A Secure IoT-Based Authentication System in Cloud Computing Environment

Hsiao-Ling Wu¹, Chin-Chen Chang¹, Yao-Zhu Zheng², Long-Sheng Chen³,* and Chih-Cheng Chen⁴,⁵

¹ Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan; 590590@gmail.com (H.-L.W.); alan3c@gmail.com (C.-C.C.)
² Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan; s107062653@m107.nthu.edu.tw
³ Department of Information Management, Chaoyang University of Technology, Taichung 41349, Taiwan
⁴ Information and Engineering College, Jimei University, Fujian 361021, China; ccc@gm.cyut.edu.tw
⁵ Department of Industrial Engineering and Management, Chaoyang University of Technology, Taichung 413310, Taiwan

* Correspondence: lschen@cyut.edu.tw

Received: 18 August 2020; Accepted: 27 September 2020; Published: 30 September 2020

Abstract: The Internet of Things (IoT) is currently the most popular field in communication and information techniques. However, designing a secure and reliable authentication scheme for IoT-based architectures is still a challenge. In 2019, Zhou et al. showed that schemes proposed by Amin et al. and Maitra et al. are vulnerable to off-line guessing attacks, user tracking attacks, etc. On this basis, a lightweight authentication scheme based on IoT is proposed, and an authentication scheme based on IoT is proposed, which can resist various types of attacks and realize key security features such as user audit, mutual authentication, and session security. However, we found weaknesses in the scheme upon evaluation. Hence, we proposed an enhanced scheme based on their mechanism, thus achieving the security requirements and resisting well-known attacks.

Keywords: Internet of things (IoT); lightweight authentication; user anonymity; cloud computing

1. Introduction

With the rapid development of computer science and network technology, the concept of the Internet of Things (IoT) has become a hot topic for research. A scientist named Ashton introduced this concept in 1991. In IoT, numerous sensors have the capability of collecting data and communicating with each other or providing data for human beings through the Internet.

Therefore, technology can be widely used in the smart power grid, smart home, and other fields. In a smart grid, sensors monitor electric energy consumption and time-of-use rates for power stations. Then, the stations can optimize power supply. In the intelligent transportation system, sensors monitor traffic to optimize navigation. In the smart home, users can control, monitor, and access items remotely. Though IoT is close to our lives, it suffers from security challenges due to the wireless nature of the communication channel [1].

In order to protect against those security challenges in IoT, authentication is indispensable. Authentication guarantees that the messages received by the receiver are from a legal message sender. It serves as the first line of defense against potential attackers. Authentication is considered the key requirement for IoT [2]. The cryptography in authentication falls into two broad categories: symmetric encryption and asymmetric encryption. Common asymmetric encryption includes elliptic-curve cryptography (ECC) and RSA encryption.
Asymmetric encryption uses pairs of keys, i.e., public key and private key. Although, asymmetric encryption is generally considered to have higher security, it requires a higher computational cost. On the other hand, common symmetric encryption, e.g., the advanced encryption standard (AES) and data encryption standard (DES), use a shared key between two or more parties. Symmetric encryption has the advantages of low computational cost and fast encryption speed. Some authentication schemes have been recently presented by using asymmetric encryptions [3–10]. However, traditional asymmetric encryptions do not suit IoT devices due to limited resources of most IoT devices, which gives rise to lightweight authentication schemes [11–21].

To solve security disadvantages, many lightweight authentication schemes have been proposed. In 1981, Lamport [22] first suggested lightweight authentication using a password. The scheme also uses hash chains to go through unsafe communication channel for remote user authentication. However, the scheme relies on a password table, which makes it very easy to steal personal data. After that, many user authentications with a password and key negotiation techniques have been put forward [23–30]. In 2007, Liao et al. [31] proposed an authentication scheme based on a hash function for a multi-server environment. Further, Hsiang et al. [32] pointed out that Liao et al.’s scheme [31] is subject to multiple security threats, e.g., insider attack, masquerade attack, and user/server forgery attacks. Hsiang et al. [32] then proposed a new authentication scheme and claimed their scheme has fewer computations and higher security. In 2011, Sood et al. [33] proposed an authentication scheme using a dynamic identity for multi-server circumstances and criticized Hsiang et al.’s scheme [32] for having a wrong password change phase and not resisting replay and impersonation attacks. In the same year, Lee et al. [34] assessed Sood et al.’s programme [33] and concluded that it was not safe. In 2014, Xue et al. [35] pointed out that Lee et al.’s scheme [34] failed under the circumstances of pseudonym attack and offline password guessing attack. Later, Amin et al. [36] criticized the scheme in [35], saying that it lacked identity hiding features and could not resist offline password guessing attack. Recently, some authentication schemes are also used in vehicular ad-hoc networks (VANETs) [37–40] or smart grid [41]. It shows the universality of authentication. In 2019, Zhou and other [42] proposed their scheme based on a hash function and exclusive or operation of the two-factor authentication scheme, claiming their authentication scheme has been proven safe and could resist various attacks.

We reviewed the scheme of Zhou et al. [42] and pointed out the weaknesses such as the inability of replay attacks to achieve user anonymity and provide mutual authentication. We proposed an improved scheme that has a better balance between efficiency and security. Therefore, the scheme is more suitable for IoT based environment. The contribution of this paper is to enhance the resistance to replay attack, thus improving user anonymity and providing mutual authentication based on Zhou et al.’s scheme [42].

The rest of this article is arranged as follows: Section 2 provides an overview of Zhou et al. ‘s scheme, focusing on its registration and certification phases. Then, the security analysis of the scheme proposed by Zhou et al. [42] was conducted. Section 3 introduces the scheme we proposed. Safety analysis and performance evaluation are described in Sections 4 and 5. Section 6 gives the conclusion.

2. Related Works

In Section 2.1., we will introduce the authentication scheme proposed by Zhou et al. [42]. In addition, we will present the security issues of Zhou et al.’s scheme in Section 2.2.

2.1. Review of Zhou et al.’s Scheme

Zhou et al.’s scheme is divided into three stages: registration, authentication, and password modification. Here, we introduce the first two phases.

2.1.1. Registration Phase

There are two parts in this phase: user registration and cloud server registration.
User Registration

First, user $U_i$ selects four values (i.e., identity $ID_i$, pseudo-identity $PID_i$, password $PW_i$, and a random number $b_i$) to calculate $HP_i = h(PW_i || b_i)$. The $U_i$ then sends the $ID_i$ and $PID_i$ to the control server $CS$. When $CS$ receives $(ID_i, PID_i)$, $CS$ will check whether or not $ID_i$ is in the database. If not, $CS$ uses secret key $x$ to calculate $C_1^* = h(PID_i || IDcs || x)$ and $C_2^* = h(ID_i || x)$; otherwise, $CS$ will stop the authentication. $CS$ stores $ID_i$ in its database and sends $(C_1^*, C_2^*, IDcs)$ to $U_i$. When $U_i$ receives $(C_1^*, C_2^*, IDcs)$, $U_i$ calculates three values, $C_1 = C_1^* \oplus HP_i$, $C_2 = C_2^* \oplus h(ID_i || HP_i)$, and $C_3 = b_i \oplus h(ID_i || PW_i)$, then stores $(C_1, C_2, C_3, PID_i, IDcs)$ in a smart card.

