Enhanced Authentication Protocol for the Internet of Things Environment

Chien-Ming Chen,1 Xuanang Li,1 Shuangshuang Liu,1 Mu-En Wu,2 and Saru Kumari 3

1College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China
2Department of Information and Finance Management, National Taipei University of Technology, Taipei, Taiwan
3Department of Mathematics, Chaudhary Charan Singh University, Meerut, India

Correspondence should be addressed to Saru Kumari; saryusirohi@gmail.com

Received 11 November 2021; Revised 11 January 2022; Accepted 31 January 2022; Published 10 March 2022

1.Introduction

The Internet of Things (IoT) [1–3] has become a popular topic since its conception at the end of the 20th century. The technology has developed from the simple application of a single sensor to a specific scene to the vast IoT that is currently a ubiquitous part of our lives [4]. The IoT has found application in many scenarios (shown in Figure 1), such as education [5], smart homes [6, 7], healthcare [8, 9], and VANETs [10, 11]. In schools, teachers can use IoT devices to assist them with teaching to more actively engage students in the process of learning, and the IoT makes it easier for schools to troubleshoot students’ problems more effectively. Ultimately, students’ development would improve in all respects. The traffic police could use IoT devices such as intelligent cameras to detect vehicle movements, violations, vehicle-related crime, and security, which would enable them to manage the traffic more efficiently with safety in mind. In terms of medical treatment, doctors could use intelligent detection equipment to monitor patients’ conditions in real time such that patients feel more at ease during treatment. Therefore, IoT has become ubiquitous in our daily lives, with people becoming increasingly dependent on IoT devices.

The IoT has made our lives more convenient. However, the IoT not only brings us convenience but also has significant hidden threats. For example, Amazon’s Ring home surveillance camera has a security loophole, which has been hacked. A large number of videos and photos of users have been posted online by hackers. Another example is malware named Silex, which is able to attack thousands of IoT devices, paralyzing these devices in a large area and causing considerable human, material, and financial losses. Therefore, the security of IoT must be improved to prevent further danger to human life. Researchers have developed various solutions to solve the aforementioned security problems. The primary solution involves encrypting all messages/data transmitted through public channels. This would ensure that...
the data pertaining to each entity are not leaked during the communication phase. Therefore, a secure and efficient authentication and key agreement (AKA) protocol is required.

In 2021, Alzahrani and Mahmood [12] demonstrated that Chikouche et al.’s AKA protocol [13] is insecure against device anonymity attacks and known temporary information exposure attacks and then designed a provable privacy-preserving AKA protocol for IoT. Their protocol utilized a physical unclonable function (PUF) to encrypt transmitted data. The authors claimed that the use of PUF can ensure resistance to physical capture attacks on IoT devices. Besides, the authors claimed that their protocol can resist various types of attacks and provide perfect forward security.

However, in this study, we further demonstrate that Alzahrani and Mahmood’s AKA protocol [12] is still vulnerable to physical IoT device capture attacks, privileged insider attacks, known temporary information disclosure attacks, and stolen-verifier attacks. To address the security loopholes in Alzahrani and Mahmood’s protocol [12], we further propose a new AKA protocol. In our design, we used an asymmetric encryption system to encrypt the identity of the device, considering that symmetric encryption is more efficient than asymmetric encryption. In addition, we added a login phase for IoT devices. It verifies the legality of an IoT device before it communicates with servers. The proposed protocol is then validated by a formal proof with the real-or-random (ROR) model and Burrows–Abadi–Needham (BAN) logic. We also analyze that the proposed protocol provides robustness against various kinds of potential attacks. Finally, we present a performance analysis and compare it with other related protocols. Experimental results show that the proposed protocol has low computational and communication overhead.

The remainder of this paper is organized as follows. Section 2 reviews related work. In Section 3, we briefly review the protocol of Alzahrani et al. [12]. Section 4 presents our demonstration that Alzahrani et al.’s protocol [12] is vulnerable to several attacks. The proposed protocol is described in Section 5. Sections 6 and 7 provide security and performance analyses and suitable comparisons. Finally, Section 8 concludes the paper.

