Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 1970798, 6 pages
https://doi.org/10.1155/2020/1970798

Research Article

Analysis and Improvement on an Authentication Protocol for IoT-Enabled Devices in Distributed Cloud Computing Environment

Baoyuan Kang, Yanbao Han, Kun Qian, and Jianqi Du

School of Computer Science and Technology, Tiangong University, Tianjin 300387, China

Correspondence should be addressed to Baoyuan Kang; baoyuankang@aliyun.com

Received 27 November 2019; Accepted 28 May 2020; Published 23 June 2020

Academic Editor: Haipeng Peng

Recently, a number of authentication protocols integrated with the Internet of Things (IoT) and cloud computing have been proposed for secure access control on large-scale IoT networks. In this paper, we carefully analyze Amin et al.'s authentication protocol for IoT-enabled devices in distributed cloud computing environment and find that Amin et al.'s protocol is vulnerable to several weaknesses. The main shortcoming of Amin et al.'s protocol is in authentication phase; a malicious cloud server can counterfeit the cloud server chosen by a user, and the control server cannot find this counterfeit. To overcome the shortcomings of Amin et al.'s protocol, we propose an improved protocol. In the registration phase of the improved protocol, the pseudoidentity and real identity of a user or a cloud server are bundled up with the control server's secret numbers. This measure can effectively prevent impersonation attack. We also compare the improved protocol with several existing authentication protocols in security and computational efficiency.

1. Introduction

With the development of the Internet technology, people's life and production have been greatly improved by Internet of Things (IoT) [1]. But, IoT also faces problems of efficiency due to its sensors with low memory and low power. Using powerful cloud services [2] can improve the efficiency of IoT. Now, authentication protocols integrated with the IoT and cloud computing attract people's attention.

The first suggested authentication protocol was proposed by Lamport [3]. Then, many password-based authentication protocols were proposed [4–8]. Recently, people discuss authentication protocols for multiserver in IoT and cloud environment [9–16]. Amin et al. [13] showed security vulnerabilities of two authentication protocols in multiserver cloud environment proposed by Xue et al. [11] and Chuang and Chen [12]. Then, Amin et al. [13] proposed an authentication protocol for IoT-enabled devices in distributed cloud computing environment. They claimed that the proposed protocol is protected against all possible security threats. However, in this paper, we find that Amin et al.'s protocol is vulnerable to several weaknesses. Firstly, during the registration phase of Amin et al.'s protocol, it is unreasonable for a user to register with a pseudoidentity. Secondly, the main shortcoming of Amin et al.'s protocol is that, in its authentication and key agreement phase, although the control server can identify a cloud server is legal, the control server cannot tell if this cloud server is the one chosen by a user. So, in Amin et al.'s protocol, a malicious server can counterfeit the server chosen by a user.

On the basis of analyzing the shortcomings of Amin et al.'s protocol, we propose an improvement on Amin et al.'s protocol. In the registration phase of the improved protocol, the pseudoidentity and real identity of a user or a cloud server are bundled up with the control server's secret numbers. This measure can effectively prevent impersonation attacks. We also compare the improved protocol with two existing protocols [11, 12] in security and computational efficiency.

The rest of the paper is organized as follows. In Section 2, we briefly review Amin et al.'s protocol and analyze its weaknesses. The improved protocol is proposed in Section 3. Security cryptanalysis and comparisons are given in Section 4. Finally, the article is concluded in Section 5.

2. Amin et al.'s Protocol and Its Weaknesses

This section briefly reviews the Amin et al.'s protocol [13] and shows its weaknesses. In Amin et al.'s protocol, there are three types of entity such as user U_i, service provider server S_m, and control server (CS). The CS is a trusted third party responsible for registration and authentication of users and service providing servers. The S_m provides set of services to U_i. The notations used in this article are recorded in Table 1.

Table 1: Notations table.

Symbol	Description
CS	The control server
S_m	mth cloud server
SID_m	Identity of the mth cloud server
d	Random number of S_m
U_i	ith user
ID_i	Identity of the user U_i
P_i	Password of the user U_i
b_1, b_2	Two random numbers of U_i
x, y	Secret numbers of CS
$h(\cdot)$	Hash function
T	Timestamp
$\oplus$	Bit-wise xor operation
$\|\|$	Concatenate operation

2.1. Amin et al.'s Protocol. Amin et al.'s protocol [13] contains five phases: registration, login, authentication and key agreement, password change, and identity update. For the sake of brevity, password change and identity update phases are not revised.