Cloud Server Registration

Cloud server $S_j$ sends $(SID_j, PSID_j)$ to $CS$, where $SID_j$ is the identity of $S_j$ and $PSID_j$ is the pseudo-identity of $S_j$. When $CS$ receives $(SID_j, PSID_j)$, $CS$ calculates $B_1 = h(PSID_j || IDcs || x)$ and $B_2 = h(SID_j || x)$. Finally, $CS$ stores $SID_j$ in a database and sends $(B_1, B_2, IDcs)$ to $S_j$, and $S_j$ stores $(B_1, B_2, SID_j, PSID_j, IDcs)$ in a memory.

2.1.2. Authentication Phase

When user $U_i$ wants to connect with a cloud server, the user will perform the following five steps with the cloud server ($S_j$) and the control server ($CS$).

Step 1: User inputs his $ID_i$ and $PW_i$. A smart card will select a random number $r_u$ and new pseudo-identity $PID_i^{new}$; then, it calculates $b_i = C_3^* \oplus h(ID_i || PW_i)$, $H_{PID} = h(PW_i || b_i)$, $C_1^* = C_1^* \oplus HP_i$, and $C_2^* = C_2^* \oplus h(ID_i || HP_i)$. The smart card then calculates $D_1 = C_1^* \oplus r_u$, $D_2 = h(r_u || PID_i || IDcs || ID_i)$, $D_3 = C_2^* \oplus h(ID_i || HP_i)$ $\oplus PID_i^{new} \oplus h(r_u || ID_i)$, and $D_4 = h(ID_i || PID_i || PID_i^{new} || r_u || D_5)$. $U_i$ sends the message $M_1 = \{PID_i, D_1, D_2, D_3, D_4\}$ to $S_j$.

Step 2: When $S_j$ receives $M_1$, $S_j$ selects a new pseudo-identity $PID_j^{new}$ and a random number $r_s$ to calculate $D_5 = B_2 \oplus r_s$, $D_6 = h(r_s || PSID_i || IDcs || SID_j)$, $D_7 = B_3 \oplus PID_j^{new} \oplus h(r_u || PSID_j)$, and $D_8 = h(SID_j || IDcs || PSID_i || PID_j^{new} || r_s || D_7)$. $S_j$ sends the message $M_2 = \{M_1, PSID_i, D_5, D_6, D_7, D_8\}$ to $CS$.

Step 3: When $CS$ receives $M_2$, $CS$ calculates $r_u = D_1 \oplus h(PID_i || IDcs || x)$, $ID_i = D_2 \oplus h(r_u || PID_i || IDcs)$, and $PID_j^{new} = D_3 \oplus h(ID_i || x) \oplus h(r_u || ID_i)$. $CS$ checks whether $ID_i$ is in the database and $D_9 = h(ID_i || PID_i || PID_j^{new} || r_u || D_5)$. If $ID_i$ is in the database and $D_9 = h(ID_i || PID_i || PID_j^{new} || r_u || D_5)$, it means that $CS$ confirms $U_i$ is a legal user. Otherwise, the authentication process will be terminated. Then, $CS$ calculates $r_s = D_5 \oplus h(PSID_i || IDcs || x)$, $SID_j = D_6 \oplus h(r_s || PSID_i || IDcs)$, and $PSID_j = D_7 \oplus h(SID_j || x) \oplus h(r_u || SID_j)$. $CS$ checks whether $SID_j$ is in database and $D_8 = h(SID_j || IDcs || PSID_i || PID_j^{new} || r_u || D_7)$, it means that $CS$ confirms the $S_j$ is legal. Then, $CS$ selects a random number $r_c$ to calculate the session key $SK = h(r_u || r_s || r_c)$, $D_9 = h(PID_j^{new} || IDcs || x) \oplus h(r_u || PSID_j)$, $D_{10} = h(PID_j^{new} || r_u || PSID_j)$ $\oplus h(r_s || r_c)$, $D_{11} = h(SK || C_3 || D_{10} || h(ID_i || x))$, $D_{12} = h(PID_j^{new} || IDcs || x) \oplus h(r_u || PID_j^{new})$, $D_{13} = h(PID_j^{new} || IDcs || x) \oplus h(r_u || PID_j^{new})$, and $D_{14} = h(SK || C_3 || D_{13} || h(ID_i || x))$. $CS$ sends the message $M_3 = \{D_9, D_{10}, D_{11}, D_{12}, D_{13}, D_{14}\}$ to $S_j$.

Step 4: When $S_j$ receives $M_3$, $S_j$ calculates $(r_u || r_s || r_c)$ and computes $SK = h(r_u || r_s || r_c)$. Then, $S_j$ checks $D_{11} = h(SK || C_3 || D_{10} || B_2)$ to confirm that $S$ is a legal control server or not. If $CS$ is a legal control server, $S_j$ calculates $B_1^{new} = D_9 \oplus h(r_u || PID_j^{new})$, updates $B_1$ and $PSID_j$ as $B_1^{new}$ and $PID_j^{new}$ in memory. $S_j$ sends message $M_4 = \{D_{12}, D_{13}, D_{14}\}$ to $U_i$.

When $U_i$ receives $M_4$, $U_i$ calculates $(r_u || r_s || r_c) = D_{13} \oplus h(PID_j^{new} || r_u || PID_j)$ and checks that $SK = h(r_u || r_s || r_c)$, then $U_i$ calculates $C_1^{new} = D_{12} \oplus h(r_u || PID_j^{new}) \oplus HP_i$, updates $C_1$ and $PID_j$ in memory to $C_1^{new}$ and $PID_j^{new}$.

2.2. Analysis of Zhou et al.’s Scheme

We found three weaknesses in Zhou et al.’s scheme at the certification stage. First, Zhou et al.’s scheme cannot achieve mutual authentication. Second, Zhou et al.’s scheme cannot work against a replay attack. Third, Zhou et al.’s scheme cannot guarantee anonymity in the authentication phase.
2.2.1. Zhou et al.’s Scheme Cannot Achieve Mutual Authentication

Mutual authentication refers to the mutual verification between two entities. In Zhou et al.’s scheme, CS verifies \( U_1 \) by checking \( D_2 = h(ID_1 || PID_1 || PID_{new} || r_u || D_3) \) in Step 3 of the authentication phase. We know \( D_3 = C^1_2 \oplus h(ID_1 || HP_j || PID_{new} || r_u || ID_i) \) and \( C^2_2 = h(ID_1 || x) \) from Step 1 of the authentication phase and the user registration. When CS computes \( D_3 = h(ID_1 || x) \oplus h(r_u || ID_i) \), CS only can obtain \( h(D_3 || HP_j || PID_{new}) \), where the parameter \( HP_j \) is only known by \( U_i \). CS cannot successfully calculate \( PID_{new} \) from \( D_3 = h(ID_1 || x) \oplus h(r_u || ID_i) \), even if the message \( M_1 = \{ PID_i, D_1, D_2, D_3, D_4 \} \) is sent from a legal user \( U_i \). Therefore, Zhou et al.’s scheme was unable to complete mutual authentication.