2. Related Work

In 2015, Sun et al. [14] proposed an AKA protocol using hash functions. However, this protocol [14] did not provide secure identity verification. In 2018, Gope and Sikdar [15] proposed a lightweight privacy-preserving two-party AKA protocol, but this protocol is not vulnerable to desynchronization attacks and does not provide perfect forward security. Various AKA protocols that were designed to improve the level of security [16–18] are based on elliptic curve cryptography (ECC). Kalra and Sood [19] proposed an ECC-based AKA protocol for the IoT. They claimed that this protocol provided perfect forward security. However, Chang et al. found that it was unable to provide basic authentication and session key agreement mechanisms. Consequently, in an attempt to address the security loopholes in the protocol of Kalra et al., Chang et al. improved the protocol such that it offered a higher level of security. In addition, Kumari et al. [20] found that Kalra et al.’s protocol is not resistant to device anonymity attacks, interest password-guessing attacks, and privileged insider attacks. Recently, Chikouche et al. [13] proposed another AKA protocol for IoT.
3. Review of Alzahrani et al.’s Protocol

In this section, we review the protocol proposed by Alzahrani and Mahmood [12] for the IoT environment. The protocol consists of two phases: a registration phase and a mutual authentication phase. The symbols used in the paper are listed in Table 1.

3.1. Registration Phase. Assuming that an IoT device $D_i$ with the identity $ID_i$ desires to register on a server, the following steps are performed:

1. First, $D_i$ selects an identity $ID_i$ for itself; subsequently, it sends a registration request $REG_{req}$ to $S$, and outputs its identity $ID_i$ to $S$ through a secure channel. Thus, $S$ can identify whether the $D_i$ communicating with it during the authentication phase has been registered earlier.

2. After $S$ receives the registration request and the identity of $D_i$, it first generates a valid period of time $ETIME$ and then calculates

$$CK = h(ID_i \parallel K_s \parallel ETime),$$

$$O_i = h(ID_i \parallel K_s).$$

Thereafter, $S$ generates the PUF to request the information $C_i$ and $TIM_i$, and generates a series of $S$ pseudo-identities $SID_i (i = 1, 2, 3, \ldots)$. Subsequently, $S$ sends the calculated $CK$, $Q_i$, $C_i$, and $TIM_i$ to $D_i$.

3. After $D_i$ receives the message $\{CK, Q_i, C_i, TIM_i\}$, it first uses the PUF to calculate the $C_i$ transmitted from $S$ to obtain $R_i$, and then stores its own identity $ID_i$ and pseudo-identities $TIM_i$, $O_i$, and $CK$ in its own memory, while transmitting $R_i$ to $S$.

4. Finally, $S$ stores $D_i$’s identities $ID_i$, $C_i$, $R_i$, and $RTIME$ in its own memory. Consequently, the entire registration phase of $D_i$ is completed.

3.2. Mutual Authentication Phase

1. $D_i$ must perform normal authentication and communication with $S$. First, $S$ generates a random number $N_i$ and then calculates

$$c = ID_i \oplus N_i,$$

$$P_i = h(O_i \oplus N_i \oplus CK).$$

Thereafter, $D_i$ transmits the calculated $c$, $P_i$, and the pseudo-identity of $D_i$ to $S$ through the public channel.

2. After $S$ receives the transmission information from $D_i$, it first extracts the information $\{ID_i, C_i, R_i, ETime\}$ from its own memory and subsequently calculates the random number

$$N_i = c \oplus ID_i,$$

$$O_i = h(ID_i \parallel K_s).$$

Then, it calculates

$$CK = h(ID_i \parallel K_s \parallel ETime).$$

Subsequently, it is verified whether the received value of $P_i$ is equal to the value of $h(O_i \oplus N_i \oplus CK)$. If equal, the authentication is successful, and subsequently, $S$ generates its own random number $N_i$ and the pseudo-identity $TIM_i$ of the new $D_i$. Thereafter, $S$ calculates

$$V = N_i \oplus (N_i \parallel TIM_i \parallel C_i),$$

$$P_2 = h(O_i \parallel CK \parallel N_i \parallel C_i).$$

And finally, $S$ generates the session key with $D_i$.

$$SK = h(ID_i \parallel N_i \parallel N_i \parallel C_i \parallel R_i).$$

Then, the data stored in $S$ is updated, and the pseudo-identity of $D_i$ is $TIM_i$. Finally, $S$ transmits $\{P_2, V\}$ to $D_i$ through the public channel.

