

4.1.4. Resist Man in Middle Attack. Suppose that $\mathscr{A}$ gets all the parameters transmitted on the public network channel. The authentication message is $\vartheta_i = h(\mathrm{ID}_i \| R_{si} \| \mathrm{SPI}_{\alpha_i} \| \mathrm{PN}_i)$ and $\mu_i = h(\mathrm{ID}_i \| R_{si} \| \mathrm{SPI}_{\alpha_i} \| \mathrm{PN}_i)$, in which $\mathrm{SPI}_{\alpha_i} = h\ ((\mathrm{ID}_i\ \oplus R_{si}) \| x \| y)$ is not saved in the smart card and transferred in public channel, x and y are the server private key and number, respectively, and R_{si} is the random number chosen for the server randomly. Only U' knows all the above parameters, and it is possible to intercept during the session. All the secret parameters are unlikely to be correctly guessed at the same time. Therefore, the scheme can resist man in the middle attack.

4.2. Efficiency Analysis. From the aspects of memory space, communication cost, and computational complexity, this section analyzes our scheme and other schemes in [4–8, 11] and compares the cost and efficiency of seven schemes in the registration, login, and authentication phase. Assume that all parameter byte lengths are equal to the strong one-way hash function output byte length.

Table 3 demonstrates the memory and communication cost, where L means the output byte length of the hash function. In our scheme, the security parameters saved in the smart card are $\left\{\mathrm{SPI}_{\beta_i}, \mathrm{SPI}_{\gamma_i}, \mathrm{SPI}_{\chi_i}, \eta_i, h(\cdot)\right\}$. Therefore, the memory cost is $5L$, and the communication cost includes all message bits $\{\lambda_i, \omega_i, \vartheta_i, \mathrm{PN}_i, \mu_i, \mathrm{PN}_s\}$ transferred in public network channel in the login and authentication phase. Therefore, the communication cost is $6L$. In Table 3, it illustrates that the proposed scheme is equal to the scheme in [5, 7, 8, 11], which is the lowest, in terms of communication cost. The storage overhead of our scheme is equal to Kaul and Awasthi's scheme, which indicates the proposed scheme does not increase memory cost contrast to Kaul and Awasthi's scheme [8]. Combined with the security analysis of six attacks mentioned in Table 2, only our scheme can resist all attacks. Therefore, considering the proposed scheme meets all security properties shown in Table 2, our scheme performs best in terms of security attributes and communication cost on the whole, which is more appropriate for resource-constrained intelligent terminal systems.

Table 4 shows the results of the computational complexity comparison to other scheme, in which T_h indicates the runtime of the hash function operation and T_{XOR} indicates the runtime of the XOR operation. The computational complexity of Kaul and Awasthi's scheme is $(20T_h + 28T_{\mathrm{XOR}})$, and ours has a smaller computational complexity $(18T_h + 21T_{\mathrm{XOR}})$ to meet more security needs. Our scheme only uses strong one-way hash function and simple XOR operation to operate. It is suitable for smart terminals with lightweight operation. It has low system overhead and low computational complexity, while ensuring the security and reliability of the system. The requirements of smart devices for data processing performance make them have good scalability.

The diagram in Figure 3 shows comparative analysis of the calculation complexity about four phases including Kaul and Awasthi's, AES, ECC, and ours schemes. The scheme proposed by Kaul and Awasthi is test time 57 μs. The scheme

Table 4: The comparison of computational complexity.

Scheme	Registration phase	Login phase	Authentication phase	Total
Wang et al.'s [4]	$2T_h + 2T_{\text{XOR}}$	$2T_h + 4T_{\text{XOR}}$	$5T_h + 10T_{\text{XOR}}$	$9T_h + 16T_{\text{XOR}}$
Wen and Li's [5]	$5T_h + 4T_{\text{XOR}}$	$6T_h + 6T_{\text{XOR}}$	$9T_h + 11T_{\text{XOR}}$	$20T_h + 21T_{\text{XOR}}$
Chang et al.'s [6]	$2T_h + T_{\text{XOR}}$	$4T_h + 3T_{\text{XOR}}$	$6T_h + 2T_{\text{XOR}}$	$12T_h + 6T_{\text{XOR}}$
Kumari et al.'s [7]	$4T_h + 5T_{\text{XOR}}$	$5T_h + 10T_{\text{XOR}}$	$7T_h + 3T_{\text{XOR}}$	$16T_h + 18T_{\text{XOR}}$
Wang et al.'s [11]	$2T_h + 2T_{\text{XOR}}$	$2T_h + 5T_{\text{XOR}}$	$6T_h + 9T_{\text{XOR}}$	$10T_h + 16T_{\text{XOR}}$
Kaul and Awasthi's [8]	$6T_h + 7T_{\text{XOR}}$	$8T_h + 12T_{\text{XOR}}$	$6T_h + 9T_{\text{XOR}}$	$20T_h + 28T_{\text{XOR}}$
Ours	$6T_h + 7T_{\text{XOR}}$	$7T_h + 10T_{\text{XOR}}$	$5T_h + 4T_{\text{XOR}}$	$18T_h + 21T_{\text{XOR}}$