2.2.2. Zhou et al.’s Scheme Cannot Guarantee Anonymity in Authentication Phase

A solution that provides anonymity must ensure that no one except the server knows the user’s personal information. We assume that the attacker \( U_A \) is a legitimate user. Hence, \( U_A \) will obtain \( C^1_2 = h(PID_A || ID_{cs} || x) \), \( C^2_2 = h(ID_A || x) \), \( ID_{cs} \) from CS in the user registration phase. Once \( U_A \) intercepts the message \( M_1 = \{ PID_i, D_1, D_2, D_3, D_4 \} \) from \( U_i \) and uses \( PID_i \) as a new pseudo-identity to restart an authentication session, \( U_A \) can obtain the \( ID_1 \) of the user \( U_i \). Details of the process are as follows.

Step 1: First, \( U_A \) chooses a random number \( r_A \) to calculate \( \overline{D_1} = C^1_1 \oplus r_A \), \( \overline{D_2} = h(r_A || PID_A || ID_{cs} || ID_A) \), \( \overline{D_3} = C^2_2 \oplus h(ID_A || HP_A || PID_i || ID_A) \), and \( \overline{D_4} = h(ID_A || PID_A || ID_A || r_u || D_3) \). \( U_A \) sends the message \( M_2 = \{ \overline{PID_A}, \overline{D_1}, \overline{D_2}, \overline{D_3}, \overline{D_4} \} \) to \( S_j \).

Step 2: When \( U_A \) receives \( M_4 = \{ \overline{D_{12}}, \overline{D_{13}}, \overline{D_{14}} \} \), \( U_A \) can compute \( ID_i = D_2 \oplus h(D_1 \oplus D_{12} \oplus h(r_A || PID_i) || PID_A || ID_{cs}) \), where \( D_1 = h(PID_i || ID_{cs} || x) \oplus r_u \), \( D_2 = h(r_u || PID_i || ID_{cs}) \oplus ID_A \), and \( D_{12} = h(PID_i || ID_{cs} || x) \oplus h(r_A || PID_i) \).

Therefore, Zhou et al.’s scheme cannot guarantee anonymity in the authentication phase.

3. Proposed Scheme

After we reviewed the shortcomings of Zhou et al.’s scheme, an improved scheme is put forward. The improvements include registration, authentication, and password modification.

3.1. Notations

The following is the introduction to the notations that will be used in our scheme.

- \( U_i \) is the \( i \)th user.
- \( ID_i \) is the \( i \)th user’s identity.
- \( PW_i \) is the \( i \)th user’s password.
- \( n_i \) is a random number.
- CS is the control server.
- \( PID_i \) is the \( i \)th user’s pseudo-identity.
- \( ID_{cs} \) is the control server’s identity.
- \( SID_j \) is the \( j \)th server’s identity.
- \( PSID_j \) is the \( j \)th server’s pseudo-identity.
- \( x \) is the secret key of CS.
- \( h() \) is a one-way hash function.
- \( r_u, r_h, r_{cs} \) are the random numbers selected by \( U_i, S_j, \) and CS.
- \( SK_u, SK_s, SK_{cs} \) are the session keys for \( U_i, S_j, \) and CS.
- \( M_1, M_2, M_3, M_4 \) are the messages in the authentication.

3.2. Registration Phase

This phase is divided into two parts: user registration and cloud server registration. When a user or a cloud server wants to join this system, he/she must run this phase first. After the user and the cloud server successfully finish this phase, they can connect with each other to start the authentication phase.
3.2.1. User Registration

User $U_i$ selects their own id $ID_i$, password $PW_i$, random number $n_i$. He/she sends $ID_i$ to CS by the secure channel. When CS receives $ID_i$, CS checks it for its validity. If it is invalid, CS will stop this phase; otherwise, CS selects a pseudo-identity $PID_i$ for $U_i$ and uses the secret key $x$ to compute $A_i = h(PID_i||ID_c||x)$ and $B_i = h(ID_i||x)$. CS stores $ID_i$ in its database and sends $(A_i, B_i, PID_i, ID_c)$ to $U_i$ by the secure channel. Once $U_i$ obtains these parameters, $U_i$ calculates $C_1 = A_i \oplus h(ID||n_i)$, $C_2 = B_i \oplus h(PW||n_i)$, $C_3 = n_i \oplus h(ID||PW_i)$, and $C_4 = h(ID||PW_i||n_i)$ and then stores $(C_1, C_2, C_3, C_4, PID_i, ID_c)$ in a smart card. The flowchart for user registration is shown in Figure 1.

![Figure 1. The flowchart of the user registration phase.](image)

3.2.2. Cloud Server Registration

A cloud server $S_j$ sends its identity $SID_j$ and a pseudo-identity $PSID_j$ to CS by a secure channel. Then, CS uses the secret key $x$ to compute $A_j = h(PSID_j||ID_c||x)$ and $B_j = h(SID_j||x)$, stores $SID_j$ in its database, and sends $(A_j, B_j, ID_c)$ to $S_j$ by a secure channel. When $S_j$ receives these parameters, $S_j$ stores $(A_j, B_j, SID_j, SPID_j, ID_c)$ in its memory. The flowchart of the cloud server registration phase is shown in Figure 2.

![Figure 2. The flowchart of the cloud server registration phase.](image)

3.3. Authentication Phase

When the user $U_i$ needs to retrieve services from the cloud server $S_j$, this authentication must start to make sure of the legitimacy of both the user and the cloud server. After the authentication phase is completed, the user will negotiate a session key $SK$. By this session key, $U_i$ can connect with $S_j$ securely. The processes of the authentication phase are shown as follows and Figure 3.
3.2. Authentication Phase

When the user \( U_i \) needs to retrieve services \( \text{from the cloud server} \), \( \text{he/she inserts the smart card into a reader machine and keys in} \) \( ID_i \) and \( PW_i \). Then, the smart card selects a random number \( r_u \) and calculates \( n_u = C_3 \oplus h(ID_i || PW_i) \). Then, the smart card checks \( h(ID_i || PW_i || n_u) = C_4 \) to verify the identity and password. If the verification passed, the smart card will calculate \( A_1 = C_1 \oplus h(ID_i || n_u) \), \( B_i = C_2 \oplus h(PW_i || n_u) \), \( D_1 = A_1 \oplus r_u \), \( D_2 = h(r_u || PID_i || ID_{cs} || r_u) \oplus ID_i \), and \( D_3 = h(ID_i || PID_i || r_u) \). Finally, the smart card sends \( M_1 = \{ \text{PID}_i, D_1, D_2, D_3 \} \) to \( S_j \).