2.1.1. Registration Phase. During cloud server registration, the cloud server S_m sends (SID_m, d) to CS. After receiving it, the CS computes $\mathrm{PSID}_m = h(\mathrm{SID}_m \| d)$ and $\mathrm{BS}_m = h(\mathrm{PSID}_m \| y)$ and sends BS_m to S_m securely. Finally, S_m stores secret parameter (BS_m, d) into his memory.

During user registration, the user U_i computes $A_i = h(P_i \| b_1)$ and $\mathrm{PID}_i = h(\mathrm{ID}_i \| b_2) bb_i = b_2 \oplus A_i$ and sends (A_i, PID_i) to the CS securely. On getting (A_i, PID_i), the CS calculates $C_i = h(A_i \| \mathrm{PID}_i)$, $D_i = h(\mathrm{PID}_i \| x)$, and $E_i = D_i \oplus A_i$. Finally, the CS prepares and delivers a smartcard for each U_i after recording $(C_i, E_i, h(\cdot))$ in the smartcard and transports it to U_i through private communication. After getting it, U_i records (DP, bb_i) in the smartcard, where $\mathrm{DP} = h(\mathrm{ID}_i \| P_i) \oplus b_1$. Finally, the smartcard holds $(C_i, E_i, \mathrm{DP}, bb_i, h(\cdot))$.

2.1.2. Login Phase. For accessing server resources, a legal user U_i first punches the smartcard into card reader and inputs ID_i^* and P_i^* to the terminal. Then, the card reader calculates $b_1^* = \mathrm{DP} \oplus h(\mathrm{ID}_i^* \| P_i^*)$, $A_i^* = h(P_i^* \| b_i^*)$, $b_2^* = bb_i^* \oplus A_i^*$, $\mathrm{PID}_i^* = h(\mathrm{ID}_i^* \| b_2^*)$, and $C_i^* = h(A_i^* \| \mathrm{PID}_i^*)$. Then, the card reader checks the condition $C_i^* = C_i$. If $C_i^* = C_i$, it means that $\mathrm{ID}_i^* = \mathrm{ID}_i$ and $P_i^* = P_i$. The card reader produces a random number N_i and computes $D_i = E_i \oplus A_i$, $G_i = h(\mathrm{PID}_i \| \mathrm{SID}_m \| N_i \| \mathrm{TS}_i \| D_i)$, $F_i = D_i \oplus N_i$, and $Z_i = \mathrm{SID}_m \oplus h(D_i \| N_i)$, where SID_m is the cloud server's identity chosen by the user U_i. Then, the CR transmits the login messages $(G_i, F_i, Z_i, \mathrm{PID}_i, \mathrm{TS}_i)$ to S_m publicly.

2.1.3. Authentication and Key Agreement Phase. This phase is necessary for performing mutual authentication as well as key agreement among U_i, S_m, and CS. The detail explanation of this phase is as follows:

Step 1: the S_m first checks the condition whether $\mathrm{TS}_m - \mathrm{TS}_i < \Delta T$ holds or not on receiving the login message, where TS_m and ΔT are the cloud server's current timestamp and expected valid time interval for transmission delay, respectively. If the condition is not true, the S_m terminates the connection; otherwise, the S_m produces a random number N_m and computes $J_i = \mathrm{BS}_m \oplus N_m$ and $K_i = h(N_m \| \mathrm{BS}_m \| G_i \| \mathrm{TS}_m)$. Finally, S_m sends $(J_i, K_i, \mathrm{PSID}_m, G_i, F_i, Z_i, \mathrm{PID}_i, \mathrm{TS}_i, \mathrm{TS}_m)$ to the CS publicly.

