sensors

MDPI

Article

A Secure Lightweight Three-Factor Authentication Scheme for IoT in Cloud Computing Environment

SungJin Yu, KiSung Park and YoungHo Park *

School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea
* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-7842

Received: 1 July 2019; Accepted: 16 August 2019; Published: 19 August 2019

Abstract: With the development of cloud computing and communication technology, users can access the internet of things (IoT) services provided in various environments, including smart home, smart factory, and smart healthcare. However, a user is insecure various types of attacks, because sensitive information is often transmitted via an open channel. Therefore, secure authentication schemes are essential to provide IoT services for legal users. In 2019, Pelaez et al. presented a lightweight IoT-based authentication scheme in cloud computing environment. However, we prove that Pelaez et al.'s scheme cannot prevent various types of attacks such as impersonation, session key disclosure, and replay attacks and cannot provide mutual authentication and anonymity. In this paper, we present a secure and lightweight three-factor authentication scheme for IoT in cloud computing environment to resolve these security problems. The proposed scheme can withstand various attacks and provide secure mutual authentication and anonymity by utilizing secret parameters and biometric. We also show that our scheme achieves secure mutual authentication using Burrows–Abadi–Needham logic analysis. Furthermore, we demonstrate that our scheme resists replay and man-in-the-middle attacks usingthe automated validation of internet security protocols and applications (AVISPA) simulation tool. Finally, we compare the performance and the security features of the proposed scheme with some existing schemes. Consequently, we provide better safety and efficiency than related schemes and the proposed scheme is suitable for practical IoT-based cloud computing environment.

Keywords: authentication; cloud computing; internet of things; formal security analysis; Burrows–Abadi–Needham logic; AVISPA

1. Introduction

With the recent advances in wireless sensor networks and embedded technologies, internet of things (IoT) connects objects and shares various useful data with internet through resource-constrained devices to provide convenient services for users such as smart home, healthcare, vehicle to everything and smart gird. However, a single server environment also is inefficient for IoT because an ocean of data is generated by resource-constrained devices such as microsensor, RFID tag and smart cards.

Cloud computing is a distributed computing mechanism for a large-scale data and allows sharing resources among all of the servers and users. The cloud computing provides five essential characteristics: *on-demand self-services*, *ubiquitous network access*, *rapid elasticity*, *measured service* and *resource pooling* [1,2]. *On-demand self-service* handles cloud services without human interaction and *ubiquitous network access* controls access service using standard protocols. *Rapid elasticity* and *measured service* optimize the resource usage. *Resource pooling* provides cloud service using homogeneous infrastructure among service users. The cloud computing deals with an ocean of data generated by devices and sensors and provides data managing service for users through these essential characteristics.

However, these services are vulnerable to potential attacks by malicious adversaries because they are provided through an open channel, including sensitive data of legitimate user about location, health, payment, etc. Therefore, a secure and efficient authentication for IoT environment has become essential security requirements to provide useful services to user.

In 1981, Lamport [3] proposed one factor user authentication scheme using passwords to ensure user's privacy. However, security of the password based authentication scheme is easily broken because its security only relies on the passwords. In 2002, Chien et al. proposed two factor authentication scheme to overcome this security flaw using password and smart cards. However, their scheme is vulnerable to smart card stolen attack as the data stored in smart cards can be extracted by power analysis attacks [4]. When a malicious adversary obtains smart cards and password, they can perform various attacks such as impersonation, replay and insider attacks. To overcome the above-mentioned security weaknesses, three-factor authentication schemes have been proposed [5–7]. Biometrics (e.g., face, retina, fingerprint, iris, etc.) have several important characteristics: they cannot be lost or forgotten; they are hard to forge, copy, share or distribute; and they are difficult to guess.

In 2019, Pelaez et al. [8] demonstrated that the previous scheme is vulnerable to insider, off-line guessing and disclosure attacks and proposed enhanced IoT-based authentication scheme in cloud computing environment. This paper demonstrates that Pelaez et al.'s scheme does not withstand impersonation, session key disclosure and replay attacks. We also show that their scheme does not achieve secure mutual authentication and anonymity. Moreover, we propose a secure and lightweight three-factor authentication scheme for IoT in cloud computing environment to resolve these security weaknesses, considering computational costs.

1.1. Adversary Model

We present the Dolev–Yao (DY) model [9] to evaluate security of ours and previous schemes, which is widely accepted as security threat model. The detailed description of the DY model is as below:

- A malicious adversary can modify, intercept, delete or insert the transmitted messages via an open channel. A malicious adversary can obtain or steal the smart card of legitimate user and can extract the data stored in the smart card by using power-analysis [4].
- A malicious adversary can perform various attacks such as man-in-the-middle (MITM), replay, impersonation, and session key disclosure attack [10,11].

1.2. Our Contributions

Our contributions in this paper are as follows.

- We demonstrate that Pelaez et al.'s scheme is not secure against various attacks such as impersonation, session key disclosure and replay attacks and does not achieve secure mutual authentication and anonymity.
- We propose a secure and lightweight three-factor authentication scheme for IoT in cloud computing environment to address the security shortcomings of Pelaez et al.'s scheme. The proposed scheme withstands impersonation, session key disclosure, and replay attacks and achieve secure mutual authentication and anonymity. Moreover, the proposed scheme is more efficient than Pelaez et al.'s scheme because it utilizes only bitwise exclusive or (XOR) and hash operations.
- We prove that the proposed scheme provides secure mutual authentication using the Burrows–Abadi–Needham (BAN) logic [12] and perform an informal security analysis to prove that our scheme is secure against various attacks such as MITM, impersonation, replay and session key disclosure attacks. Furthermore, we compare the security properties and performance of proposed protocol with other related schemes.
- We perform a formal security analysis using the automated validation of internet security protocols and applications (AVISPA) simulation tool to prove that the proposed protocol resists the MITM and replay attacks.

1.3. Organization

We introduce the related works and review Pelaez et al.'s scheme in Sections 2 and 3. In Sections 4 and 5, we cryptanalyze Pelaez et al.'s scheme and propose a lightweight IoT-based three-factor authentication scheme in cloud computing environment to enhance the security shortcomings of Pelaez et al.'s scheme. Sections 6 and 7 prove the security of proposed scheme and present the simulation analysis using AVISPA. In Section 8, we compare the security properties and performances of proposed protocol with other related schemes. Finally, Section 9 concludes the paper.

2. Related Works

In last few decades, numerous authentication and key agreement schemes have been proposed to ensure privacy of user, considering resource-constrained environments such as wireless sensor networks, global mobility networks and vehicular networks [3,13–19]. In 1981, Lamport [3] firstly proposed a lightweight password based user authentication scheme to provide secure communication. However, Lamport's scheme has low security level because its security only relies on passwords. In 2002, Chien et al. [13] presented a two-factor user authentication protocol using smart card and password to resolve this problem. Unfortunately, the two-factor authentication schemes using password and smart cards cannot ensure user's privacy [13–19], when the data stored in token (e.g., smart card, mobile device, etc.) are compromised.

Later, several authentication and key agreement schemes for IoT have been presented in various fields [20–22]. However, these environments are not suitable for IoT because it cannot handle a large number of data. In 2019, Zhou et al. [23] presented a lightweight IoT-based authentication scheme in cloud computing environment to overcome this issue. Zhou et al. claimed that their scheme can prevent various attacks such as insider, forgery and tracking attacks and provide secure mutual authentication and session key security. However, in 2019, Pelaez et al. [8] pointed out that Zhou et al.'s scheme [23] cannot withstand insider, off-line guessing and session key disclosure attacks and provide secure mutual authentication. To resolve these security problems, Pelaez et al. [8] presented a lightweight IoT-based authentication scheme in cloud computing environment. They also claimed that their scheme is secure against off-line password guessing, insider, impersonation and replay attacks.

3. Review of Pelaez et al.'s Scheme

We briefly review Pelaez et al.'s IoT based authentication scheme in cloud computing environment. Their scheme comprises of three processes: registration, authentication, and password change. These processes are presented as below (for details, see [8]).

3.1. User Registration Process

In Pelaez et al.'s scheme, a new user U_i is registered from control server CS via a secure channel. Figure 1 shows the user registration process of Pelaez et al.'s scheme. In Figure 1, U_i sends the registration request to CS and then CS issues the smart cards.

3.2. Cloud Server Registration Process

In Pelaez et al.'s scheme, a cloud server S_j is registered from control server CS via a secure channel. Figure 2 shows the cloud server registration process of the Pelaez et al.'s scheme. In Figure 2, S_j sends the registration request to CS and then CS sends parameters B_2 and B_3 to S_j.

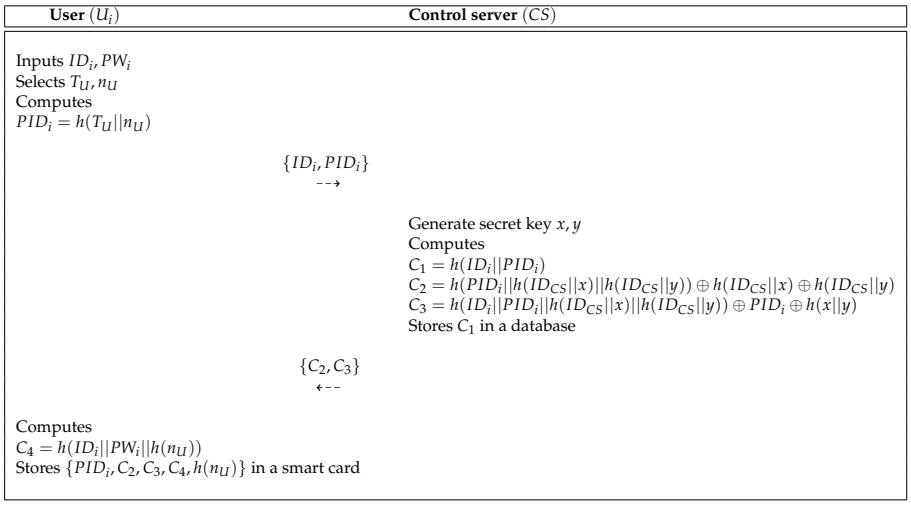

Figure 1. User registration process of the Pelaez et al.'s scheme [8].