Step 2: When \( S_j \) receives \( M_1 \), \( S_j \) selects a new pseudo-identity \( \text{PSID}'_j \) and a random number \( r_s \) to calculate \( D_3 = h(ID_j || \text{PSID}'_j || ID_{cs} || r_s) \). Then, \( S_j \) checks whether \( D_3 \) is valid and \( D_7 = h(ID_j || PID_j || r_s) \) or not. If the ID \( D_3 \) is in its database and \( D_3 = h(ID_j || PID_j || r_s) \), it means that \( U_i \) is legal. For the cloud server \( S_j \), \( CS \) uses the secret key \( x \) to compute \( r_s = D_3 \oplus h(\text{PSID}'_j || ID_{cs} || x) \) and \( ID_j = D_2 \oplus h(r_s || PID_j || ID_{cs}) \) and then checks whether \( ID_j \) is valid and \( D_7 = h(ID_j || PID_j || r_s) \) or not. If the ID \( D_7 \) is in its database and \( D_7 = h(ID_j || PID_j || r_s) \), then \( S_j \) selects a new pseudo-identity \( \text{PSID}'_j \) and a random number \( r_s \) to calculate \( D_3 = h(ID_j || \text{PSID}'_j || ID_{cs} || r_s) \). Then, \( S_j \) sends message \( M_2 = \{ \text{PID}_j, D_4, D_5, D_6, D_7 \} \) to CS.

Step 3: Once \( CS \) receives \( M_2 \), \( CS \) uses the secret key \( x \) to compute \( r_s = D_3 \oplus h(\text{PSID}'_j || ID_{cs} || x) \), \( SJD = D_5 \oplus h(\text{PSID}'_j || ID_{cs} || x) \), and then checks whether \( SJD \) is in the database and \( D_7 = h(ID_j || PID_j || r_s) \) or not.
$h(SID)||PSID||PSID'||r_s||D_6)$. If both conditions hold, it means that $S_j$ is legal. The processes of authentication phase will be stopped when any verification is wrong; otherwise, $CS$ selects a random number $r_cs$ to compute the session key $SK_{cs} = h(r_s\oplus r_t \oplus r_cs)$ for this round. Subsequently, for $S_j$, $CS$ computes $D_8 = h(PSID'||ID_{cs}\oplus h(r_s||PSID'))$, $D_9 = h(PSID'|||r_s||PSID)\oplus (r_s \oplus r_cs)$, and $D_{10} = h(SK_{cs}||D_8||D_9||h(SID||x))$. For $U_i$, $CS$ selects a new pseudo-identity $PID'_i$ to compute $D_{11} = PID'_i \oplus h(ID||x) \oplus h(r_u||ID_{j3})$, $D_{12} = h(PID'_i ||ID_{cs}||x) \oplus h(r_u||PID'_i)$, $D_{13} = h(PID'_i ||r_u||PID) \oplus (r_s \oplus r_cs)$, and $D_{14} = h(SK_{cs}||D_{12}||D_{13}||h(ID||x))$. Finally, $CS$ sends the message $M_3 = \{D_8, D_9, D_{10}, D_{12}, D_{13}, D_{14}\}$ to $S_j$.

Step 4: While $S_j$ receives $M_3$, $S_j$ uses $PSID'$ and $r_s$ to extract $(r_u \oplus r_cs) = D_9 \oplus h(PSID'||r_s||PSID)$. Then, $S_j$ checks $D_{10} = h(SK_{cs}||D_8||D_9||B_i)$, where $SK_s = h(r_s \oplus r_t \oplus r_cs)$. If this equation holds, it means that $CS$ is legal; otherwise, this authentication process will be terminated. $S_j$ continues to calculate $A'_j = D_8 \oplus h(r_u||PSID'_j)$ and updates $A_j$ and $PSID_j$ as $A'_j$ and $PID'_j$ in the memory. At the end of this step, $S_j$ sends the message $M_4 = \{D_{11}, D_{12}, D_{13}, D_{14}\}$ to $U_i$.

Step 4: Once the smart card receives $M_4$, the smart card uses $B_i$, $r_u$, and $ID_i$ to extract $PID'_i$ and $(r_s \oplus r_cs)$ from $D_{11}$ and $D_{13}$, respectively. That is, $PID'_i = B_i \oplus D_{11} \oplus h(r_u||ID_i)$ and $(r_s \oplus r_cs) = D_{13} \oplus h(PID'_i ||r_u||PID_i)$. The smart card will check whether or not $D_{14} = h(SK_{cs}||ID_{12}||D_{13}||B_i)$, where $SK_u = h(r_u \oplus r_t \oplus r_cs)$. If this equation holds, it means that $CS$ is legal; otherwise, this authentication process will be terminated. The smart card uses the new pseudo-identity $PID'_i$ to calculate $C'_i = D_{12} \oplus h(ID||PID'_i)$, and updates $C_i$ and $PID_i$ as $C'_i$ and $PID'_i$. Finally, the smart card sends $h(SK_{cs})$ to $S_j$.

Step 5: When $S_j$ receives $h(SK_{cs})$, $S_j$ will check $h(SK_{cs})? = h(SK_c)$. If $h(SK_{cs}) = h(SK_c)$, this means that they already correctly negotiate the session key.

### 3.4. Password Change Phase

If the user $U_i$ needs to change the password, you must need to start the password change phase. First, we assume that the smart card of $U_i$ contains $(C'_1, C_2, C_3, C_4, PID'_i, ID_{cs})$. The $U_i$ inserts the smart card into the card reader for key verification in identity $ID_i$ and the original password $PW_i$. The smart card will calculate $n_i = C_3 \oplus h(ID||PW_i)$ and check $h(ID||PW_i||n_i) = C_4$. If the equation holds, $U_i$ can input the new password $PW'_i$. The smart card calculates $C'_2 = C_2 \oplus h(PW'_i||n_i) \oplus h(PW'_i||n_i)$, $C_3 = C_3 \oplus h(ID||PW_i||n_i) \oplus h(ID||PW'_i||n_i)$, and $C'_4 = C_4 \oplus h(ID||PW_i||n_i) \oplus h(ID||PW'_i||n_i)$ and replaces $(C_2, C_3, C_4)$ with $(C'_2, C'_3, C'_4)$. Finally, there are $(C'_1, C'_2, C'_3, C'_4, PID'_i, ID_{cs})$ in the smart card, and $U_i$ can use the new password $PW'_i$ to perform the authentication phase in the next round. The flowchart of password modification phase is shown in Figure 4.

![Figure 4](image-url)
4. Security Analysis

In this section, we will analyze nine fundamental security requirements in which an authentication scheme should be achieved.

4.1. Mutual Authentication

As we discussed in Section 2.2.1., mutual authentication means that the identities of the two entities should be recognized before they connect. In our scheme, CS can be mutually authenticated with $U_i$ and $S_j$, respectively.