Step 2: on getting messages from S_m, CS first checks the time interval, i.e., $\mathrm{TS}_{\mathrm{CS}} - \mathrm{TS}_m < \Delta T$, where $\mathrm{TS}_{\mathrm{CS}}$ and ΔT are the CS's current timestamp and expected valid time interval for transmission delay, respectively. If the verification holds, CS computes $D_i = h(\mathrm{PID}_i \| x)$, $N_i^* = F_i \oplus D_i$, $\mathrm{SID}_m^* = Z_i \oplus h(D_i \| N_i^*)$, and $G_i^* = h(\mathrm{PID}_i \| \mathrm{SID}_m^* \| N_i^* \| \mathrm{TS}_i \| D_i)$. After that, the CS checks the condition $G_i^* = G_i$. If $G_i^* = G_i$, the CS thinks that the U_i is legal; otherwise, the procedures are terminated. After that, the CS computes $\mathrm{BS}_m^* = h(\mathrm{PSID}_m \| y)$, $N_m^* = \mathrm{BS}_m^* \oplus J_i$, and $K_i^* = h(\mathrm{BS}_m^* \| N_m^* \| G_i \| \mathrm{TS}_m)$. Again, the CS checks the condition $K_i^* = K_i$. If $K_i^* = K_i$, the CS thinks that S_m is legal; otherwise, the procedure is terminated. After that, the CS chooses a random number N_{CS} and computes $P_{\mathrm{CS}} = N_m \oplus N_{\mathrm{CS}} \oplus h(N_i \| D_i)$, $R_{\mathrm{CS}} = N_i \oplus N_{\mathrm{CS}} \oplus h(BS_m^* \| N_m^*)$, $\mathrm{SK}_{\mathrm{CS}} = h(N_i \oplus N_m \oplus N_{\mathrm{CS}})$, $Q_{\mathrm{CS}} = h((N_m \oplus N_{\mathrm{CS}}) \| \mathrm{SK}_{\mathrm{CS}})$, and $V_{\mathrm{CS}} = h((N_i \oplus N_{\mathrm{CS}}) \| \mathrm{SK}_{\mathrm{CS}})$, where $\mathrm{SK}_{\mathrm{CS}}$ is the secret session key. Finally, the CS sends $(P_{\mathrm{CS}}, R_{\mathrm{CS}}, Q_{\mathrm{CS}}, V_{\mathrm{CS}})$ to S_m for achieving mutual authentication of the protocol through public communication.

Step 3: on getting reply messages from CS, S_m computes $W_m = h(BS_m \| N_m)$, $N_i \oplus N_{CS} = R_{\mathrm{CS}} \oplus W_m$, $SK_m = h(N_i \oplus N_{\mathrm{CS}} \oplus N_m)$, and $V_{\mathrm{CS}}^* = h((N_i \oplus N_{\mathrm{CS}}) \| \mathrm{SK}_m)$. Then, the S_m checks the condition $V_{\mathrm{CS}}^* = V_{\mathrm{CS}}$ or not. If $V_{\mathrm{CS}}^* = V_{\mathrm{CS}}$, the session is terminated; otherwise, messages $(P_{\mathrm{CS}}, Q_{\mathrm{CS}})$ are sent to the U_i publicly.

Step 4: on obtaining messages from S_m, the U_i calculates $L_i = h(N_i \| D_i)$, $N_m \oplus N_{\mathrm{CS}} = P_{\mathrm{CS}} \oplus L_i$, $\mathrm{SK}_i = h(N_m \oplus N_{\mathrm{CS}} \oplus N_i)$, and $Q_{\mathrm{CS}}^* = h((N_m \oplus N_{\mathrm{CS}}) \| \mathrm{SK}_i)$. Then, the U_i checks the condition $Q_{\mathrm{CS}}^* = Q_{\mathrm{CS}}$, and if $Q_{\mathrm{CS}}^* = Q_{\mathrm{CS}}$, it

proves the authenticity of S_m and CS. Finally, the proposed protocol achieves mutual authentication among U_i, S_m, and CS. Now, the U_i and the S_m can exchange their secret information securely using $\mathrm{SK}_m = \mathrm{SK}_i$.

2.2. The Weaknesses of Amin et al.'s Protocol. This section shows that Amin et al.'s protocol [13] has some security drawbacks.

2.2.1. Weaknesses in User Registration Phase. During registration in CS, the user U_i sends (A_i, PID_i) to the CS. But, $A_i = h(P_i \| b_i)$ and $\mathrm{PID}_i = h(\mathrm{ID}_i \| b_2)$.

PID_i is just a pseudoidentity. It is unreasonable for a user to register with a pseudoidentity.