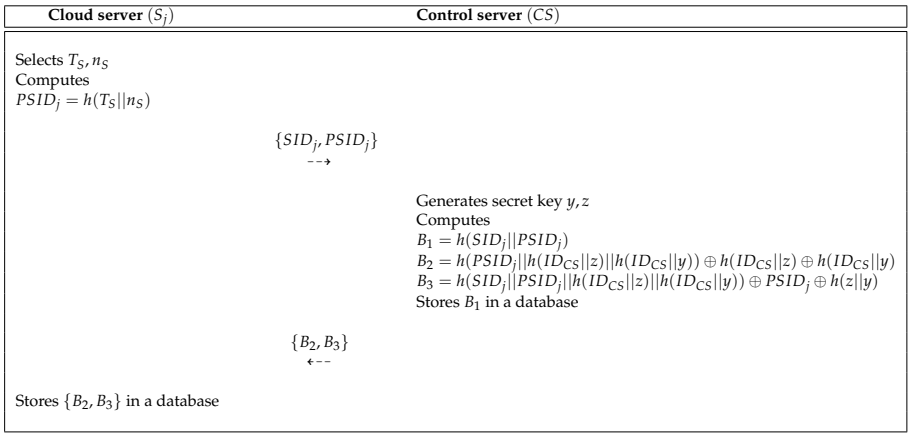

Figure 2. Cloud server registration process of the Pelaez et al.'s scheme [8].

3.3. Login Process

When U_i wants to access the service, U_i firstly sends login request message to S_j. In Figure 3, U_i sends login request messages $\{T_U^{new}, D_1, PID_i, D_2\}$ to S_j, and then S_j sends the messages $\{T_U^{new}, D_1, PID_i, D_2, T_S^{new}, D_3, PSID_j, D_4, D_5\}$ to CS in order to check validation of U_i.

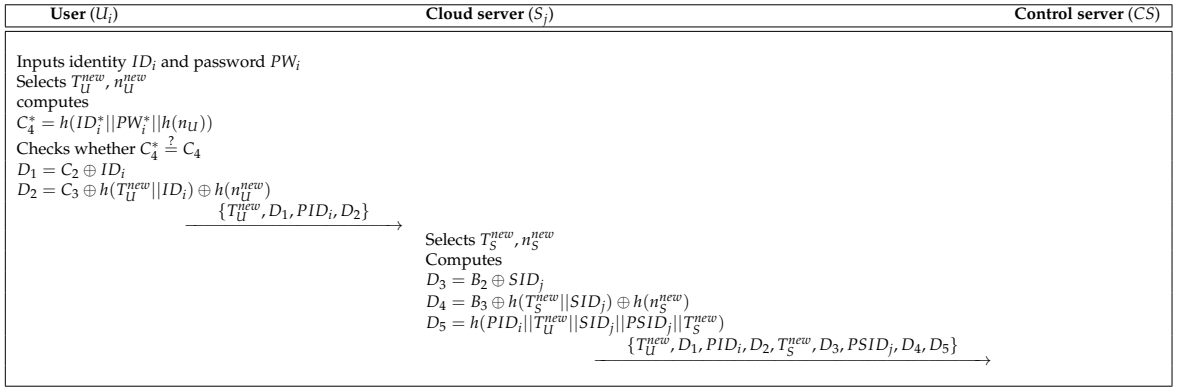

Figure 3. Login process of the Pelaez et al.'s scheme [8].

3.4. Authentication Process

After finishing the login process, U_i, S_j and CS perform mutual authentication with each entity, and then U_i and S_j can share the session key SK_{U-S}. Figure 4 shows the authentication process of the Pelaez et al.'s scheme.

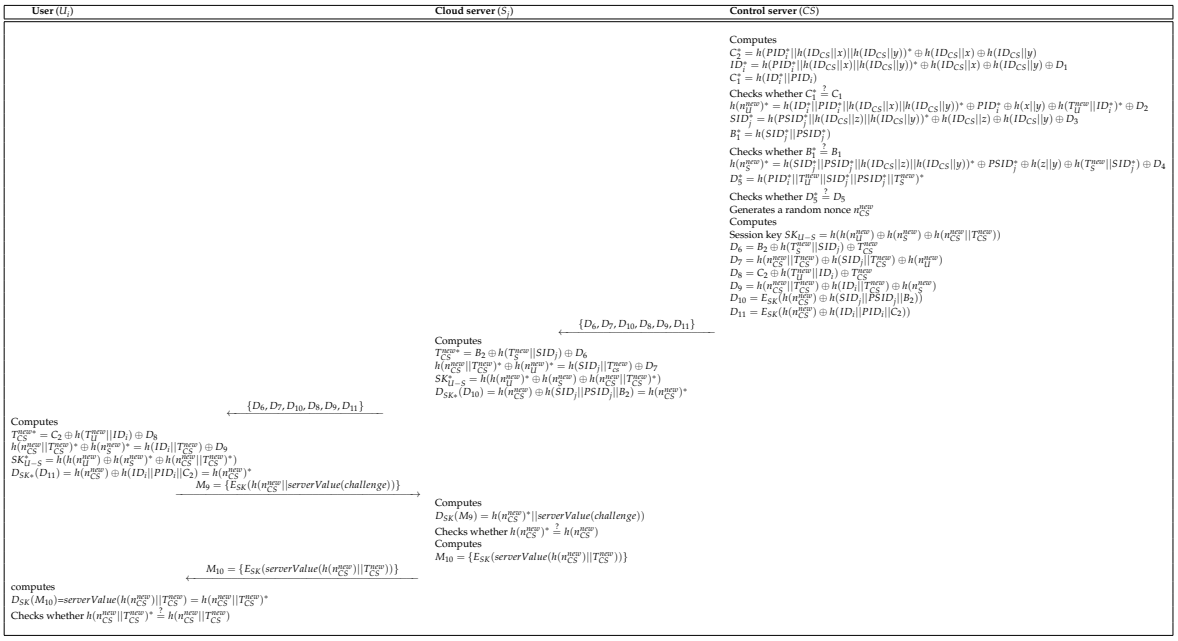

Figure 4. Authentication process of the Pelaez et al.'s scheme [8].

4. Cryptanalysis of Pelaez et al.'s Scheme

In this section, we demonstrate that Pelaez et al.'s scheme does not resist replay, session key disclosure and impersonation attacks and show that their scheme does not achieve secure mutual authentication and anonymity.

4.1. Impersonation Attack

The impersonation attack is that a malicious adversary try to impersonate as a legitimate user. When a malicious adversary U_{MA} may attempt to impersonate a legal user, U_{MA} can easily generate the login request message of U_i. According to Section 1.1, U_{MA} can obtain smart card of U_i and can extract the data $\{PID_i, C_2, C_3, C_4, h(n_U)\}$ stored in smart card. Furthermore, U_{MA} intercepts the message transmitted via an open channel. Finally, U_{MA} performs the impersonation attack as below:

Step 1: A malicious adversary U_{MA} can compute real identity $ID_i = C_2 \oplus D_1$ of legitimate user U_i and $h(n_U^{new}) = D_2 \oplus C_3 \oplus h(T_{MA}^{new}||ID_i)$. Then, U_{MA} generates timestamp T_{MA}^{new} and random nonce n_{MA}^{new}, computes $D_{2MA} = C_3 \oplus h(T_{MA}^{new}||ID_i) \oplus h(n_{MA}^{new})$, and sends $\{T_{MA}^{new}, D_1, PID_i, D_{2MA}\}$ to the S_j.

Step 2: Upon getting the message from U_{MA}, the S_j generates random nonces T_S^{new} and n_S^{new} and computes $D_3 = B_2 \oplus SID_j$, $D_4 = B_3 \oplus h(T_S^{new}||SID_j) \oplus h(n_S^{new})$ and $D_5 = h(PID_i||T_{MA}^{new}||SID_j||PSID_j||T_S^{new})$. Then, the S_j sends $\{T_{MA}^{new}, D_1, PID_i, D_{2MA}, T_S^{new}, D_3, PSID_j, D_4, D_5\}$ to the CS.