4.1.1. CS Verifies the Identity of $U_i$ through Checking $D_3$?

In the user registration phase, CS computes $A_i = h(PID_i||ID_{cs}||x)$ and $B_i = h(ID_i||x)$ for $U_i$, and two parameters are only known by CS and $U_i$. When $U_i$ uses $A_i$ to hide the random number $r_u$ in the authentication phase, i.e., $D_1 = A_i \oplus r_u$, CS can use $h(PID_i||ID_{cs}||x)$ to extract $r_u$. Finally, CS can verify the identity of $U_i$ by equation $D_3 = h(ID_i||PID_i||r_u)$.

4.1.2. CS Verifies the Identity of $S_j$ through Checking $D_7$?

In the cloud server registration phase, CS computes $A_j = h(PSID_j||ID_{cs}||x)$ and $B_j = h(SID_j||x)$ for $S_j$, and two parameters are only known by CS and $S_j$. When $S_j$ uses $A_j$ to hide the random number $r_s$ in the authentication phase, i.e., $D_4 = A_j \oplus r_s$, CS can use $h(PSID_j||ID_{cs}||x)$ to extract $r_s$. Finally, CS can verify the identity of $S_j$ by equation $D_7 = h(SID_j||PSID_j||ID_{cs}||r_s||D_6)$.

4.1.3. $S_j$ Verifies the Identity of CS through Checking $D_{10}$? = $h(SK_s||D_8||D_{9}||B_j)$

Because $B_j$ is only shared between $S_j$ and CS, they only have the capability of computing $h(SK_s||D_8||D_{9}||B_j)$. Therefore, $S_j$ can verify the identity of CS by equation $D_{10} = h(SK_s||D_8||D_{9}||B_j)$.

4.1.4. $U_i$ Verifies the Identity of CS through Checking $D_{14}$? = $h(SK_u||D_{12}||D_{13}||B_i)$

Because $B_i$ only shares between $U_i$ and CS, they only have the capability of computing $h(SK_u||D_{12}||D_{13}||B_i)$. Therefore, $U_i$ can verify the identity of CS by equation $D_{14} = h(SK_u||D_{12}||D_{13}||B_i)$.

4.2. Session Key for All Entities

In the authentication phase, $U_i$, $S_j$, and CS generate $r_u$, $r_s$, and $r_{cs}$, respectively. In addition, $U_i$, $S_j$, and CS obtain $(r_u \oplus r_{cs})$, $(r_s \oplus r_{cs})$, and $(r_u, r_s, r_{cs})$ from $D_{13}$, $D_9$, and $(D_1, D_4)$, respectively. Therefore, all entities can compute one same session key $SK = SK_{cs} = SK_s = SK_u = (r_u \oplus r_s \oplus r_{cs})$ in one session.

4.3. User Anonymity

The attacker’s use of user anonymity means that the user $U_i$ cannot be identified through the messages in the communication session [43]. In our authentication phase, $U_i$’s identity $ID_i$ is protected by a hash function $D_2 = h(r_u||PID_i||ID_{cs}||r_u)$. Therefore, if an attacker wants to obtain $U_i$’s identity, he/she must compute $h(r_u||PID_i||ID_{cs})$. However, he/she cannot acquire the $r_u$ because he/she does not have the secret key $x$ of CS to derive $r_u$ from $D_1 = A_i \oplus r_u$, where $A_i = h(PSID_i||ID_{cs}||x)$. Even if the attacker is a legal user, he/she still cannot obtain $h(r_u||PID_i||ID_{cs})$ by adopting the strategy shown in Section 2.2.2. Therefore, the attacker cannot identify $U_i$’s identity; furthermore, it shows that our proposed scheme has user anonymity.

4.4. Resistance to Off-Line Guessing Attack

Off-line guesswork attacks happen when an attacker obtains all the information stolen from the user, pass through insecure channels, and store in smart CARDS. The attacker can use the information held to guess the user’s identity and password.
We assume that an attacker gets \((C_1, C_2, C_3, C_4, PID_i, ID cs)\) that is stored in the user \(U_i\)’s smart card and all messages \((M_1, M_2, M_3, M_4)\) that pass by a nonsecure channel in the last session. Then, the attacker wants to guess a pair \((ID_i, PW_i)\) from information. He/she can use the equation \(D_2 = h(r_u\|PID_i\|ID cs)\oplus ID_i\) to confirm her/his guess \(ID_i\). According to the above hypothesis, the attacker has \(PID_i\) and \(D_2\) from \(M_2\); \(ID cs\) is from the smart card. Therefore, he/she needs to get \(r_u\). Then, \(r_u\) can be derived by rearranging \(D_1 = A_i\oplus r_u\) to \(r_u = A_i\oplus D_1\). However, the attacker cannot compute \(A_i = h(PSID_i\|ID cs\|x)\) without the secret key \(x\) of \(CS\). Therefore, he/she cannot successfully guess \(ID_i\). In addition, \(PW_i\) only appears on \(C_2 = h(ID_i\|x)\oplus h(PW_i\|n_i), C_3 = n_i\oplus h(ID_i\|PW_i),\) and \(C_4 = h(ID_i\|PW_i\|n_i)\). If the attacker wants to guess it, he/she needs to obtain \(ID_i, x\) or \(n_i\) first. However, the attacker cannot extract those values from intercepted messages. Therefore, he/she cannot successfully guess \(PW_i\). The results show that the scheme can resist offline guessing attack.

4.5. Resistance to Insider Attack

An insider attack means that an attacker is an inside member of the company of \(CS\). He has the right to access the data stored in the \(CS\)’s database, e.g., the registered users’ identities and passwords. Then, he/she can use the information to simulate a legitimate user or cloud server. In our proposed scheme, only \(ID_i\) and \(SID_j\) are stored in \(CS\) for registration. There is no any other information for authentication stored in \(CS\), i.e., \(A_1, B_1, A_2, B_2\). Therefore, even if the inside attacker accesses the database of \(CS\), he/she only can obtain the identity \(ID_i\) of \(U_i\) and \(SID_j\) of \(S_j\); besides, the inside attacker still cannot impersonate the user \(U_i\) or the cloud server \(S_j\). Thus, the scheme is able to resist internal attack.

4.6. Resistance to Stolen Smart Card Attack

Stolen card attack points to an attacker who steals the user’s smart card and extracts data stored in a smart card. Then, he/she uses these data to impersonate the user whose smart card was stolen. Here, we assume that an attacker already extracts the data \((C_1, C_2, C_3, C_4, PID_i, ID cs)\) from user \(U_i\)’s smart card. In our proposed scheme, if the attacker wants to impersonate user \(U_i\), he/she needs to perform the authentication phase. According to the description of Step 1 in Section 3.2., the attacker needs to key in the correct \(ID_i\) and \(PW_i\) for checking the equation \(h(ID_i\|PW_i\|n_i) = C_4\). However, he/she does not have \(ID_i\) and \(PW_i\). Therefore, when the attacker initiates an authentication run, he/she cannot pass the check \(h(ID_i\|PW_i\|n_i) = C_4\) in this step, then his/her authentication process will be terminated. The results show that the scheme can resist the attack of stolen smart cards.