2.2.2. Weaknesses in Authentication and Key Agreement Phase. In authentication and key agreement phase, when the CS receives messages $(J_i, K_i, \mathrm{PSID}_m, G_i, F_i, Z_i, \mathrm{PID}_i, \mathrm{TS}_i, \mathrm{TS}_m)$ from the cloud server S_m, although CS can know the identity SID_m of the server chosen by the user from following calculation,

$$\begin{aligned} D_i &= h\left(\mathrm{PID}_i \| x\right), \\ N_i &= F_i \oplus D_i, \\ \mathrm{SID}_m &= Z_i \oplus h\left(D_i \| N_i\right), \\ G_i^* &= h\left(\mathrm{PID}_i \| \mathrm{SID}_m \| N_i \| \mathrm{TS}_i \| D_i\right), \end{aligned} \tag{1}$$

and verifying $G_i^* = G_i$.

CS also can know the server with pseudoidentity PSID_m, and the secret value BS_m is a legal server by following calculation:

$$\begin{aligned} \mathrm{BS}_m &= h\left(\mathrm{PSID}_m \| y\right), \\ N_m &= \mathrm{BS}_m \oplus J_i, \\ K_i^* &= h\left(\mathrm{BS}_m \| N_m \| G_i \| \mathrm{TS}_m\right), \end{aligned} \tag{2}$$

and verifying $K_i^* = K_i$.

But, the CS cannot tell if the server with pseudoidentity PSID_m and the secret value BS_m is the one the user chose with real identity SID_m.

Due to the above weaknesses, a malicious server can counterfeit the server chosen by the user, and the CS cannot see through him.

2.2.3. Puzzling Question of the User. Due to the weaknesses in Section 2.2.2, the user cannot be convinced that the session key SK_i is shared with his chosen server.

3. The Improved Protocol

To overcome the shortcomings of Amin et al.'s protocol, in this section, an improved protocol is proposed. Also, for the sake of brevity, only the registration, login, and authentication key agreement phases are described.

3.1. Registration Phase. Suppose the control server CS is a trusted third party responsible for registration and authentication of users and cloud servers. CS chooses two random secret numbers x and y.

In registration phase, any cloud server and user can register with CS. When one cloud server S_m wants register with CS, it chooses its identity SID_m and a random number d. Then, it sends (SID_m, d) to the control server CS. After CS receives (SID_m, d), CS computes

$$\begin{aligned} \mathrm{PSID}_m &= h\left(\mathrm{SID}_m \| d\right), \\ \mathrm{BS}_m &= h\left(\mathrm{PSID}_m \| \mathrm{SID}_m \| y\right), \end{aligned} \tag{3}$$

and sends BS_m to the cloud server S_m through the secure channel. Once S_m receives BS_m, S_m stores secret parameters (BS_m, d).

When one user U_i registers with CS, U_i chooses his identity ID_i and password P_i. Then, U_i calculates $A_i = P_i \oplus h(B_i)$. Here, B_i is his biometric. Finally, U_i submits (ID_i, A_i) to the CS through the secure channel. On receiving the message, CS chooses a random number b_i and computes

$$\begin{aligned} \mathrm{PID}_i &= h\left(\mathrm{ID}_i \| b_i\right), \\ C_i &= h\left(\mathrm{ID}_i \| A_i\right), \\ D_i &= h\left(\mathrm{PID}_i \| x\right), \\ E_i &= D_i \oplus A_i, \\ \Delta_i &= h\left(\mathrm{PID}_i \| \mathrm{ID}_i \| x\right), \\ \Omega_i &= b_i \oplus A_i, \end{aligned} \tag{4}$$

and issues a smart card containing the information $(C_i, \Omega_i, \Delta_i, E_i, h(\cdot))$ to the user U_i.

3.2. Login Phase. After punching his smart card, a user U_i provides ID_i^*, P_i^*, and B_i^* to the card reader. The card reader computes

$$\begin{aligned} A_i^* &= P_i^* \oplus h\left(B_i^*\right), \\ C_i^* &= h\left(\mathrm{ID}_i^* \| A_i^*\right). \end{aligned} \tag{5}$$