Step 3: Upon getting the message from S_j, the CS computes $C_2^* = h(PID_i^*||h(ID_{CS}||x)||h(ID_{CS}||y))^* \oplus h(ID_{CS}||x) \oplus h(ID_{CS}||y)$, $ID_i^* = h(PID_i^*||h(ID_{CS}||x)||h(ID_{CS}||y))^* \oplus h(ID_{CS}||x) \oplus h(ID_{CS}||y) \oplus D_1$ and $C_1^* = h(ID_i^*||PID_i)$. Then, the CS checks whether $C_1^* \stackrel{?}{=} C_1$. If it is valid, the CS authenticates U_{MA}. Then, the CS computes $h(n_{MA}^{new})^* = h(ID_i^*||PID_i^*||h(ID_{CS}||x)||h(ID_{CS}||y))^* \oplus$

$PID_i^* \oplus h(x||y) \oplus h(T_{MA}^{new}||ID_i^*)^* \oplus D_2$. After that, the CS computes $SID_j^* = h(PSID_j^*||h(ID_{CS}||z)||h(ID_{CS}||y))^* \oplus h(ID_{CS}||z) \oplus h(ID_{CS}||y) \oplus D_3$ and $B_1^* = h(SID_j^*||PSID_j^*)$. Then, the CS checks whether $B_1^* \stackrel{?}{=} B_1$. If it is valid, the CS authenticate S_j. After that, the CS recovers $h(n_S^{new})^* = h(SID_j^*||PSID_j^*||h(ID_{CS}||z)||h(ID_{CS}||y))^* \oplus PSID_j^* \oplus h(z||y) \oplus h(T_S^{new}||SID_j^*) \oplus D_4$. Then, the CS computes $D_5^* = h(PID_i^*||T_{MA}^{new}||SID_j^*||PSID_j^*||T_S^{new})^*$ and checks whether $D_5^* \stackrel{?}{=} D_5$. If it is valid, the CS have evidence of the connection attempt between U_{MA} and S_j. To key agreement and mutual authentication, the CS generates a random nonce n_{CS}^{new} and computes the session key $SK_{MA-S} = h(h(n_{MA}^{new}) \oplus h(n_S^{new}) \oplus h(n_{CS}^{new}||T_{CS}^{new}))$. Then, the CS computes $D_6 = B_2 \oplus h(T_S^{new}||SID_j) \oplus T_{CS}^{new}$, $D_{7MA} = h(n_{CS}^{new}||T_{CS}^{new}) \oplus h(SID_j||T_{CS}^{new}) \oplus h(n_{MA}^{new})$, $D_{8MA} = C_2 \oplus h(T_{MA}^{new}||ID_i) \oplus T_{CS}^{new}$, $D_9 = h(n_{CS}^{new}||T_{CS}^{new}) \oplus h(ID_i||T_{CS}^{new}) \oplus h(n_S^{new})$, $D_{10MA} = E_{SK}(h(n_{CS}^{new}) \oplus h(SID_j||PSID_j||B_2))$ and $D_{11MA} = E_{SK}(h(n_{CS}^{new}) \oplus h(ID_i||PID_i||C_2))$. Finally, the CS sends $\{D_6, D_{7MA}, D_{10MA}, D_{8MA}, D_9, D_{11MA}\}$ to the S_j.

Step 4: Upon getting the message from CS, the S_j computes $T_{CS}^{new*} = B_2 \oplus h(T_S^{new}||SID_j) \oplus D_6$, $h(n_{CS}^{new}||T_{CS}^{new})^* \oplus h(n_{MA}^{new})^* = h(SID_j||T_{cs}^{new}) \oplus D_{7MA}$, $SK_{U-S}^* = h(h(n_{MA}^{new})^* \oplus h(n_S^{new}) \oplus h(n_{CS}^{new}||T_{CS}^{new})^*)$ and decrypts $D_{SK*}(D_{10MA}) = h(n_{CS}^{new}) \oplus h(SID_j||PSID_j||B_2) = h(n_{CS}^{new})^*$. After that, the S_j sends $\{D_6, D_{7MA}, D_{10MA}, D_{8MA}, D_9, D_{11MA}\}$ to the U_{MA}.

Step 5: Upon getting the messages from S_j, the U_{MA} computes $T_{CS}^{new*} = C_2 \oplus h(T_{MA}^{new}||ID_i) \oplus D_{8MA}$, $h(n_{CS}^{new}||T_{CS}^{new})^* \oplus h(n_S^{new})^* = h(ID_i||T_{CS}^{new}) \oplus D_9$, $SK_{MA-S}^* = h(h(n_U^{new}) \oplus h(n_S^{new})^* \oplus h(n_{CS}^{new}||T_{CS}^{new})^*)$ and decrypts $D_{SK*}(D_{11MA}) = h(n_{CS}^{new}) \oplus h(ID_i||PID_i||C_2) = h(n_{CS}^{new})^*$. For mutual authentication with S_j, the U_{MA} computes $M_{9MA} = \{E_{SK}(h(n_{CS}^{new}||serverValue(challenge)))\}$ and sends M_{9MA} to the S_j.

Step 6: Upon getting the messages from U_{MA}, the S_j computes $D_{SK}(M_{9MA}) = h(n_{CS}^{new})^*||serverValue(challenge))$ and checks whether $h(n_{CS}^{new})^* \stackrel{?}{=} h(n_{CS}^{new})$. Finally, the S_j computes $M_{10MA} = \{E_{SK}(serverValue(h(n_{CS}^{new})||T_{CS}^{new}))\}$ and sends M_{10MA} to the U_{MA}.

Step 7: Upon getting the messages from S_j, the U_{MA} computes $D_{SK}(M_{10MA}) = serverValue(h(n_{CS}^{new})||T_{CS}^{new}) = h(n_{CS}^{new}||T_{CS}^{new})^*$ and checks whether $h(n_{CS}^{new}||T_{CS}^{new})^* \stackrel{?}{=} h(n_{CS}^{new}||T_{CS}^{new})$.

U_{MA} can successfully generates the login request message and session key between U_{MA} and S_j. As a result, we show that Pelaez et al.'s scheme cannot withstand impersonation attack.

4.2. Session Key Disclosure Attack

The session key disclosure attack is that a malicious adversary can obtain the session key between U_i and S_j. Pelaez et al. claimed that their scheme can ensure security of session key because a malicious adversary cannot obtain random nonce n_U^{new}, n_S^{new}, n_{CS}^{new} and current timestamp T_{CS}^{new}. However, according to Section 1.1, a malicious adversary U_{MA} can extract the data $\{PID_i, C_2, C_3, C_4, h(n_U)\}$ stored in the smart card and can obtain the transmitted messages $D_1, D_2, T_U^{new}, D_8, D_9$ via an open channel. Therefore, a malicious adversary U_{MA} can easily compute session key $SK_{U-S}^* = h(h(n_U^{new})^* \oplus h(n_S^{new}) \oplus h(n_{CS}^{new}||T_{CS}^{new})^*)$.

4.3. Replay Attack

Replay attack is that a malicious adversary try to obtain sensitive messages of user using the messages transmitted in previous and current session. Pelaez et al. claimed that their scheme can resist replay attack because a malicious adversary U_{MA} cannot obtain random nonce and timestamp. However, U_{MA} can calculate the random nonce and timestamp of legitimate user correctly. According to 4.1 , U_{MA} also impersonates a legitimate user U_i. Therefore, U_{MA} can obtain n_U^{new}, n_S^{new} and n_{CS}^{new} and timestamp T_U^{new}, T_S^{new} and T_{CS}^{new}. As a result, Pelaez et al.'s scheme does not withstand replay attack.

4.4. Mutual Authentication

Pelaez et al claimed that their protocol allows secure mutual authentication among the user U_i, the cloud server S_j, and the control server CS. However, according to Section 3.1, their protocol does not withstand to impersonation attack , as a malicious adversary U_{MA} can successfully generate authentication request message $D_2 = C_3 \oplus h(T_U^{new}||ID_i) \oplus h(n_U^{new})$. Therefore, Pelaez et al.'s scheme does not achieve secure mutual authentication.

4.5. Anonymity

Pelaez et al claimed that a malicious adversary U_{MA} cannot obtain the real identity ID_i of legitimate user. However, according to Section 1.1, a malicious adversary U_{MA} can extract the secret parameter C_2 stored in the smart card and can intercept the transmitted message D_1 via an open channel. U_{MA} can also compute $ID_i = C_2 \oplus D_1$ and easily obtain real identity of legitimate user U_i. Therefore, Pelaez et al.'s scheme does not guarantee anonymity.

5. Proposed Scheme

In this section, we propose a secure and lightweight three-factor authentication scheme for IoT in cloud computing environment to enhance security drawbacks of Pelaez et al.'s scheme. The proposed scheme consists of three processes: registration, login and authentication, and password change. The details of each process are presented below.

5.1. User Registration Process

A new user U_i who requests the use of the IoT services must register with control server CS. Figure 5 shows the user registration process of proposed scheme and the detailed processes are as below.

Step 1: The U_i selects ID_i and PW_i and imprints biometric BIO_i. After that, U_i computes $\langle R_i, P_i \rangle = Gen(BIO_i)$, $RPW_i = h(PW_i||R_i)$ and sends messages $\{ID_i, RPW_i\}$ to control server CS via a secure channel.

Step 2: After getting the messages from U_i, the CS generates a random nonce S_1 and computes $RID_i = h(ID_i||h(S_1||K_S))$, $X_i = h(RID_i||K_S||S_1)$, $A_i = X_i \oplus h(RID_i||RPW_i)$, and $B_i = h(X_i||RPW_i)$. Then, the CS stores $\{S_1\}$, $\{A_i, B_i\}$ in a database and smart card, respectively. The CS sends $\{RID_i\}$ and issues smart card to U_i via a secure channel.

Step 3: After getting the message and smart card from CS, the U_i computes $Q_i = h(ID_i||PW_i||R_i) \oplus RID_i$ and stores $\{Q_i\}$ in a smart card SC.

5.2. Cloud Server Registration Process

A cloud server S_j must register with the control server CS to provide IoT service to the users. Figure 6 shows the cloud server registration process of proposed scheme and the detailed processes are as below.

Step 1: The cloud server S_j selects SID_j and generates a random nonce r_j. After that, the S_j sends messages $\{SID_j, r_j\}$ to the CS via a secure channel.

Step 2: After getting the messages, the CS generates a random nonce S_2 and computes $RSID_j = h(SID_j||r_j||K_S)$ and $SI_j = h(RSID_j||h(S_2||K_S))$. Then, the CS stores $\{S_2\}$ in a database and sends messages $\{RSID_j, SI_j\}$ to the S_j via a secure channel.