4.7. Resistance to De-Synchronization Attack

An anti-synchronization attack means that an attacker interrupts and modifies the response message from the control server during the authentication phase, so that the authentication data between the client and the database of the control server are not synchronized \([44]\). Then, even if he/she is a legitimate user passing through the controlled server, all future authentication processes will fail.

In our proposed scheme, only users’ identities are stored in the control server’s database. In addition, those identities will not be changed in any phases, i.e., the authentication and password change phases. For the user, data changes occurred in the authentication stage and the last step of the password change phase. However, password change only needs to be involved on the user side; thus, the attacker cannot interfere. In the last step of the authentication phase, the data in the user’s smart card will be updated \((C_1, PID_i)\) to \((C’_1, PID’_i)\) when authentication processes are successfully finished. If the update was interrupted, the user can still use the old data \((C_1, PID_i)\) to run a successful authentication process. It can be concluded that the scheme can resist synchronous attack.
4.8. Resistance to Forgery Attack

Counterfeit attack points to the attacker in the session is sent to the user, the cloud server and control server message, then the receiver will believe these messages are sent from a legal user, a cloud server, or the control server.

In our scenario, if an attacker wants to forge a user \( U_i \), he/she would need to forge a message \( M_1 \) to pass the equation \( D_3? = h(ID_i∥PID_i∥r_u) \). However, the attacker cannot forge \( D_1 = A_i∥r_u \) because \( A_i = h(PID_i∥ID_{cs}∥x) \) contains the secret key \( x \) of a control server. If the attacker wants to forge a cloud server, he/she needs to fabricate two messages, \( M_2 \) and \( M_4 \). To pass the equation \( D_7? = h(SID_i∥PSID_i∥ID_{cs}∥x) \) and \( D_9? = h(SK_{cs}∥D_{12}∥D_{13}∥B_j) \); however, he/she cannot forge \( D_4 = A_i∥r_u \), \( D_6 = B_i∥PSID_i∥ID_{cs}∥x \) because \( A_i \) and \( B_i \) both contain the secret key \( x \) of a control server. If the attacker wants to forge the control server, he/she needs to make up a message \( M_3 \) to pass the equation \( D_{10}? = h(SK_{cs}∥D_8∥D_9∥B_j) \). However, he/she cannot forge \( D_8 = h(PSID_i∥ID_{cs}∥x)∥h(r_u∥PSID_i) \) and \( D_{10} = h(SK_{cs}∥D_8∥D_9∥h(SID_i∥x)) \) because those messages contain the secret key \( x \) of the control server. As a result, we provide a solution to staying away from forgery attacks.

4.9. Resistance to User Tracking Attack

In terms of user tracking attacks, when an attacker eavesdrops on the delivered messages in different sessions, and then the attacker can confirm that two messages are from a fixed user according to a stable pseudo-identity being used. In our proposed scenario, the user \( U_i \)’s pseudo-identity would change in different sessions. Therefore, the attacker cannot ensure that any two messages are from the same user. The results show that the scheme can resist the user tracking attack.

5. Performance Evaluation

In this section, we will present the schemes of Maitra et al. [45], Amin et al. [36], Zhou et al. [42], and the performance evaluation of our schemes. Four authentication schemes only use a one-way hash operation, exclusive or operation, and concatenate operation. By comparing the execution time of an exclusive or operation to that of a one-way hash function or a symmetric algorithm, we ignored the execution time of an exclusive or operation. We chose SHA-2(256 bits) and AES as one-way hash functions and symmetric encryption/decryption algorithms, two of which are the most commonly used encryption methods in secure communications.

Tables 1–3 show a comparison of the security properties, computation cost, and communication cost among four respective authentication schemes. In Table 1, “O” means that the scheme can achieve a security requirement or resist the attack; “X” means that the scheme cannot achieve a security requirement or resist the attack. In Table 2, “\( T_h \)” is one computation time of one-way hash function operation, and “\( T_s \)” is one computation time of symmetric encryption/decryption. The “\( T_h \)” and “\( T_s \)”’s values are 0.00517 ms and 0.02148 ms, respectively according to Zhou et al. [42].

| Property | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 | R9 |
|----------|----|----|----|----|----|----|----|----|----|
| Amin et al.’s scheme [36] | O  | X  | O  | O  | O  | O  | O  | O  | X  |
| Maitra et al.’s scheme [45] | O  | X  | O  | O  | O  | O  | O  | O  | X  |
| Zhou et al.’s [42] | X  | O  | X  | O  | O  | O  | O  | O  | O  |
| Ours | O  | O  | O  | O  | O  | O  | O  | O  | O  |

R1: Mutual authentication. R2: Session key for all entities. R3: User anonymity. R4: Resistance to off-line guessing attack. R5: Resistance to insider attack. R6: Resistance to stolen smart card attack. R7: Resistance to de-synchronization attack. R8: Resistance to forgery attack. R9: Resistance to user tracking attack.
The proposed scheme is more suitable for the Internet of Things environment than other related schemes. Sensors 2020, x FOR PEER REVIEW The scheme has acceptable computation and good security. Therefore, we believe that this new certification scheme to compensate for Zhou et al.’s scheme. The proposed scheme can resist common attacks and provide important features such as user anonymity and mutual authentication. Then, we designed during authentication. The Figure 5 shows the bar chart of the comparison of total calculation cost. In this paper, we demonstrated that Zhou et al.’s scheme is not fully secure. Mutual authentication for cloud computing is also proposed, and the performance evaluation results show whether or not the input identity and password are right at an early stage. Improved IoT-based authentication scheme is applicable to real-world IoT devices. As can be seen from Table 1, the scheme proposed by us has better security than other schemes. We also assessed the communication costs of our scheme and other schemes, as shown in Table 3. The communication costs are the bits of parameters which passed during authentication. The Figure 5 shows the bar chart of the comparison of total calculation cost. Our scheme gets more cost than Zhou et al.’s [42] because we add an additional step at the last of the authentication phase to achieve mutual authentication. We only calculate the communication cost in the login and authentication phases due to the use of fewer number of times in the registration phase and password change phase. Therefore, in terms of security and efficiency, we can argue that our proposed scheme is more suitable for the Internet of Things environment than other related schemes.