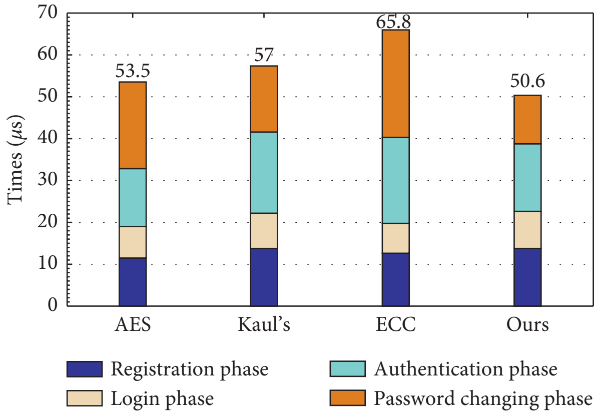

Figure 3: The efficiency comparison of four schemes.

based on ECC is test time 65.8 μs. ECC has high computational complexity in password changing phase, but it has good performance in registration and login phase. Our scheme computational complexity of test time is 50.6 μs, one of the lowest values of four schemes. Taking computational complexity as metric, we also proved the model of product authorization chain performs much better than the other three schemes during registration stage, login, and authentication stage.

5. Conclusions

In view of the limitation of Kaul and Awasthi's scheme in resisting offline password guessing attack, this paper retains other security advantages of Kaul and Awasthi's scheme and proposes a lightweight wireless two-way authentication scheme based on product authentication chain. This scheme is suitable for solving the problem of limited authentication calculation of IoT terminals. It can effectively avoid common attacks such as offline password guessing attack and man in the middle attack, establish a secure session key, modify the password freely, and ensure the security of Internet of things system. At the same time, in order to avoid strict clock synchronization of various devices in network, the continuously updated sending packet sequence number is used to ensure the validity of the message. According to the analysis of security, cost, and efficiency, the proposed scheme has higher security, less system overhead, lower computational cost, and higher operational efficiency and is more fit for resource-limited user intelligent terminal equipment.

Abbreviations

SK:	Session key
BAN:	Burrows–Abadi–Needham
SPI:	Security parameters index.

Data Availability

All the data in this study are from experimental data statistics.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

Hanqing Ding and Yifeng Yin conceptualized the study. Qing Zhang proposed the methodology. Weihua Liu provided the software. Qing Zhang and Yifeng Yin validated the study. Hanqing Ding was responsible for formal analysis. Qing Zhang and Yifeng Yin investigated the study. Qing Zhang was responsible for resources. Hanqing Ding and Yifeng Yin performed data curation. Yifeng Yin and Hanqing Ding prepared the original draft. Yifeng Yin and Hanqing Ding reviewed and edited the manuscript. Weihua Liu visualized the study. Yong Gan supervised the study. Yong Gan was responsible for project administration. Yong Gan was responsible for funding acquisition. All authors have read and agreed to the published version of the manuscript. Hanqing Ding, Qing Zhang, and Yong Gan contributed equally to this work.

Acknowledgments

The authors would like to acknowledge contributions by several colleagues at the CNIS (Xidian University). The authors are grateful to Baocang Wang and Fagen Li for their critical review of the manuscript. This research was funded by Natural Science Foundation of Henan Province of China under Grant no. 202300410508, National Natural Science Foundation of China under Grant nos. U1804263, 61572445, 61901418, and 61272038, Science and Technology Research Project of Henan Province under grant nos. 212102210563 and 202102210124, and Key Foundation of Science and Technology Development of Henan Province under Grant no. 142102210081.