3. $D_i$ first calculates the values of $N_i$, $C_i$, and $TIM_i$ according to the transmitted $V$ and the random number $N_i$ generated by itself. Then, $D_i$ calculates

$$P_2' = h(O_i \parallel CK \parallel N_i \parallel N_i).$$

Thereafter, it is verified whether $P_2'$ is equal to the transmitted $P_2$ value; if equal, it implies that the $S$ that is, sending the information is legal. Subsequently, $D_i$ uses the physical IoT unclonable function to calculate the value of $R_i$, and $D_i$ finally obtains the session key to communicate with $S$ as follows:

$$SK = h(ID_i \parallel N_i \parallel N_i \parallel C_i \parallel R_i).$$

4. Cryptanalysis of Alzahrani et al.’s Protocol

In this section, we first describe the attack model used in this study and then explain the vulnerability thereof against physical IoT device capture attacks, stolen verification attacks, privileged internal attacks, and known temporary information exposure attacks.
4.1. Threat Model. Dolev–Yao (DY) model was proposed by Dolev and Yao in 1983 [21]. DY model is based on the hierarchical idea of security protocol. It first considers whether there are defects in the behavior logic of the security protocol itself, and then considers whether there are problems in the implementation method. The reason we select DY model is to offer the maximum capabilities to an adversary who is allowed to breach the security of the proposed protocol.

In the DY model, the capabilities of an adversary $\mathcal{A}$ are as follows:

1. $\mathcal{A}$ can intercept, tamper with, and delete the information transmitted from $D$ to $S$ through the public channel.
2. If $D$ is captured by $\mathcal{A}$, $\mathcal{A}$ can obtain the information stored in the memory of $D$ and use the information to perform illegal operations.
3. $\mathcal{A}$ can use power analysis to obtain the information stored in the smart card.
4. $\mathcal{A}$ can extract the registration information of $D$ stored in $S$ as a privileged insider, that is, the information sent by $D$ to $S$ during the registration phase can be obtained by $\mathcal{A}$.
5. During the authentication phase of $D$ and $S$, $D$ and $S$ generate temporary information to encrypt certain parameters, and $\mathcal{A}$ can obtain the random number generated by $D$ or $S$.

4.2. Physical IoT Device Capture Attack. In our daily life and work, IoT devices are usually everywhere. People can easily obtain IoT devices. It means that an adversary $\mathcal{A}$ can also easily capture IoT devices. After this, $\mathcal{A}$ can capture the data in IoT devices through data analysis to attack [22].

In Alzahrani et al.’s protocol, during the registration and mutual authentication phases of $D_i$ and $S$, $D_i$ can be physically captured by a certain $\mathcal{A}$. Subsequently, $\mathcal{A}$ is easily stored in the information in $D_i$. Thereafter, it performs certain calculations considering the information obtained from the memory of $D_i$, and finally obtains the session key for the communication between $D_i$ and $S$. The specific process of this attack are as follows:

1. After capturing $D_i$, $\mathcal{A}$ obtains the information $\{ID_i,TID_i,\alpha,\beta,\gamma,\phi\}$ stored in the memory and subsequently obtains the information $c$ sent by $D_i$ to $S$ through the public channel in the authentication phase. It then calculates
   \[ N_i = ID_i \oplus c. \] (9)
2. $\mathcal{A}$ obtains the value of $V$ through the public channel. Thereafter,
   \[ (N_i \| TID_i^{\text{new}} \| C_i) = N_i \oplus V. \] (10)
   Is calculated to obtain the random numbers $N_i$ and $C_i$ generated by $S$.

3. With the value of $C_i$ obtained in the previous step, $\mathcal{A}$ uses the PUF to obtain the value of $R_i$.
4. Finally, $\mathcal{A}$ obtains the session key
   \[ SK = h(ID_i \| N_i \| N_s \| C_i \| R_i). \] (11)
   Used for communication between $S$ and $D_i$ according to the parameters $\{N_i, N_s, C_i, R_i, ID_i\}$. Therefore, $\mathcal{A}$ can successfully perform a physical IoT capture attack.

4.3. Privileged Insider Attack. The whole process of the protocol includes various participants, and the staff are also part of the participants. However, it cannot ensure the credibility of the staff. In case the $\mathcal{A}$ sneaks into the staff, the secret data contacted by the staff will not be saved [23]. In Alzahrani et al.’s protocol, during the registration phase, $D_i$ sends a registration request and its own identity information to $S$ through a secure channel. In general, the registration information of $D_i$ is confidential and is not made available to others. However, should a certain $\mathcal{A}$ exist among the $S$ administrators, the registration information would be readily accessible. Consequently, $\mathcal{A}$ could perform certain calculations based on the registration information and the information obtained through the public channel to finally obtain the session key between $D_i$ and $S$. The specific process is as follows:

1. As a privileged insider of $S$, $\mathcal{A}$ obtains the registration information $ID_i$ of $D_i$.
2. Thereafter, $D_i$ obtains the information $C_i$ and $V$ transmitted by $S$ and $D_i$, respectively, through the common channel during the authentication phase. Then, the parameters $C_i$ and $ID_i$, which were obtained in this manner, are used to calculate $N_i$. Thereafter,
   \[ (N_i \| ID_i^{\text{new}} \| C_i) = N_i \oplus V. \] (12)
   Is calculated using the calculated $N_i$ and $V$ values to obtain the parameters $N_i$ and $C_i$.
3. A PUF is used in the protocol and the method whereby this function is calculated is fixed. Provided that $\mathcal{A}$ obtains the value of $C_i$, the value of $R_i$ can be obtained with the aforementioned method.
4. Consequently, according to the aforementioned parameters, $\mathcal{A}$ can easily obtain the session key between $S$ and $D_i$
   \[ SK = h(ID_i \| N_i \| N_s \| C_i \| R_i). \] (13)
   Equipped with this information, $\mathcal{A}$ can access the private content of the communication between $S$ and $D_i$ based on the session key between the two parties. Therefore, we can conclude that the protocol of Alzahrani and Mahmood [12] does not offer resistance against privileged insider attacks.
4.4. Known Temporary Information Disclosure Attack. Temporary data will be generated during the operation of various devices, and these temporary data will be temporarily stored in the memory. $A$ can easily obtain the temporarily stored data and further attack [24].

In general, when $S$ and $D_i$ perform mutual authentication, $D_i$ transmits a certain amount of its own private information to $S$. Then, $D_i$ is verified on the basis of this private information, which is stored in $S$ and which cannot be accessed by other people. Therefore, $D_i$ generates random numbers to encrypt the private information and then transmit the encrypted parameters to $S$. However, under certain circumstances, this temporary information may also be obtained by $A$, who could consequently use it to crack the session key based on the obtained temporary random number information. Here, we demonstrate that the protocol of Alzahrani et al. is not robust against temporary information disclosure attacks at all times. The attack could take place via the following steps.

1. $A$ obtains the random number $N_i$ generated by $D_i$ during the authentication phase with $S$ through certain illegal channels and then obtains the parameter $V$ transmitted by $S$ to $D_i$ through the public channel.

2. $A$ can calculate
   \[ (N_i)\|TID_i^{\text{new}}\|C_i) = N_i \oplus V. \]  
   Considering the obtained parameter information $N_i$ and $V$, $A$ can easily obtain the random number $N_i$ generated by $S$ and the parameter $C_i$ used by the PUF. Thus, $R_i$ can be calculated according to parameter $C_i$.

3. Regarding the session key $SK$ for the communication between $D_i$ and $S_i$, according to certain parameters obtained by $A$ earlier, information about the keys of both parties can be easily obtained. Therefore, the protocol of Alzahrani et al. cannot resist known temporary information disclosure attacks.

4.5. Stolen-Verifier Attack. The verifier stored in the server will also be leaked. When the server is unattended, the $A$ can analyze and obtain the verifier stored in it, so as to further attack [25]. During the registration and authentication phases, $S$ stores certain information in its own memory for subsequent calculations. A stolen verification attack implies that $A$ accesses the information in the memory and then performs calculations to finally obtain the session key between the two parties. The specific attack process is as follows:

1. In the registration phase, $S$ stores the identity of $D_i$, that is, $ID_i$ and $C_i$ used by the PUF, the result $K_i$, and the validity period $ETIME$ in its own memory. However, $A$ can gain access to the parameters in $S$'s memory through certain means.

2. Subsequently, $A$ uses the acquired identity information $ID_i$ of $D_i$ to calculate the parameter $N_i$, and the parameter $V$ is used to derive the parameters $N_s$ and $C_i$.

3. $S$ and $D_i$ communicate using the session key $SK$. As $A$ has obtained the parameters required to calculate the session key, it can thus access the session key of both parties. Therefore, we can conclude that the protocol of Alzahrani et al. does not provide resistance against stolen verification attacks.

5. Proposed Protocol

This section proposes a new AKA protocol for the IoT environment. The protocol contains three phases, the pre-deployment phase, the IoT device registration phase, and the login and authentication phase. The proposed protocol considers two roles, IoT devices, and a server. In the pre-deployment phase, an IoT device and the server negotiate a shared key for later use. The IoT device registration phase enables IoT devices to register to the server. Devices and the server further authenticate each other and generate a session key.

5.1. Predeployment Phase. Before $D_i$ and $S$ are authenticated, a shared key $K$ is first assigned to $D_i$ and $S$, such that $D_i$ can encrypt its own identity in the later registration phase. The shared key between the two is only known to $D_i$ and $S$ and is inaccessible to other devices and personnel.