### Table 2. Calculation cost comparison of four certification schemes.

| Entities        | Registration Phase | Login Phase | Authentication Phase | Password Change Phase | Total Operations of Login and Authentication |
|-----------------|--------------------|-------------|----------------------|-----------------------|---------------------------------------------|
| Amin et al.’s scheme [36] | U_i 2 T_h | 6 T_h | 3 T_h | 7 T_h | 23 T_h |
|                 | S_i 0 T_h | 0 T_h | 4 T_h | 0 T_h |               |
|                 | CS 4 T_h | 0 T_h | 10 T_h | 0 T_h |               |

| Maitra et al.’s scheme [45] | U_i 3 T_h | 6 T_h | 4 T_h | 9 T_h | 19 T_h + 6 T_s |
|                             | S_i 0 T_h | 0 T_h | 0 T_h + 1 T_s | 0 T_h |               |
|                             | CS 3 T_h + 1 T_s | 0 T_h | 5 T_h + 3 T_s | 2 T_h + 2 T_s |               |

| Zhou et al.’s [42]       | U_i 4 T_h | 0 T_h | 12 T_h | 6 T_h | 39 T_h |
|                          | S_i 0 T_h | 0 T_h | 8 T_h | 0 T_h |               |
|                          | CS 4 T_h | 0 T_h | 19 T_h | 0 T_h |               |

| Ours                       | U_i 4 T_h | 0 T_h | 12 T_h | 6 T_h | 39 T_h |
|                           | S_i 0 T_h | 0 T_h | 8 T_h | 0 T_h |               |
|                           | CS 4 T_h | 0 T_h | 19 T_h | 0 T_h |               |

| Total Operations of Login and Authentication | 23 T_h | 19 T_h + 6 T_s | 36 T_h | 39 T_h |

Table 2 shows that our proposed scheme is in the middle regarding calculating costs. However, it is important to consider the trade-off between security and efficiency when we were designing a secure communication scheme. As can be seen from Table 1, the scheme proposed by us has better security than other schemes. We also assessed the communication costs of our scheme and other schemes, as shown in Table 3. The communication costs are the bits of parameters which passed during authentication. The Figure 5 shows the bar chart of the comparison of total calculation cost. Our scheme gets more cost than Zhou et al.’s [42] because we add an additional step at the last of the authentication phase to achieve mutual authentication. We only calculate the communication cost in the login and authentication phases due to the use of fewer number of times in the registration phase and password change phase. Therefore, in terms of security and efficiency, we can argue that our proposed scheme is more suitable for the Internet of Things environment than other related schemes.

### Table 3. Communication cost comparison of four authentication schemes.

| Schemes                      | Communication Cost of L and A |
|------------------------------|------------------------------|
| Amin et al.’s scheme [36]   | 4736 bits                    |
| Maitra et al.’s scheme [45] | 3072 bits                    |
| Zhou et al.’s [42]          | 5760 bits                    |
| Ours                         | 6016 bits                    |

Figure 5. Comparison of total calculation cost (ms).
Note that the outputs of the one-way hash function and the AES algorithm are 256 bits, and identities, pseudo-identities, and random numbers are 128 bits.

6. Conclusions

In this paper, we demonstrated that Zhou et al.’s scheme is not fully secure. Mutual authentication and anonymity cannot be guaranteed in the authentication phase. Then, we designed a new certification scheme to compensate for Zhou et al.’s scheme. The proposed scheme can resist common attacks and provide important features such as user anonymity and mutual authentication. We also added a new parameter in the first step of the authentication phase; moreover, it can detect whether or not the input identity and password are right at an early stage. Improved IoT-based authentication for cloud computing is also proposed, and the performance evaluation results show that the scheme has acceptable computation and good security. Therefore, we believe that this authentication scheme is applicable to real-world IoT devices.

In the future, we will investigate how to apply our IoT-based authentication mechanism in different computing environments, such as mobile environment and grid computing environment, etc. Furthermore, we are investigating how to make our system lightweight so that it can be widely used in the mobile computing world.

Author Contributions: Conceptualization, H.-L.W.; Data curation, H.-L.W.; Formal analysis, C.-C.C. (Chin-Chen Chang); Investigation, C.-C.C. (Chin-Chen Chang) and L.-S.C.; Methodology, C.-C.C. (Chin-Chen Chang); Project administration, L.-S.C.; Resources, Y.-Z.Z. and L.-S.C.; Software, Y.-Z.Z.; Validation, Y.-Z.Z. and C.-C.C. (Chih-Cheng Chen); Visualization, C.-C.C. (Chih-Cheng Chen); Writing—review & editing, C.-C.C. (Chih-Cheng Chen). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Jing, Q.; Vasilakos, A.V.; Wan, J. Security of the Internet of Things: Perspectives and challenges. *Wirel. Netw.* 2014, 20, 2481–2507. [CrossRef]
2. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* 2010, 54, 2787–2805. [CrossRef]
3. Qi, M.; Chen, J.; Chen, Y. A secure authentication with key agreement scheme using ECC for satellite communication systems. *Int. J. Satell. Commun. Netw.* 2019, 37, 234–244. [CrossRef]
4. Kothmayr, T.; Schmitt, C.; Hu, W.; Brüning, M.; Carle, G. DTLS based security and two-way authentication for the Internet of Things. *Ad Hoc Netw.* 2013, 11, 2710–2723. [CrossRef]
5. Pranata, I.; Athauda, R.; Skinner, G. Securing and governing access in ad-hoc networks of Internet of Things. In Proceedings of the IASTED International Conference on Engineering and Applied Science, Colombo, Sri Lanka, 27–29 December 2012; pp. 27–29.
6. Durairaj, M.; Muthuramalingam, K. A new authentication scheme with elliptical curve cryptography for Internet of Things (IoT) environments. *Int. J. Eng. Technol.* 2018, 7, 119. [CrossRef]
7. Hong, N. A security framework for the Internet of Things based on public key infrastructure. *Adv. Mater. Res.* 2013, 671–674, 3223–3226. [CrossRef]
8. Hao, P.; Wang, X.; Shen, W. A collaborative PHY-aided technique for end-to-end IoT device authentication. *IEEE Access* 2018, 6, 42279–42293. [CrossRef]
9. Mahmood, K.; Chaudhry, S.A.; Naqvi, H.; Shon, T.; Ahmad, H.F. A lightweight message authentication scheme for smart grid communications in power sector. *Comput. Electr. Eng.* 2016, 52, 114–124. [CrossRef]
10. Chall, S.; Das, A.K.; Odelu, V.; Kumar, N.; Kumari, S.; Khan, M.K.; Vasilakos, A.V. An efficient ECC-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. *Comput. Electr. Eng.* 2018, 69, 534–554. [CrossRef]
11. Chung, Y.; Choi, S.; Lee, Y.; Park, N.; Won, D. An enhanced lightweight anonymous authentication scheme for a scalable localization roaming service in wireless sensor networks. *Sensors* 2016, 16, 1653. [CrossRef]
12. Turkanović, M.; Brumen, B.; Hölbl, M. A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion. *Ad Hoc Netw.* 2014, 20, 96–112. [CrossRef]

13. Jan, M.A.; Khan, F.; Alam, M.; Usman, M. A payload-based mutual authentication scheme for Internet of Things. *Future Gener. Comput. Syst.* 2019, 92, 1028–1039. [CrossRef]

14. Sun, X.; Men, S.; Zhao, C.; Zhou, Z. A security authentication scheme in machine-to-machine home network service. *Secur. Commun. Netw.* 2015, 8, 2678–2686. [CrossRef]