Then, the card reader checks whether $C_i^* = C_i$ or not. When $C_i^* = C_i$, $\mathrm{ID}_i^* = \mathrm{ID}_i$, $P_i^* = P_i$, and $B_i^* = B_i$, $A_i^* = A_i$. Then, the card reader chooses a random number N_i and computes

$$\begin{aligned} b_i &= \Omega_i \oplus A_i, \\ \mathrm{PID}_i &= h\left(\mathrm{ID}_i \| b_i\right), \\ D_i &= E_i \oplus A_i, \\ O_i &= \mathrm{ID}_i \oplus D_i, \\ G_i &= h\left(\mathrm{ID}_i \| \mathrm{SID}_m \| N_i \| \mathrm{TS}_i \| D_i\right), \\ F_i &= \Delta_i \oplus N_i, \\ Z_i &= \mathrm{SID}_m \oplus h\left(D_i \| N_i\right), \end{aligned} \tag{6}$$

where SID_m is the identity of the cloud server S_m chosen by the user U_i. Then, the card reader sends the login messages $(G_i, F_i, Z_i, O_i, \mathrm{PID_i}, \mathrm{TS}_i)$ to the cloud server S_m publicly. TS_i is the U_i's current timestamp.

3.3. Authentication Key Agreement Phase. This phase includes four steps. It is also illustrated in Figure 1.

Step 1: once S_m receives the login message, S_m checks the condition whether $\mathrm{TS}_m - \mathrm{TS}_i < \Delta T$ holds or not. If the condition is true, S_m chooses a random number N_m and computes

$$\begin{aligned} J_i &= \mathrm{BS}_m \oplus N_m, \\ K_i &= h\left(N_m \| \mathrm{BS}_m \| \mathrm{PID}_i \| G_i \| \mathrm{TS}_m\right). \end{aligned} \tag{7}$$

Then, the S_m submits $(J_i, K_i, \mathrm{PSID}_m, G_i, F_i, Z_i, O_i, \mathrm{PID}_i, \mathrm{TS}_i, \mathrm{TS}_m)$ to the CS. Here, TS_m and ΔT are the cloud server's current timestamp and expected time interval for transmission delay, respectively.

Step 2: on receiving the messages from S_m, CS first checks whether $\mathrm{TS_{CS}} - \mathrm{TS}_m < \Delta T$ holds or not, where $\mathrm{TS_{CS}}$ and ΔT are the similar meanings mentioned before. If the verification holds, CS calculates

$$\begin{aligned} D_i &= h\left(\mathrm{PID}_i \| x\right), \\ \mathrm{ID}_i &= O_i \oplus D_i, \\ N_i &= F_i \oplus h\left(\mathrm{PID}_i \| \mathrm{ID}_i \| x\right), \\ \mathrm{SID}_m &= Z_i \oplus h\left(D_i \| N_i\right), \\ G_i^* &= h\left(\mathrm{ID}_i \| \mathrm{SID}_m \| N_i \| \mathrm{TS}_i \| D_i\right). \end{aligned} \tag{8}$$

Then, the CS checks whether $G_i^* = G_i$ holds or not. If $G_i^* = G_i$, the CS believes the U_i with real identity ID_i is legal. Then, the CS computes

$$\begin{aligned} \mathrm{BS}_m &= h\left(\mathrm{PSID}_m \| \mathrm{SID}_m \| y\right), \\ N_m &= \mathrm{BS}_m \oplus J_i, \\ K_i^* &= h\left(N_m \| \mathrm{BS}_m \| \mathrm{PID}_i \| G_i \| \mathrm{TS}_m\right). \end{aligned} \tag{9}$$

Then, the CS checks the condition $K_i^* = K_i$. If $K_i^* = K_i$, the CS believes S_m with real SID_m is legal and chosen by the user U_i. Then, the CS produces a random number N_{CS} and computes

$$\begin{aligned} P_{\mathrm{CS}} &= N_m \oplus N_{\mathrm{CS}} \oplus h\left(N_i \| D_i \| F_i\right), \\ R_{\mathrm{CS}} &= N_i \oplus N_{\mathrm{CS}} \oplus h\left(\mathrm{BS}_m \| N_m\right), \\ \mathrm{SK_{CS}} &= h\left(N_i \oplus N_m \oplus N_{\mathrm{CS}}\right), \\ Q_{\mathrm{CS}} &= h\left(\left(N_m \oplus N_{\mathrm{CS}}\right) \| \mathrm{SK_{CS}}\right), \\ V_{\mathrm{CS}} &= h\left(\left(N_i \oplus N_{\mathrm{CS}}\right) \| \mathrm{SK_{CS}}\right). \end{aligned} \tag{10}$$