Step 3: After getting the messages, the S_j stores $\{RSID_j, SI_j\}$ in a database.

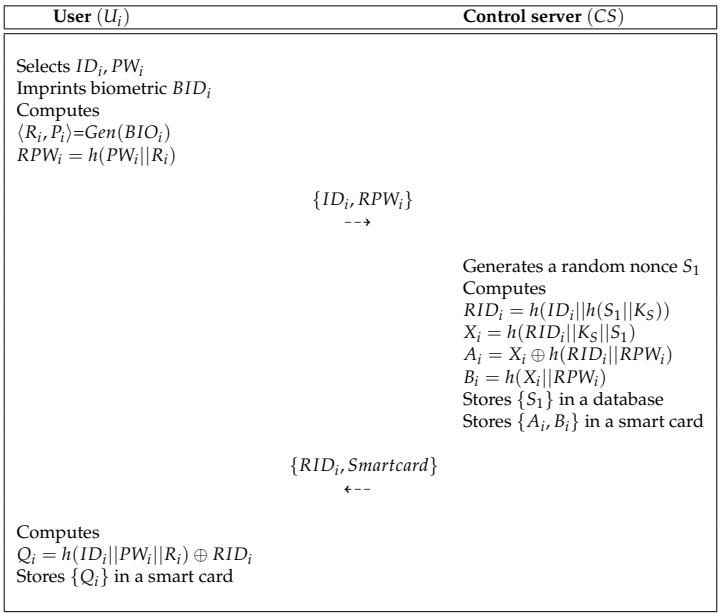

Figure 5. User registration process of the proposed scheme.

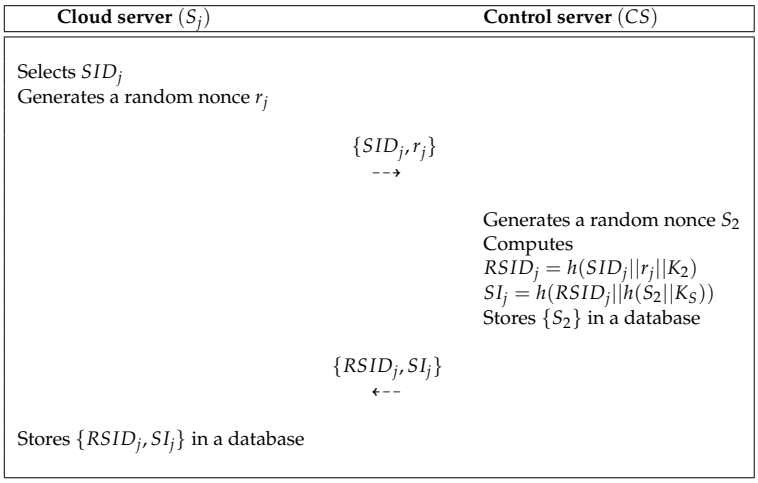

Figure 6. Cloud server registration process of the proposed scheme.

5.3. Login and Authentication Process

A user U_i who requests access to IoT service must send a login request message to the CS. Figure 7 shows the login and authentication process of the proposed scheme. The detailed process is as below.

Step 1: The U_i inputs ID_i, PW_i and imprints biometric BID_i. Then, the U_i calculates $R_i = Rep(BIO_i, P_i)$, $RID_i = h(ID_i||PW_i||R_i) \oplus Q_i$, $RPW_i = h(PW_i||R_i)$, $X_i = A_i \oplus h(RID_i||RPW_i)$ and $B_i^* = h(X_i||RPW_i)$. The U_i checks whether $B_i^* \stackrel{?}{=} B_i$. If it is correct, the U_i generates a random nonce RU_i. After that, the U_i computes $M_1 = RU_i \oplus X_i$, $CID_i = ID_i \oplus h(X_i||RU_i)$ and $M_2 = h(ID_i||X_i||RU_i)$ and sends login request messages $\{M_1, M_2, CID_i, RID_i\}$ to the S_j via an open channel.

Step 2: Upon getting the messages from the U_i, the S_j generates a random nonce RS_j and computes $D_1 = SI_j \oplus RS_j$, $CSID_j = SID_j \oplus h(SI_j||RS_j)$ and $D_2 = h(SID_j||SI_j||RS_j)$. Then, the S_j sends the messages $\{M_1, M_2, CID_i, RID_i, D_1, D_2, CSID_j, RSID_j\}$ to the CS via an open channel.

Step 3: Upon getting the messages from the S_j, the CS computes $X_i = h(RID_i||K_S||S_1)$, $RU_i = M_1 \oplus X_i$, $ID_i = CID_i \oplus h(X_i||RU_i)$, and $M_2^* = h(ID_i||X_i||RU_i)$ and checks whether

$M_2^* \stackrel{?}{=} M_2$. If it is correct, the CS computes $SI_j = h(RSID_j||h(S_2||K_S))$, $RS_j = h(D_1) \oplus SI_j$, $SID_j = CSID_j \oplus h(SI_j||RS_j)$, and $D_2^* = h(SID_j||SI_j||RS_j)$ and checks whether $D_2^* \stackrel{?}{=} D_2$. If it is valid, the CS computes $M_3 = RS_j \oplus h(ID_i||RU_i)$, $D_3 = RU_i \oplus h(SID_j||RS_j)$ and $Q_{CS} = h(RU_i||RS_j||SI_j)$. Then, the CS updates RID_i to RID_i^{new} and replaces $\{RID_i\}$ with $\{RID_i^{new}\}$. Finally, the CS sends messages $\{M_3, D_3, Q_{CS}\}$ to the S_j.

Step 4: Upon getting the messages from the CS, the S_j computes $RU_i = D_3 \oplus h(SID_j||RS_j)$ and $Q_{CS}^* = h(RU_i||RS_j||SI_j)$ and checks whether $Q_{CS}^* \stackrel{?}{=} Q_{CS}$. If it is valid, the S_j computes $SK_i = h(RU_i||RS_j)$ and $Q_{CU} = h(RU_i||RS_j||SK_i)$ and sends messages $\{M_3, Q_{CU}\}$ to the U_i.

Step 5: Upon getting the messages from the S_j, the U_i computes $RS_j = M_3 \oplus h(ID_i||RU_i)$, $SK_i = h(RU_i||RS_j)$ and $Q_{CU}^* = h(RU_i||RS_j||SK_i)$ and checks whether $Q_{CU}^* \stackrel{?}{=} Q_{CU}$. If it is correct, the U_i computes $RID_i^{new} = h(RID_i||h(RU_i||RS_j))$ and RID_i to RID_i^{new}. After that, the smart card updates $A_i^{new} = X_i \oplus h(RID_i^{new}||RPW_i)$ and $Q_i^{new} = h(ID_i||PW_i||R_i) \oplus RID_i^{new}$ and replaces $\{A_i, Q_i\}$ with $\{A_i^{new}, Q_i^{new}\}$. As a result, the U_i, S_j and CS achieve the mutual authentication successfully.

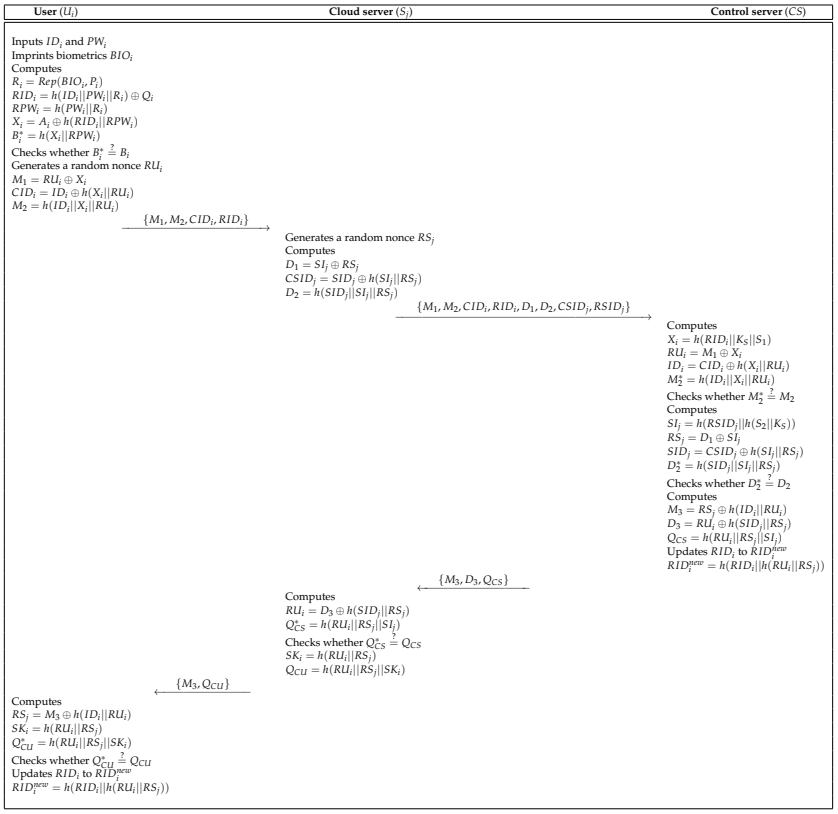

Figure 7. Login and authentication process of the proposed scheme.

5.4. Password Change Process

When U_i wants to update his/her password, the U_i can freely update their password in the proposed scheme. Figure 8 shows the password change process of the proposed scheme. The detailed process is as below.

Step 1: The U_i chooses ID_i^*, PW_i^* and imprints biometrics BIO_i^*. Then, the U_i calculates $\langle R_i, P_i \rangle = Gen(BIO_i^*)$, $RPW_i^* = h(PW_{MU}||R_i)$ and sends $\{ID_{MU}^*, RPW_i^*\}$ to the smart card SC.