15. Lyu, C.; Gu, D.; Zeng, Y.; Mohapatra, P. PBA: Prediction-based authentication for vehicle-to-vehicle communications. *IEEE Trans. Dependable Secur. Comput.* 2016, 13, 71–83. [CrossRef]

16. Gope, P.; Lee, J.; Quek, T.Q.S. Lightweight and practical anonymous authentication protocol for RFID systems using physically uncloneable functions. *IEEE Trans. Inf. Forensics Secur.* 2018, 13, 2831–2843. [CrossRef]

17. Xu, H.; Ding, J.; Li, P.; Zhu, F.; Wang, R. A lightweight RFID mutual authentication protocol based on physical uncloneable function. *Sensors* 2018, 18, 760. [CrossRef]

18. Wazid, M.; Das, A.K.; Vasilakos, A.V. Design and Analysis of Secure Lightweight Remote User Authentication and Key Agreement Scheme in Internet of Drones Deployment. *IEEE Internet Things J.* 2019, 6, 3572–3584. [CrossRef]

19. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V. Design of secure key management and user authentication scheme for fog computing services. *Future Gener. Comput. Syst.* 2019, 91, 475–492. [CrossRef]

20. Jangirala, S.; Das, A.K.; Vasilakos, A.V. Designing secure lightweight blockchain-enabled RFID-based authentication protocol for supply chains in 5G mobile edge computing environment. *IEEE Trans. Ind. Inform.* 2020, 16, 7081–7093. [CrossRef]

21. Wazid, M.; Das, A.K.; Kumar, N.; Vasilakos, A.V.; Rodrigues, J.J.P.C. Design and Analysis of Secure Lightweight Remote User Authentication and Key Agreement Scheme in Internet of Drones Deployment. *IEEE Internet Things J.* 2019, 6, 3572–3584. [CrossRef]

22. Lamport, L. Password authentication with insecure communication. *Commun. ACM* 1981, 24, 770–772. [CrossRef]

23. Katz, J.; MacKenzie, P.; Taban, G.; Gligor, V. Two-server password-only authenticated key exchange. *J. Comput. Syst. Sci.* 2012, 78, 651–669. [CrossRef]

24. Xiang, T.; Wong, K.; Liao, X. Cryptanalysis of a password authentication scheme over insecure networks. *J. Comput. Syst. Sci.* 2008, 74, 657–661. [CrossRef]

25. Sun, H.-M.; Yeh, H.-T. Password-based authentication and key distribution protocols with perfect forward secrecy. *J. Comput. Syst. Sci.* 2006, 72, 1002–1011. [CrossRef]

26. Chien, H.-Y.; Jan, J.-K.; Tseng, Y.-M. An efficient and practical solution to remote authentication: Smart card. *Comput. Secur.* 2002, 21, 372–375. [CrossRef]

27. Xu, J.; Zhu, W.-T.; Feng, D.-G. An improved smart card based password authentication scheme with provable security. *Comput. Stand. Interfaces* 2009, 31, 723–728. [CrossRef]

28. Kumar, M.; Gupta, K.; Kumari, S. An improved efficient remote password authentication scheme with smart card over insecure networks. *Int. J. Netw. Secur.* 2011, 13, 167–177.

29. Challa, S.; Das, A.K.; Gope, P.; Kumar, N.; Wu, F.; Vasilakos, A.V. Design and analysis of authenticated key agreement scheme in cloud-assisted cyber–physical systems. *Future Gener. Comput. Syst.* 2020, 108, 1267–1286. [CrossRef]

30. Lin, C.; He, D.; Huang, X.; Choo, K.-K.R.; Vasilakos, A.V.; BSeln: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. *J. Netw. Comput. Appl.* 2018, 116, 42–52. [CrossRef]

31. Liao, Y.-P.; Wang, S.-S. A secure dynamic ID based remote user authentication scheme for multi-server environment. *Comput. Stand. Interfaces* 2009, 31, 24–29. [CrossRef]

32. Hsiang, H.-C.; Shih, W.-K. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. *Comput. Stand. Interfaces* 2009, 31, 1118–1123. [CrossRef]

33. Sood, S.K.; Sarje, A.K.; Singh, K. A secure dynamic identity based authentication protocol for multi-server architecture. *J. Netw. Comput. Appl.* 2011, 34, 609–618. [CrossRef]

34. Lee, C.-C.; Lin, T.-H.; Chang, R.-X. A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. *Expert Syst. Appl.* 2011, 38, 13863–13870. [CrossRef]

35. Xue, K.; Hong, P.; Ma, C. 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]
36. Amin, R.; Kumar, N.; Biswas, G.P.; Iqbal, R.; Chang, V. A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment. *Future Gener. Comput. Syst.* 2018, 78, 1005–1019. [CrossRef]

37. He, D.; Zeadally, S.; Xu, B.; Huang, X. An Efficient Identity-Based Conditional Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks. *IEEE Trans. Inf. Forensics Secur.* 2015, 10, 2681–2691. [CrossRef]

38. Zhang, J.; Cui, J.; Zhong, H.; Chen, Z.; Liu, L. PA-CRT: Chinese Remainder Theorem Based Conditional Privacy-preserving Authentication Scheme in Vehicular Ad-hoc Networks. *IEEE Trans. Dependable Secur. Comput.* 2019, 1. [CrossRef]

39. Cui, J.; Zhang, J.; Zhong, H.; Xu, Y. SPACF: A Secure Privacy-Preserving Authentication Scheme for VANET with Cuckoo Filter. *IEEE Trans. Veh. Technol.* 2017, 66, 10283–10295. [CrossRef]

40. Azees, M.; Vijayakumar, P.; Deboarth, K.J. EAAP: Efficient Anonymous Authentication with Conditional Privacy-Preserving Scheme for Vehicular Ad Hoc Networks. *IEEE Trans. Intell. Transp. Syst.* 2017, 18, 2467–2476. [CrossRef]

41. Kong, W.; Shen, J.; Vijayakumar, P.; Cho, Y.; Chang, V. A practical group blind signature scheme for privacy protection in smart grid. *J. Parallel Distrib. Comput.* 2020, 136, 29–39. [CrossRef]

42. 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]

43. Li, C.-T.; Weng, C.-Y.; Lee, C.-C.; Wang, C.-C. Secure user authentication and user anonymity scheme based on quadratic residues for the integrated EPRIS. *Procedia Comput. Sci.* 2015, 52, 21–28. [CrossRef]

44. Yeh, K.-H.; Lo, N.-W.; Kuo, R.-Z.; Su, C.; Chen, H.-Y. Formal analysis on RFID authentication protocols against de-synchronization attack. *J. Internet Technol.* 2017, 18, 765–773.

45. Maitra, T.; Islam, S.H.; Amin, R.; Giri, D.; Khan, M.; Kumar, K.N. An enhanced multi-server authentication protocol using password and smart-card: Cryptanalysis and design. *Secur. Commun. Netw.* 2016, 9, 4615–4638. [CrossRef]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).