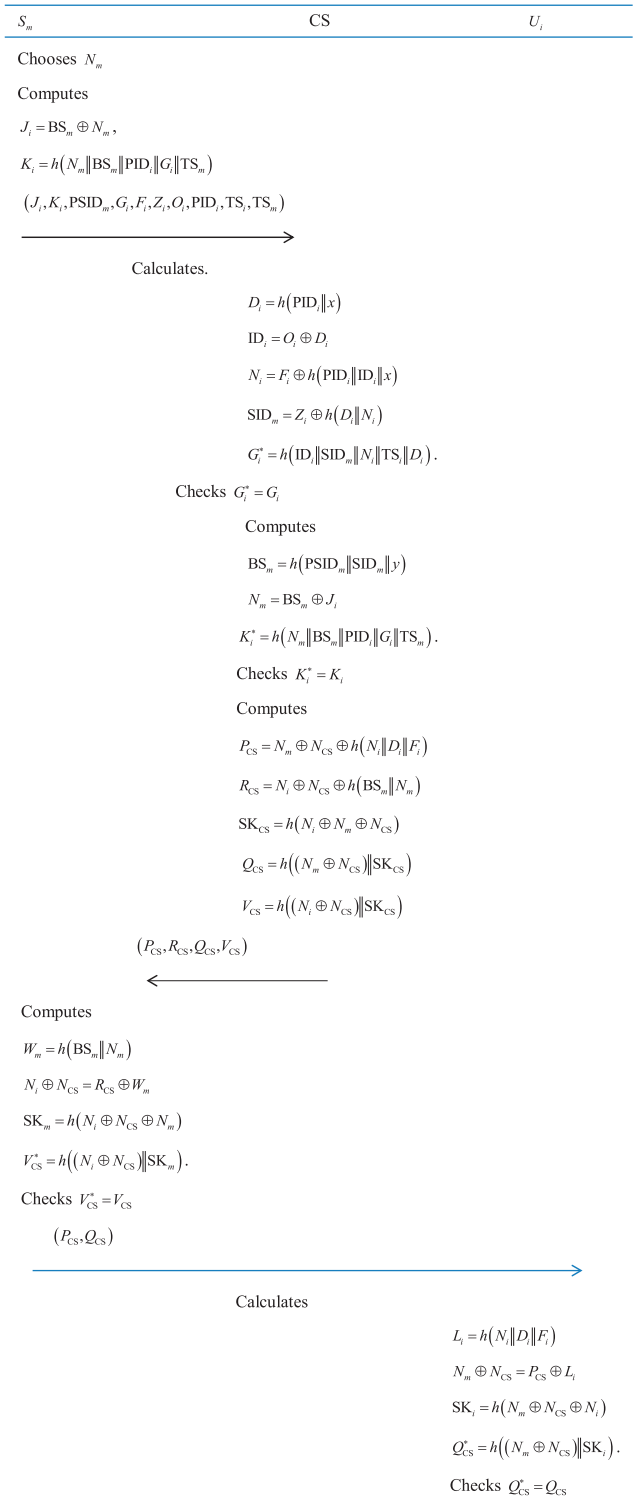

Figure 1: Authentication key agreement phase.

Then, the CS sends $(P_{\mathrm{CS}}, R_{\mathrm{CS}}, Q_{\mathrm{CS}}, V_{\mathrm{CS}})$ to the S_m publicly.

Step 3: on receiving the reply messages from CS, S_m computes

$$\begin{aligned} W_m &= h(\mathrm{BS}_m \| N_m), \\ N_i \oplus N_{\mathrm{CS}} &= R_{\mathrm{CS}} \oplus W_m, \\ \mathrm{SK}_m &= h(N_i \oplus N_{\mathrm{CS}} \oplus N_m), \\ V^*_{\mathrm{CS}} &= h((N_i \oplus N_{\mathrm{CS}}) \| \mathrm{SK}_m). \end{aligned} \tag{11}$$

Then, the S_m checks the condition $V^*_{\mathrm{CS}} = V_{\mathrm{CS}}$. If $V^*_{\mathrm{CS}} = V_{\mathrm{CS}}$, S_m sends messages $(P_{\mathrm{CS}}, Q_{\mathrm{CS}})$ to the U_i publicly.