Step 2: After getting the message from U_i, the SC computes $X_i^* = A_i^* \oplus h(ID_i^*||RPW_i^*)$ and $B_i^* = h(X_i^*||RPW_i^*)$ and checks whether $B_i^* \stackrel{?}{=} B_i$. If it is equal, the SC sends the authentication message to the U_i.

Step 3: Upon getting the message from the SC, the U_i inputs a new password PW_i^{new} and imprints a new biometrics BIO_i^{new}. U_i computes $\langle R_i^{new}, P_i^{new}\rangle$=$Gen(BIO_i^{new})$, $RPW_i^{new} = h(PW_i^{new}||R_i^{new})$ and sends $\{RPW_i^{new}\}$ to the SC.

Step 4: Upon getting the message from the U_i, the SC computes $A_i^{new} = X_i^* \oplus h(ID_i^*||RPW_i^{new})$, $B_i^{new} = h(X_i^*||RPW_i^{new})$ and replaces $\{A_i, B_i\}$ with $\{A_i^{new}, B_i^{new}\}$.

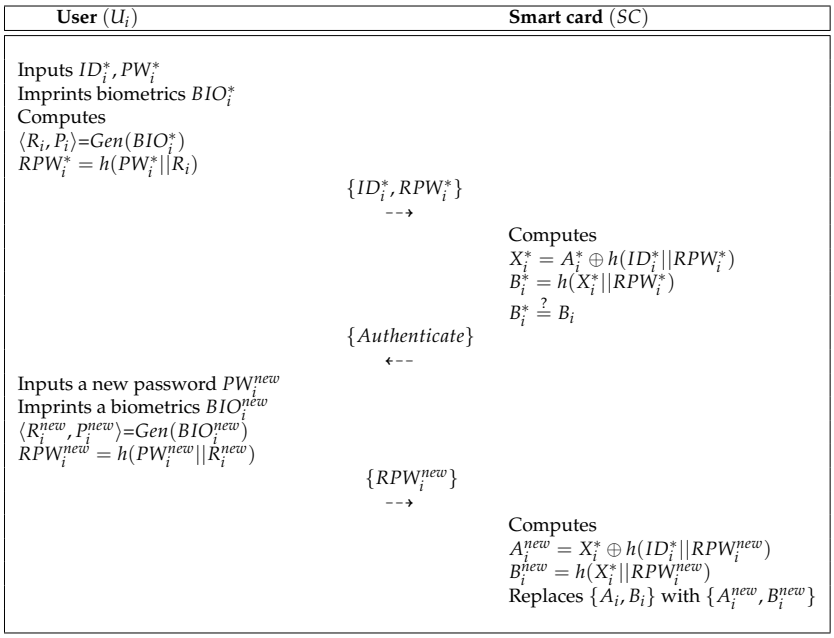

Figure 8. Password change process of the proposed scheme.

6. Security Analysis

To assess secure mutual authentication of the proposed scheme, we utilize the BAN logic, which is widely accepted formal security model. Furthermore, we perform an informal security analysis to assess the safety of proposed scheme against various types of attacks.

6.1. Informal Security Analysis

The security of the proposed scheme is accessed utilizing an informal security analysis. Our scheme can withstand against various types of attacks, including impersonation, replay, session key disclosure attacks, and allows secure mutual authentication and anonymity.

6.1.1. Impersonation Attack

When a malicious adversary U_{MA} may attempt to impersonate a legitimate user, U_{MA} must generate a login request message $M_2 = h(ID_i||X_i||RU_i)$ correctly. However, U_{MA} cannot compute it because U_{MA} cannot obtain U_i's random nonce RU_i, real identity ID_i, and secret parameter X_i. Therefore, our scheme is secure against the impersonation attack because U_{MA} cannot calculate a login request message successfully.

6.1.2. Replay Attack

If a malicious adversary U_{MA} may attempt to impersonate legal user by resending messages transmitted in a previous session, U_{MA} cannot utilize the previous messages because the CS checks

whether $M_2^* \stackrel{?}{=} M_2$ and $D_2^* \stackrel{?}{=} D_2$, respectively. Furthermore, our scheme can withstand replay attack by using dynamic random nonce RU_i and RS_j that are changed every session. Therefore, our scheme protects against replay attack.

6.1.3. Session Key Disclosure Attack

In our scheme, a malicious adversary U_{MA} cannot compute session key SK_i because U_{MA} cannot obtain random nonce RU_i and RS_j. In addition, U_{MA} cannot obtain random nonce RU_i and RS_j without secret parameter X_i and SI_j. Consequently, our scheme withstands the session key disclosure attack.

6.1.4. Smart card Stolen Attack

According to Section 1.1, we suppose that a U_{MA} can obtain a smart card and extract the data $\{A_i, B_i, Q_i\}$ stored in the smart card. However, the U_{MA} cannot obtain sensitive information ID_i and PW_i of legitimate user because the data stored in the smart card are protected $A_i = X_i \oplus h(RID_i||RPW_i)$, $B_i = h(X_i||RPW_i)$ and $Q_i = h(ID_i||PW_i||R_i) \oplus RID_i$ by using a hash function and XOR operation.

6.1.5. Mutual Authentication

In our scheme, after getting the request message $\{M_1, M_2, CID_i, RID_i\}$ from the U_i, the control server CS checks whether $M_2^* \stackrel{?}{=} M_2$. If it is correct, CS authenticates U_i. After getting the messages $\{D_1, D_2, CSID_j, RSID_j\}$ from cloud server S_j, the CS checks whether $D_2^* \stackrel{?}{=} D_2$. If it is equal, CS authenticates S_j. After getting the messages $\{M_3, D_3, Q_{CS}\}$ from the CS, the S_j checks whether $Q_{CS}^* \stackrel{?}{=} Q_{CS}$. If it is correct, S_j authenticates CS. After getting the messages $\{Q_{CU}\}$ from the S_j, the U_i checks whether $Q_{CU}^* \stackrel{?}{=} Q_{CU}$. Finally, the U_i authenticates S_j. As a result, our scheme achieve secure mutual authentication among U_i, S_j, and CS because a malicious adversary U_{MA} does not know secret parameters X_i and SI_j.

6.1.6. Anonymity

A malicious adversary U_{MA} cannot obtain the real identity ID_i of legitimate user because it is masked by using hash function and XOR operation such as $CID_i = ID_i \oplus h(X_i||RU_i)$. In addition, the U_{MA} cannot obtain secret parameter X_i and random nonce RU_i. Consequently, our scheme provides anonymity.

6.2. Security Features

We shows the better security levels achieved by the proposed scheme compared with some existing schemes [8,23–25]. The existing schemes are insecure against various attacks, including impersonation, session key disclosure smart card stolen, and replay attacks and cannot provide mutual authentication and anonymity. Table 1 shows the analysis results of the security features.

Table 1. Security features comparison.

Security Features	Xue et al. [24]	Amin et al. [25]	Zhou et al. [23]	Pelaez et al. [8]	Ours
Impersonation attack	×	×	×	×	○
Smart card stolen attack	×	×	○	×	○
Session key disclosure attack	×	○	×	×	○
Replay attack	○	○	×	×	○
Anonymity	×	○	○	×	○
Mutual authentication	×	○	×	×	○

○, preserves the security features; ×, does not preserve the security features;

6.3. BAN Logic Based Authentication Proof

We performed security analysis utilizing the BAN logic to demonstrate the secure mutual authentication of the proposed scheme. We present the BAN logic notations in Table 2. Furthermore, we define the rules, the goals, the idealized form, and the assumptions for BAN logic analysis. We prove that the proposed scheme provides secure mutual authentication among U_i, S_j and CS.

Table 2. Notations for BAN logic.

Notation	Description
$A\mid\equiv X$	A **believes** statement X
#X	Statement X is **fresh**
$A \lhd X$	A **sees** statement X
$A\mid\sim X$	A once **said** X
$A \Rightarrow X$	A has got **jurisdiction** of X
$< X >_Y$	X is **combined** with Y
$\{X\}_K$	X is **encrypted** under key K
$A \overset{K}{\leftrightarrow} B$	A and B may use **shared key** K to communicate
SK	Session key used in the current session

6.3.1. BAN Logic Rules

The rules of BAN logic are as below.

1. Message meaning rule :
$$\frac{A \mid\equiv A \overset{K}{\leftrightarrow} B, \quad A \lhd \{X\}_K}{A \mid\equiv B \mid\sim X}$$
2. Nonce verification rule :
$$\frac{A \mid\equiv \#(X), \quad A \mid\equiv B \mid\sim X}{A \mid\equiv B \mid\equiv X}$$
3. Jurisdiction rule :
$$\frac{A \mid\equiv B \mid\Longrightarrow X, \quad A \mid\equiv B \mid\equiv X}{A \mid\equiv X}$$
4. Freshness rule :
$$\frac{A \mid\equiv \#(X)}{A \mid\equiv \#(X, Y)}$$
5. Belief rule :
$$\frac{A \mid\equiv (X, Y)}{A \mid\equiv X.}$$

6.3.2. Goals

To assess the BAN logic proof, we present the goals of the proposed scheme as below.

Goal 1: $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$

Goal 2: $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$

Goal 3: $U_i \mid\equiv S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$

Goal 4: $S_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$

6.3.3. Idealized Forms

To assess the BAN logic proof, we define the assumptions of the proposed scheme as below.