References

[1] M. Ammar, G. Russello, and B. Crispo, "Internet of things: a survey on the security of IoT frameworks," *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2018.

[2] S. Hong, S. Park, L. W. Park et al., "An analysis of security systems for electronic information for establishing secure internet of things environments: focusing on research trends in the security field in South Korea," *Future Generation Computer Systems*, vol. 82, pp. 769–782, 2017.

[3] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, pp. 770–772, 1981.

[4] Y. Y. Wang, J. Y. Liu, F. X. Xiao, and J. Dan, "A more efficient and secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 32, no. 4, pp. 583–585, 2009.

[5] F. Wen and X. Li, "An improved dynamic ID-based remote user authentication with key agreement scheme," *Computers & Electrical Engineering*, vol. 38, no. 2, pp. 381–387, 2012.

[6] Y. F. Chang, W. L. Tai, and H. C. Chang, "Untraceable dynamic identity-based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3430–3440, 2014.

[7] S. Kumari, M. K. Khan, and X. Li, "An improved remote user authentication scheme with key agreement," *Computers & Electrical Engineering*, vol. 40, no. 6, pp. 1997–2012, 2014.

[8] S. D. Kaul and A. K. Awasthi, "Security enhancement of an improved remote user authentication scheme with key agreement," *Wireless Personal Communications*, vol. 89, no. 2, pp. 621–637, 2016.

[9] C. Y. Wang and G. A. Xu, "Cryptanalysis of three password-based remote user authentication schemes with non-tamper-resistant smart card," *Security and Communication Networks*, vol. 2017, Article ID 1619741, 14 pages, 2017.

[10] B. Liu, B. Yang, and X. Su, "An improved two-way security authentication protocol for RFID system," *Information*, vol. 9, no. 4, 2018.

[11] J. H. Wang, H. P. Liu, H. R. Shao et al., "Novel two-way security authentication wireless scheme based on hash function," *Computer Science*, vol. 43, no. 11, pp. 205–209, 2016.

[12] D. Wang, Z. Zhang, P. Wang et al., "Targeted online password guessing: an underestimated threat," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pp. 1242–1254, ACM, Vienna, Austria, October 2016.

[13] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 1–14, 2020.

[14] D. Wang and P. Wang, "Two birds with one stone: two-factor Authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, p. 1, 2016.

[15] L. Y. Y. Yifeng and X. Mingming, "A scalable lightweight grouping-proof protocol for RFID tags," *Modern Electronics Technique*, vol. 40, no. 17, pp. 86–90, 2017.

[16] J. L. Li, W. G. Zhang, S. Kumari et al., "Security analysis and improvement of a mutual authentication and key agreement solution for wireless sensor networks using chaotic maps," *Transactions on Emerging Telecommunications Technologies*, vol. 29, pp. 1–17, 2018.

[17] B. Mario, D. B. Alessandra, L. M. Erasmo, M. Antonino, and N. Mazzocca, "A PUF-based mutual authentication scheme for cloud-edges IoT systems," *Future Generation Computer Systems*, vol. 101, pp. 246–261, 2019.

[18] F. Xiong, R. Xiao, W. Ren, R. Zheng, and J. Jiang, "A key protection scheme based on secret sharing for blockchain-based construction supply chain system," *IEEE Access*, vol. 7, Article ID 126773, 2019.

[19] B. H. Shimaa, A. A. E. G. Mohamed, and H. Kim, "A decentralized lightweight authentication and privacy protocol for vehicular networks," *IEEE Access*, vol. 7, Article ID 119689, 2019.

[20] Y. Yin, K. Liu, C. Hu, and Y. Gan, "The group key agreement protocol based on multi-dimensional virtual permutation," *IEEE Communications Letters*, vol. 24, no. 12, pp. 2728–2732, 2020.

[21] M. Karuppiah, A. K. Das, X. Li et al., "Secure remote user mutual authentication scheme with key agreement for cloud environment," *Mobile Networks and Applications*, vol. 11, pp. 1–17, 2018.

[22] R. Y. Patil and R. S. Devane, "Formal verification of secure evidence collection protocol using BAN logic and AVISPA," *Procedia Computer Science*, vol. 167, pp. 1334–1344, 2020.