Step 4: on receiving messages from S_m, the U_i calculates

$$\begin{aligned} L_i &= h(N_i \| D_i \| F_i), \\ N_m \oplus N_{\mathrm{CS}} &= P_{\mathrm{CS}} \oplus L_i, \\ \mathrm{SK}_i &= h(N_m \oplus N_{\mathrm{CS}} \oplus N_i), \\ Q^*_{\mathrm{CS}} &= h((N_m \oplus N_{\mathrm{CS}}) \| \mathrm{SK}_i). \end{aligned} \tag{12}$$

Next, the U_i checks whether $Q^*_{\mathrm{CS}} = Q_{\mathrm{CS}}$. If $Q^*_{\mathrm{CS}} = Q_{\mathrm{CS}}$, U_i believes the authenticity of S_m and CS and shares a session key $\mathrm{SK}_i (= \mathrm{SK}_m)$ with the cloud server S_m.

4. Security Analysis and Comparisons

4.1. Security Analysis. This section shows that the improved protocol is well protected against relevant security threats. Firstly, like Amin et al.'s protocol [13], the improved protocol is user anonymous and protected against password guessing attack, replay attack, insider attack, and session key discloser attack. For the shortcomings of Amin et al.'s protocol, the following analysis is focused on the improved protocol against impersonation attack.

In cloud server registration phase of the improved protocol, the cloud server S_m with identity SID_m and pseudoidentity PSID_m has secret value

$$\mathrm{BS}_m = h(\mathrm{PSID}_m \| \mathrm{SID}_m \| y), \tag{13}$$

computed by the control server CS. So, in the authentication phase, if one cloud server S^*_m not chosen by the user U_i counterfeits S_m, S^*_m intercepts the login messages $(G_i, F_i, Z_i, O_i, \mathrm{PID}_i, \mathrm{TS}_i)$ from U_i and computes

$$\begin{aligned} J_i &= \mathrm{BS}^*_m \oplus N^*_m, \\ K_i &= h(N_m \| \mathrm{BS}^*_m \| \mathrm{PID}_i \| G_i \| \mathrm{TS}^*_m), \end{aligned} \tag{14}$$

where $\mathrm{BS}^*_m = h(\mathrm{PSID}^*_m \| \mathrm{SID}^*_m \| y)$. Then, S^*_m sends

$$(J_i, K_i, \mathrm{PSID}^*_m, G_i, F_i, Z_i, O_i, \mathrm{PID}_i, \mathrm{TS}_i, \mathrm{TS}^*_m), \tag{15}$$

to the CS publicly. But, CS obtains the identity SID_m of the cloud server S_m chosen by the user U_i from Z_i and computes

$$\begin{aligned} \overline{\mathrm{BS}}^*_m &= h(\mathrm{PSID}^*_m \| \mathrm{SID}_m \| y), \\ N^*_m &= \overline{\mathrm{BS}}^*_m \oplus J_i, \\ K^*_i &= h(N^*_m \| \overline{\mathrm{BS}}^*_m \| \mathrm{PID}_i \| G_i \| \mathrm{TS}^*_m). \end{aligned} \tag{16}$$

Obviously, due to $\mathrm{BS}^*_m \neq \overline{\mathrm{BS}}^*_m$, then $K^*_i \neq K_i$. So, S^*_m cannot pass the CS's verification.

Table 2: Comparison of security.

	F1	F2	F3	F4	F5	F6	F7
Xue et al. [11]	Yes	No	Yes	Yes	Yes	Yes	Yes
Amin et al. [13]	Yes	Yes	Yes	Yes	Yes	No	No
Improved protocol	Yes	Yes	Yes	Yes	Yes	Yes	Yes

F1: user anonymity; F2: resist password guessing attack; F3: resist replay attack; F4: resist insider attack; F5: resist session key discloser attack; F6: resist impersonation attack; F7: CS knows the real identities of users and cloud servers.

Table 3: Comparison of computation costs.

	P1	P2	P3	P4
Xue et al. [11]	7H + 2X	5H + 5X	25H + 25X	37H + 32X
Improved protocol	7H + 3X	5H + 6X	18H + 21X	30H + 30X

P1: registration phase; P2: login phase; P3: authentication phase; P4: total computation cost; H: hash computation and its time cost; X: xor operation and its time cost.