Msg_1: $U_i \rightarrow S_j$: $(RID_i, ID_i, RU_i)_{X_i}$

Msg_2: $S_j \rightarrow CS$: $(RID_i, ID_i, RU_i, RSID_j, SID_j, RS_j)_{SI_j}$

Msg_3: $CS \rightarrow S_j$: $(ID_i, SID_j, RU_i, RS_j)_{SI_j}$

Msg_4: $S_j \rightarrow U_i$: $(ID_i, RU_i, RS_j, (U_i \overset{SK}{\longleftrightarrow} S_j))_{X_i}$

6.3.4. Assumptions

We present the initial assumptions to assess the BAN logic proof.

A_1: $S_j \mid\equiv (U_i \overset{X_i}{\longleftrightarrow} S_j)$

A_2: $S_j \mid\equiv \#(RU_i)$

A_3: $CS \mid\equiv (CS \overset{SI_j}{\longleftrightarrow} S_j)$

A_4: $CS \mid\equiv \#(RS_j)$

A_5: $S_j \mid\equiv (CS \overset{SI_j}{\longleftrightarrow} S_j)$

A_6: $FA \mid\equiv \#(RS_j)$

A_7: $U_i \mid\equiv (U_i \overset{X_i}{\longleftrightarrow} S_j)$

A_8: $U_i \mid\equiv \#(RS_j)$

A_9: $U_i \mid\equiv S_j \Rightarrow (U_i \overset{SK}{\longleftrightarrow} S_j)$

A_{10}: $S_j \mid\equiv U_i \Rightarrow (U_i \overset{SK}{\longleftrightarrow} S_j)$

6.3.5. Proof Using BAN Logic

The proof then proceeds as below.

Step 1: According to Msg_1, we could get

$$(S_1) : S_j \lhd (RID_i, ID_i, RU_i)_{X_i}$$

Step 2: Using the message meaning rule with S_1 and A_1, we get

$$(S_2) : S_j \mid\equiv U_i \mid\sim (RID_i, ID_i, RU_i)_{X_i}$$

Step 3: From the freshness rule with S_2 and A_2, we obtain

$$(S_3) : S_j \mid\equiv \#(RID_i, ID_i, RU_i)_{X_i}$$

Step 4: Using the nonce verification with S_2 and S_3, we get

$$(S_4) : S_j \mid\equiv U_i \mid\equiv (RID_i, ID_i, RU_i)_{X_i}$$

Step 5: From the belief rule with S_4, we obtain

$$(S_5) : S_j \mid\equiv U_i \mid\equiv (RU_i)_{X_i}$$

Step 6: According to Msg_2, we could get

$$(S_6) : CS \lhd (RID_i, ID_i, RU_i, RSID_j, SID_j, RS_j)_{SI_j}$$

Step 7: Using the message meaning rule with S_6 and A_3, we get

$$(S_7) : CS \mid\equiv S_j \mid\sim (RID_i, ID_i, RU_i, RSID_j, SID_j, RS_j)_{SI_j}$$

Step 8: From the freshness rule with S_7 and A_4, we obtain

$$(S_8) : CS \mid\equiv \#(RID_i, ID_i, RU_i, RSID_j, SID_j, RS_j)_{SI_j}$$

Step 9: Using the nonce verification rule with S_7 and S_8, we get

$$(S_9) : CS \mid\equiv S_j \mid\equiv (RID_i, ID_i, RU_i, RSID_j, SID_j, RS_j)_{SI_j}$$

Step 10: According to Msg_3, we could get

$$(S_{10}) : S_j \lhd (ID_i, SID_j, RU_i, RS_j)_{SI_j}$$

Step 11: Using the message meaning rule with S_{10} and A_5, we get

$$(S_{11}) : S_j \mid\equiv CS \mid\sim (ID_i, SID_j, RU_i, RS_j)_{SI_j}$$

Step 12: From the freshness rule with S_{11} and A_6, we obtain

$$(S_{12}) : S_j \mid\equiv \#(ID_i, SID_j, RU_i, RS_j)_{SI_j}$$

Step 13: Using the nonce verification rule with S_{11} and S_{12}, we get

$$(S_{13}) : S_j \mid\equiv CS \mid\equiv (ID_i, SID_j, RU_i, RS_j)_{SI_j}$$

Step 14: According to Msg_4, we could get

$$(S_{14}) : U_i \lhd (ID_i, RU_i, RS_j, (U_i \overset{SK}{\longleftrightarrow} S_j))_{X_i}$$

Step 15: Using the message meaning rule with S_{14} and A_7, we get

$$(S_{15}) : U_i \mid\equiv S_j \mid\sim (ID_i, RU_i, RS_j, (U_i \overset{SK}{\longleftrightarrow} S_j))_{X_i}$$

Step 16: From the freshness rule with S_{15} and A_8, we obtain

$$(S_{16}) : U_i \mid\equiv \#(ID_i, RU_i, RS_j, (U_i \overset{SK}{\longleftrightarrow} S_j))_{X_i}$$

Step 17: Using the nonce verification with S_{15} and S_{16}, we get

$$(S_{17}) : U_i \mid\equiv S_j \mid\equiv (ID_i, RU_i, RS_j, (U_i \overset{SK}{\longleftrightarrow} S_j))_{X_i}$$

Step 18: From the belief rule with S_{17}, we obtain

$$(S_{18}) : U_i \mid\equiv S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j) \qquad \textbf{(Goal 3)}$$

Step 19: Using the jurisdiction rule with S_{18} and A_9, we get

$$(S_{19}) : U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j) \qquad \textbf{(Goal 1)}$$

Step 20: Because of $SK = h(RU_i||RS_j)$, from the S_5, S_9, S_{13} and S_{17} we could get

$$(S_{20}) : S_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j) \qquad \textbf{(Goal 4)}$$

Step 21: Using the jurisdiction rule with S_{19} and A_{10}, we obtain

$$(S_{21}) : S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j) \qquad \textbf{(Goal 2)}$$

Referring to Goals 1–4, we show that proposed scheme achieves secure mutual authentication among U_i, S_j and CS.

7. Simulation for Security Verification with the AVISPA tool

We performed a formal security verification of the proposed scheme utilizing AVISPA simulation tool [26,27] to evaluate the safety of the authentication protocol against MITM and replay attacks, which is widely accepted for formal security analysis [28–31]. To perform AVISPA simulation tool, the environment and the session of security protocol must be implemented using the High Level Protocols Specification Language (HLPSL).

7.1. HLPSL Specifications

We considered three basic roles: user U_i, cloud server S_j, and control server CS. Then, we present *session* and *environment* utilizing HLPSL in Figure 9, which contains the security goals. The role specifications of U_i, S_j, and CS are as shown in Figures 10–12.

```
role environment()
def=
const ua, s, cs : agent,
skuacs, skscs: symmetric_key,
h: hash_func,
sidj,idi: text,
ua_cs_rui, s_cs_rsj, cs_s_qcs, s_ua_qcu: protocol_id,
sp1, sp2, sp3, sp4, sp5: protocol_id

intruder_knowledge = {ua,s,cs,sidj,idi,h}
composition
session(ua,s,cs, skuacs, skscs,h)/
\session(i,s,cs, skuacs,skscs, h)
/\session(ua,i,cs, skuacs,skscs,h)
/\session(ua,s,i, skuacs,skscs,h)

end role
goal
secrecy_of sp1, sp2, sp3, sp4, sp5
authentication_on ua_cs_rui
authentication_on s_cs_rsj
authentication_on cs_s_qcs
authentication_on s_ua_qcu
end goal

environment()
```

```
Role for the session

role session(UA, S, CS : agent, SKuacs, SKscs : symmetric_key,
H: hash_func)

def=
local SN1, SN2, SN3, RV1, RV2, RV3: channel(dy)
composition
user(UA, S, CS, SKuacs, H, SN1, RV1)
/\ server(UA, S, CS, SKscs, H, SN2, RV2)
/\ controlserver(UA, S, CS,SKuacs, SKscs, H, SN3, RV3)
end role
```

Figure 9. Role for environment and session in HLPSL.

```
%%%%%%%%%%%% User

role user(UA, S, CS : agent, SKuacs : symmetric_key, H: hash_func, SND, RCV :
channel(dy))

played_by UA
def=
local State: nat,
   IDi,PWi,Ri,Pi,RPWi,RIDi,Xi,Ai,Bi,Qi,SIDj,Rj,RSIDj,SIj,S1,S2,Ks: text,
   RUi,M1,CIDi,M2,RSj,D1,CSIDj,D2,M3,D3,Qcs,SKi,Qcu : text

const sp1, sp2, sp3, sp4, sp5, ua_cs_rui, s_cs_rsj, cs_s_qcs, s_ua_qcu: protocol_id
init State := 0
transition

%%%%%%%%%%%%Registration phase
1. State = 0 /\ RCV(start) =|>
State' := 1 /\ Ri':=new()
       /\ RPWi' := H(PWi.Ri')
       /\ SND({IDi.RPWi'}_SKuacs)
          /\ secret({IDi, PWi, Ri'}, sp1, {UA})

%%%%%%%%%%%%Recieve smartcard
2. State = 1 /\ RCV
({H(IDi'.H(S1'.Ks)).xor(H(H(IDi.H(S1'.Ks)).Ks.S1'),H(IDi.H(PWi.Ri'))).H(H(H(IDi.
H(S1'.Ks)).Ks.S1').H(PWi.Ri'))}_SKuacs)=|>
 State' := 2  /\ Qi' := xor(H(IDi'.PWi.Ri'),H(IDi.H(S1'.Ks)))
%%%%%%%%%%%%Login & Authentication phase
       /\ RUi' := new()
       /\ M1' := xor(RUi',H(H(IDi.H(S1'.Ks)).Ks.S1'))
       /\ CIDi' := xor(IDi',H(H(H(IDi.H(S1'.Ks)).Ks.S1').RUi'))
       /\ M2' := H(IDi.H(H(IDi.H(S1'.Ks)).Ks.S1').RUi')
       /\ SND(M1'.M2'.CIDi'.H(IDi.H(S1'.Ks)))
       /\ witness(UA,CS,ua_cs_rui,RUi')
3. State = 2 /\ RCV(xor(RSj',H(IDi.RUi')).H(RUi'.RSj'.H(RUi'.RSj'))) =|>
State' := 3 /\ SKi' := H(RUi'.RSj')
       /\ request(UA,S,s_ua_qcu,SKi')
end role
```