If S^*_m wants to tamper Z_i by computing $Z^*_i = \mathrm{SID}^*_m \oplus h(D^*_i \| N^*_i)$, since $D^*_i \neq D_i$ and $N^*_i \neq N_i$, S^*_m also cannot pass the CS's verification.

Therefore, the improved protocol is protected against cloud server impersonation attack.

In Amin et al.'s protocol, CS does not know the real identities of the user. But, in improved protocol, we use O_i to show the real identities of the user. Also, $\Delta_i = h(\mathrm{PID}_i \| \mathrm{ID}_i \| x)$ is used in the improved protocol, and the user cannot pass CS's verification if he uses false identity.

In summary, the improved protocol completely overcomes the shortcomings of Amin et al.'s protocol. In the improved protocol, neither the user nor the cloud server can launch impersonation attacks. In the improved protocol, the user and the cloud server can use the shared session key between them with trust.

4.2. Comparisons. In this section, the comparison of the improved protocol with other protocols [11,13] is shown. The comparison results of the security features and computation costs are shown, respectively, in Tables 2 and 3.

From Table 2, the improved protocol is superior to the protocols [11,13] in terms of security. Furthermore, in Table 3, the comparison of computation costs is shown between the improved protocol and the relatively good protocol [11]. From Table 3, the total computation cost of the protocol [11] is 37H + 32X, but the total computational cost of the improved scheme is 30H + 30X. The computation cost of the improved protocol is significantly less than the protocol [11].

5. Conclusion

In this paper, we find that Amin et al.'s authentication protocol is vulnerable to several weaknesses. To overcome the shortcomings of Amin et al.'s protocol, we propose an improved protocol. We also compare the improved protocol with several existing authentication protocols in security and computational efficiency. The improved protocol not only

completely overcomes the shortcomings of Amin et al.'s protocol but also has less computation cost.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Applied Basic and Advanced Technology Research Programs of Tianjin (no. 15JCYBJC15900) and the National Natural Science Foundation of China (no. 61972456).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] G. Ateniese, R. Burns, R. Curtmola et al., "Provable data possession at untrusted stores,," in *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07)*, pp. 598–609, Virginia, VA, USA, November 2007.

[3] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[4] A. K. Awashti and S. Lal, "An enhanced remote user authentication scheme using smart cards," *IEEE Transaction on Consumer Electronics*, vol. 50, no. 2, pp. 583–586, 2004.

[5] M. K. Khan, S. Kumari, and M. K. Gupta, "More efficient key-hash based fingerprint remote authentication scheme using mobile device," *Computing*, vol. 96, no. 9, pp. 793–816, 2013.

[6] M. Kumar, M. K. Gupta, and S. Kumari, "An improved efficient remote password authentication scheme with smartcard over insecure networks," *International Journal of Network Security*, vol. 13, no. 3, pp. 167–177, 2011.

[7] B. Kang, J. Han, and Q. Wang, "Cryptanalysis and improvement on an IC-card-based remote login mechanism," in *Proceedings of the 2010 2nd International Conference on Computer Engineering and Technology*, vol. 1, pp. 65–68, Bali Island, Indonesia, April 2010.

[8] B. Kang and J. Han, "Cryptanalysis and improvement on three-party protocols for password authenticated key exchange," in *Proceedings of the 2010 2nd International Conference on Education Technology and Computer*, vol. 5, pp. 5197–5201, Shanghai, China, June 2010.

[9] S. K. Sood, A. K. Sarje, and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 609–618, 2011.

[10] X. Li, Y. Xiong, J. Ma, and W. Wang, "An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 763–769, 2012.

[11] K. Xue, P. Hong, and C. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 195–206, 2014.

[12] M.-C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1411–1418, 2014.

[13] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed Cloud Computing environment," *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018.

[14] L. Zhou, X. Li, K. Yeh, C. Su, and W. Chiu, "Lightweight IoT-based authentication scheme in cloud computing circumstance," *Future Generation Computer Systems*, vol. 91, pp. 1244–1251, 2019.

[15] K.-H. Wang, C.-M. Chen, W. Fang, and T.-Y. Wu, "A secure authentication scheme for internet of things," *Pervasive and Mobile Computing*, vol. 42, pp. 15–26, 2017.

[16] O. Ruan, N. Kumar, D. He, and J.-H. Lee, "Efficient provably secure password-based explicit authenticated key agreement," *Pervasive and Mobile Computing*, vol. 24, pp. 50–60, 2015.