Figure 10. Role specification for user U_i.

```
%%%%%%%%%%%% Cloud Server

role server(UA, S, CS : agent, SKscs : symmetric_key, H: hash_func, SND, RCV :
channel(dy))

played_by S
def=
local State: nat,
   IDi,PWi,Ri,Pi,RPWi,RIDi,Xi,Ai,Bi,Qi,SIDj,Rj,RSIDj,SIj,S1,S2,Ks: text,
   RUi,M1,CIDi,M2,RSj,D1,CSIDj,D2,M3,D3,Qcs,SKi,Qcu : text
const sp1, sp2, sp3, sp4, sp5, ua_cs_rui, s_cs_rsj, cs_s_qcs, s_ua_qcu: protocol_id
init State := 0
transition

1. State = 0 /\ RCV(start) =|>
  State' := 1 /\ Rj' := new() /\S2':=new()
         /\ SND({SIDj.Rj'}_SKscs)
         /\ RCV({H(SIDj.Rj'.Ks).H(H(SIDj.Rj'.Ks).H(S2'.Ks))}_SKscs)

2. State = 2
/\ RCV(xor(RUi',H(H(IDi.H(S1'.Ks)).Ks.S1')).H(IDi.H(H(IDi.H(S1'.Ks)).Ks.S1').RUi
').xor(IDi',H(H(H(IDi.H(S1'.Ks)).Ks.S1').RUi')).H(IDi.H(S1'.Ks))) =|>
State' := 3 /\ RSj':= new() /\S2':=new() /\ Rj' := new()
       /\ D1' := xor(H(H(SIDj.Rj'.Ks).H(S2'.Ks)),RSj')
       /\ CSIDj' := xor(SIDj,H(H(H(SIDj.Rj'.Ks).H(S2'.Ks)).RSj'))
       /\ D2' := H(SIDj.H(H(SIDj.Rj'.Ks).H(S2'.Ks)).RSj')
       /\ SND(xor(RUi',H(H(IDi.H(S1'.Ks)).Ks.S1')).H(IDi.H(H(IDi.H(S1'.Ks)).Ks.S1'
).RUi').xor(IDi',H(H(H(IDi.H(S1'.Ks)).Ks.S1').RUi')).H(IDi.H(S1'.Ks)).D1'.D2'.CSID
j'.H(SIDj.Rj'.Ks))
       /\ witness(S,CS,s_cs_rsj,RSj')
3. State = 3
/\ RCV(xor(RSj',H(IDi.RUi')).xor(RUi',H(SIDj.RSj')).H(RUi'.RSj'.H(H(SIDj.Rj'.Ks).
H(S2'.Ks)))) =|>
State' := 4 /\ SKi' :=  H(RUi'.RSj')
       /\ Qcu' := H(RUi'.RSj'.SKi')
       /\ witness(S,UA,s_ua_qcu,SKi')
       /\ SND(xor(RSj',H(IDi.RUi')).Qcu')
       /\ request(CS,S,cs_s_qcs,RUi')
end role
```

Figure 11. Role specification for cloud server S_j.

The U_i receives the initial message and updates the updates the state value from 0 to 1. The U_i then sends the registration request messages $\{ID_i, RPW_i\}$ to the CS via a secure channel and receives $\{RID_i, Smartcard\}$ from the CS. The U_i updates the state value from 1 to 2. In the login and authentication phase, the U_i declares $witness(UA, CS, ua\_sn\_rui, RU_i')$ from the S_j, and then updates the state value from 2 to 3. Finally, the U_i receives the authentication messages $\{M_3, Q_{CU}\}$ from the S_j. The U_i checks whether $Q_{CU}^* \stackrel{?}{=} Q_{CU}$. If it is valid, the U_i authenticates the S_j successfully. The role specification for S_j is similarly defined.

```
%%%%%%%%%%%% Control Server

role controlserver(UA, S, CS : agent, SKuacs, SKscs : symmetric_key, H: hash_func,
SND, RCV : channel(dy))

played_by CS
def=
local State: nat,
   IDi,PWi,Ri,Pi,RPWi,RIDi,Xi,Ai,Bi,Qi,SIDj,Rj,RSIDj,SIj,S1,S2,Ks: text,
   RUi,M1,CIDi,M2,RSj,D1,CSIDj,D2,M3,D3,Qcs,SKi,Qcu : text
const sp1, sp2, sp3, sp4, sp5, ua_cs_rui, s_cs_rsj, cs_s_qcs, s_ua_qcu : protocol_id
init State := 0
transition

1. State = 0 /\ RCV({SIDj.Rj'}_SKscs) =|>
State' := 1 /\ S2' := new() /\ RSIDj' := H(SIDj.Rj'.Ks)
      /\ SIj' := H(RSIDj'.H(S2'.Ks))
      /\ SND({RSIDj'.SIj'}_SKscs)
      /\ secret({S2'},sp3,{CS})
      /\ secret({RSIDj',SIj'},sp4,{S,CS})
2. State = 1 /\ RCV({IDi.H(PWi.Ri')}_SKuacs) =|>
State' := 2 /\ S1':= new() /\ RIDi' := H(IDi.H(S1'.Ks))
      /\ Xi' := H(RIDi'.Ks.S1')
      /\ Ai' := xor(Xi',H(IDi.H(PWi.Ri')))
      /\ Bi' := H(Xi'.H(PWi.Ri'))
      /\ SND({RIDi'.Ai'.Bi'}_SKuacs)
      /\ secret({S1'},sp5,{CS})
3. State = 2
/\ RCV(xor(RUi',H(H(IDi.H(S1'.Ks)).Ks.S1')).H(IDi.H(H(IDi.H(S1'.Ks)).Ks.S1').RUi
').xor(IDi',H(H(H(IDi.H(S1'.Ks)).Ks.S1').RUi')).H(IDi.H(S1'.Ks)).xor(H(H(SIDj.Rj'.
Ks).H(S2'.Ks)),RSj').H(SIDj.H(H(SIDj.Rj'.Ks).H(S2'.Ks)).RSj').xor(SIDj,H(H(H(SID
j.Rj'.Ks).H(S2'.Ks)).RSj')).H(SIDj.Rj'.Ks)) =|>
State' := 3 /\ M3' := xor(RSj',H(IDi.RUi'))
       /\ D3' := xor(RUi',H(SIDj.RSj'))
       /\ Qcs' := H(RUi'.RSj'.H(H(SIDj.Rj'.Ks).H(S2'.Ks)))
      /\ witness(CS,S,cs_s_qcs,RUi')
      /\ SND(M3'.D3'.Qcs')
      /\ request(UA,CS, ua_cs_rui,RUi')
      /\ request(S,CS, s_cs_rsj,RSj')
end role
```

Figure 12. Role specification for control server CS.

7.2. AVISPA Simulation Result

We show the AVISPA results to verify the safety of the proposed scheme using OFMC and CL-AtSe. The OFMC checks whether the proposed scheme is safe from MITM attack. In addition, the CL-AtSe demonstrates the safety of the protocol against replay attack. Consequently, Figure 13 shows that the proposed scheme is secure against MITM and replay attacks though AVISPA simulation.

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/sj.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 7.92s
  visitedNodes: 1040 nodes
  depth: 9 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/sj.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed  : 0 states
  Reachable  : 0 states
  Translation: 0.08 seconds
  Computation: 0.00 seconds
```

Figure 13. Analysis of AVISPA simulation using OFMC and CL-AtSe.

8. Performance Analysis

We compared the computation cost, communication cost and security features of the proposed scheme with some existing schemes [8,23–25]. We show that the proposed scheme provides better efficiency and security features.

8.1. Computation Cost

We compared the computation overheads of the proposed scheme with some existing schemes [8,23–25]. To analyze of computation cost, we estimated using the following parameters. Table 3 shows the analysis results of computation cost and the detailed total cost are as below.

Table 3. A comparative summary: computation costs.

Schemes	User	Cloud Server	Control Server	Total	Total Cost (Case 1)	Total Cost (Case 2)
Xue et al. [24]	$12T_h$	$6T_h$	$18T_h$	$36T_h$	0.18612 ms	0.0011808 ms
Amin et al. [25]	$12T_h$	$4T_h$	$14T_h$	$30T_h$	0.1551 ms	0.000984 ms
Zhou et al. [23]	$13T_h$	$7T_h$	$23T_h$	$43T_h$	0.22231 ms	0.0014104 ms
Pelaez et al. [8]	$9T_h + 3T_s$	$6T_h + 3T_s$	$33T_h + 2T_s$	$48T_h + 8T_s$	0.42 ms	0.1730824 ms
Ours	$12T_h$	$6T_h$	$16T_h$	$34T_h$	0.17578 ms	0.0011152 ms

T_h, hash function; T_s, symmetric key cryptography operation using AES algorithm

The total computation cost for the proposed scheme and Pelaez et al.'s scheme are $34T_h$ and $48T_h + 8T_s$, respectively. We provide better efficiency than some existing schemes because the proposed scheme uses only hash and XOR operations. Therefore, our scheme is secure and efficient for practical IoT-based cloud computing environment.

- T_h denotes the time for the hash function (Case 1 $\approx$ 0.00517 ms [23] and Case 2 $\approx$ 0.0000328 ms [32]).
- T_s denotes the time for the symmetric key cryptography operation using AES algorithm (case 1 $\approx$ 0.02148 ms [23] and Case 2 $\approx$ 0.0214385 ms [32]).
- The XOR operation was not included because it is negligible compared to the other operations.

8.2. Communication Cost

We compared the communication overhead of the proposed scheme with some existing schemes [8,23–25]. In authentication phase of the proposed scheme, the transmitted messages $\{M_1, M_2, CID_i, RID_i\}$, $\{M_1, M_2, CID_i, RID_i, D_1, D_2, CSID_j, RSID_j\}$, $\{M_3, D_3, Q_{CS}\}$ and $\{M_3, Q_{CU}\}$ require (128 + 128 + 128 + 128 = 512 bits), (128 + 128 + 128 + 128 + 128 + 128 + 128 + 128 = 1024 bits), (128 + 128 + 128 = 384 bits), and (128 + 128 = 256 bits), respectively. Table 4 shows the analysis results of communication cost. Consequently, the proposed scheme is thus more efficient than other related schemes [8,23–25] because the total communications cost are 2176 bits (Case 1) and 4352 bits (Case 2).

- Case 1 defines that the pseudo-identity, random nonce, timestamp, identity, password, and hash function are 128 bits, respectively.
- Case 2 defines that the pseudo-identity, random nonce, timestamp, identity, password, and hash function are 256 bits, respectively.
- The block length for symmetric encryption is 128 bits.

Table 4. A comparative summary: communication costs.

Schemes	Message Length	Total Cost (Case 1)	Total Cost (Case 2)
Xue et al. [24]	30	3840 bits	7680 bits
Amin et al. [25]	27	3456 bits	6912 bits
Zhou et al. [23]	34	4352 bits	8704 bits
Pelaez et al. [8]	34	4352 bits	8704 bits
Ours	25	2176 bits	4352 bits

9. Conclusions

This paper shows that Pelaez et al.'s scheme does not defend various attacks such as impersonation, session key disclosure and replay attacks. Furthermore, we show that Pelaez et al.'s scheme cannot allow mutual authentication and anonymity. We propose a secure and lightweight three-factor authentication scheme for IoT in cloud computing environment to enhance the security drawbacks of Pelaez et al.'s scheme. Our scheme can withstand various types of attacks, including impersonation, session key disclosure and replay attacks, and can provide mutual authentication and anonymity. Then, we demonstrate that our scheme allows secure mutual authentication among U_i, S_j, and CS utilizing BAN logic analysis. We also performed a formal security verification analysis of the proposed scheme utilizing the AVISPA simulation tool. In addition, we compared the security features and performance of the proposed scheme with some existing schemes. We show that our scheme provides better safety and efficiency than related schemes. Therefore, our scheme can be suitable for practical IoT-based cloud computing environment because it is more secure and lightweight than the previous schemes.

Author Contributions: Conceptualization, S.Y.; software, S.Y. and K.P.; validation, K.P.; formal analysis, K.P.; writing—original draft preparation, S.Y.; writing—review and editing, K.P. and Y.P.; supervision, Y.P.

Funding: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning under Grant 2017R1A2B1002147 and in part by the BK21 Plus project funded by the Ministry of Education, Korea under Grant 21A20131600011.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Effectively and Securely Using the Cloud Computing Paradigm (v0.25). Available online: http://csrc.nist.gov/groups/SNS/cloud-computing (accessed on 5 August 2019)
2. Grobauer, B.; Walloscheck, T.; Stocker, E. Understanding cloud computing vulnerabilities. *IEEE Secur. Priv.* **2011**, *9*, 50–57. [CrossRef]
3. Lamport, L. Password authentication with insecure communication. *Commun. ACM* **1981**, *24*, 770–772. [CrossRef]
4. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In *Advances in Cryptology*; Springer: Berlin, Germany, 1999; pp. 388–397.
5. Amin, R.; Islam, S.K.; Biswas, G.P.; Khan, M.K.; Leng, L.; Kumar, N. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. *Comput. Netw.* **2016**, *101*, 42–62. [CrossRef]
6. Jiang, Q.; Zeadally, S.; Ma, J.; He, D. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access* **2017**, *5*, 3376–3392. [CrossRef]
7. Li, X.; Niu, J.; Kumari, S.; Wu, F.; Choo, K.K.R. A robust biometrics based three-factor authentication scheme for global mobility networks in smart city. *Future Gener. Comput. Syst.* **2018**, *83*, 607–618. [CrossRef]
8. Pelaez, R.M.; Cruz, H.T.; Michel, J.R.; Garcia, V.; Mena, L.J.; Felix, V.G.; Brust, A.O. An enhanced lightweight IoT-based authentication scheme in cloud computing circumstances. *Sensors* **2019**, *19*, 2098. [CrossRef] [PubMed]

9. Dolev, D.; Yao, A.C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
10. Park, Y.; Park, K.; Park, Y. Secure user authentication scheme with novel server mutual verification for multiserver environments. *J. Commun. Syst.* **2019**, *32*, 1–17. [CrossRef]
11. Park, K.; Park, Y.; Das, A.K.; Yu, S.; Lee, J.; Park, Y.H. A dynamic privacy-preserving key management protocol for V2G in social internet of things. *IEEE Access* **2019**, *7*, 76812–76832. [CrossRef]
12. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
13. Chien, H.Y.; Jan, J.; Tseng, Y.M. An efficient and practical solution to remote authentication: Smart card. *Comput. Secur.* **2002**, *21*, 372–375. [CrossRef]
14. Zhu, J.; Ma, J. A new authentication scheme with anonymity for wireless environments. *IEEE Trans. Cons. Elec.* **2004**, *50*, 231–235.
15. Lee, Y.; Kim, S.; Won, D. Enhancement of two-factor authenticated key exchange protocols in public wireless LANs. *Comput. Electr. Eng.* **2010**, *36*, 213–223. [CrossRef]
16. Kim, J.; Lee, D.; Jeon, D.; Lee, Y.; Won, D. Security anaylsis and improvements two-factor mutual authentication with key agreement in wireless sensor networks. *Sensors* **2014**, *14*, 6443–6462. [CrossRef] [PubMed]
17. Wang, D.; Wang, P. On the anonymity of two-factor authentication schemes for wireless sensor networks. *Comput. Netw.* **2014**, *73*, 41–57. [CrossRef]
18. Wang, D.; Li, W.; Wang, P. Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks. *IEEE Trans. Indust. Inform.* **2018**, *14*, 4081–4092. [CrossRef]
19. Wong, K.H.; Zheng, Y.; Cao, J.; Wang, S. A dynamic user authentication scheme for wireless sensor networks. *IEEE Inter. Conf. Sensor Netw. Ubiq. Trustworthy Comp.* **2006**, *1*, 1–8.
20. Li, X.; Peng, J.; Niu, J.; Wu, F.; Liao, J.; Choo, K.K.R. A robust and energy efficient authentication protocol for industrial internet of things. *IEEE Internet Things J.* **2018**, *5*, 1606–1615. [CrossRef]
21. Li, X.; Niu, J.; Kumari, S.; Wu, F.; Sangaiah, A.; Choo, K.K.R. A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments. *J. Netw. Comp. Appl.* **2018**, *103*, 194–204. [CrossRef]
22. Lee, J.; Yu, S.; Park, K.; Park, Y.; Park, Y. Secure three-factor authentication protocol for multi-gateway IoT environments. *Sensors* **2019**, *19*, 2358. [CrossRef]
23. Zhou, L.; Li, X.; Yeh, K.H.; Su, C.; Chiu, W. Lightweight IoT-based authentication scheme in cloud computing circumstance. *Future Gener. Comput. Syst.* **2019**, *91*, 244–251. [CrossRef]
24. Xue, K.; Hong, P.; Ma, C.A. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *J. Comput. Syst. Sci.* **2014**, *80*, 195–206. [CrossRef]
25. Amin, R.; Kumar, N.; Biswas, G.P.; Iqbal, R.; Chang, V. A lightweight authentication protocol for IoT-enabled devices in distributed cloud computing environment. *Future Gener. Comput. Syst.* **2018**, *78*, 1005–1019. [CrossRef]
26. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 6 May 2019).
27. SPAN: A Security Protocol Animator for AVISPA. Available online: http://www.avispa-project.org/ (accessed on 6 May 2019).
28. Park, K.; Park, Y.; Park, Y.; Reddy, A.G. ; Das, A.K. Provably secure and efficient authentication protocol for roaming service in global mobility networks. *IEEE Access* **2017**, *5*, 25110–25125. [CrossRef]
29. Park, K.; Park, Y.; Park, Y.; Das, A.K. 2PAKEP: Provably secure and efficient two-party authenticated key exchange protocol for mobile environment. *IEEE Access* **2018**, *6*, 30225–30241. [CrossRef]
30. Yu, S.; Lee, J.; Lee, K.; Park, K.; Park, Y. Secure authentication protocol for wireless sensor networks in vehicular communications. *Sensors* **2018**, *18*, 3191. [CrossRef] [PubMed]
31. Park, Y.; Park, Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123. [CrossRef] [PubMed]
32. Wu, F.; Xu, L.; Kumari, S.; Li, X.; Shen, J.; Choo, K.K.R.; Wazid, M.; Das, A.K. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. *J. Netw. Comput. Appl.* **2017**, *89*, 72–85. [CrossRef]