5.2. IoT Device Registration Phase. Figure 2 illustrates the IoT device registration phase. The detailed steps are as follows:

1. First, $D_i$ selects an identity $ID_i$ and password $PW_i$ and then uses a symmetric encryption algorithm to encrypt the identity $ID_i$ of $D$ to obtain the pseudo-identity of the IoT device.

\[ RID_i = \text{Enc}_k(ID_i). \]  

Thereafter, the IoT device transmits the registration request $R_i$ and pseudo-identity $RID_i$ to $S$ through a secure channel.

2. After $S$ receives the registration information, it first decrypts a validity period $ETIME$ and subsequently decrypts the pseudo-identity to obtain the device identity.

\[ ID_i = \text{De}_k(RID_i). \]  

Furthermore, $S$ encrypts the identity of $D_i$ and the private key of $S$ to obtain

\[ A = h(ID_i \| K_i \| ETIME), \] 
\[ B = h(ID_i \| K_i). \]

Consequently, $S$ generates $C_i$ for $D_i$ and also generates a series of pseudo-identities $SID_i(i = 1, 2, 3 \ldots)$ for its use. Finally, $S$ sends the
5.3. Login and Authentication Phase. Figure 3 shows the login phase of $D_i$ and the authentication phase with $S$. The specific details of the process are as follows.

(1) $D_i$ enters its own $ID_i$ and password $PW_i$ and generates a temporary random number $N_i$. Then, $D_i$ uses a symmetric encryption algorithm to encrypt its own identity to obtain a pseudo-identity as follows:

$$ RID_i = Enc_k(ID_i) $$

$$ V_i = h(ID_i||B||PW_i). $$

The verification $V_i$ is passed, and $D_i$ stores it in its own memory to check whether it is equal to $V$, which already resides in the memory, to prove the legality of $D_i$. If the values are equal, the login is successful and then the following is calculated.

$$ c = N_i \oplus RID_i, $$
$$ U = h(B \oplus N_i \oplus A \oplus T_1). $$

Finally, $D_i$ sends $\{c, SID_i, U, T_1\}$ to $S$ through the public channel.

(2) After receiving the information, $S$ first verifies the freshness of the timestamp $T_1$, and if $T_2 - T_1 \leq T'$, then the authentication phase continues. The $RID_i$ is determined through $SID_i$, and $S$ uses a symmetric decryption algorithm to decrypt the $RID_i$ stored in the memory to obtain the real identity of $D_i$ as follows:

$$ ID_i = De_s_k(RID_i). $$

Following which certain additional parameters are calculated as follows:

$$ N_i = RID_i \oplus c, $$
$$ B = h(ID_i||K_s), \quad A = h(ID_i||K_s||ETime), $$

$$ U' = h(B \oplus N_i \oplus A \oplus T_1). $$

Subsequently, it is verified whether $U'$ is equal to the transmitted $U$; if equal, it implies that a legitimate $D_i$ is communicating with it. Then, $S$ generates a random number $N_s$ and calculates the following:

$$ G = N_i \oplus (N_s || C_i), $$
$$ W = h(B \parallel A \parallel N_s \parallel N_j). $$

Finally, the session key to communicate with $D_i$ is generated as

$$ SK = h(ID_i || N_s || N_j || C_i || R_i), $$

whereupon $S$ sends $\{G, W, T_2\}$ to $D_i$ through the public channel.

(3) $D_i$ first checks the freshness of the timestamp $T_2$ generated by $S$ and then obtains $N_s$ and $C_i$ through the parameters $G$ and $N_s$ as

$$ W' = h(B \parallel A \parallel N_s \parallel N_j). $$

$D_i$ compares the calculated $W'$ with the received $W$ and proves that if they are equal, $S$ is legitimate. Finally, $D_i$ calculates the session key

$$ SK = h(ID_i || N_s || N_j || C_i || R_i). $$

For the communication between $S$ and $D_i$ according to $R_i$, calculated by the PUF.

The aforementioned steps represent the entire process according to which $D_i$ registers with $S$ and performs the key exchange.

6. Security Analysis

In this section, we present the analyses we conducted to prove that the proposed protocol is sufficiently secure.

6.1. BAN Logic Analysis. Burrows—Abadi—Needham logic has been used in several studies to prove whether a protocol can be executed securely. This section uses BAN logic to prove the security and reliability of our proposed protocol. This proof verifies that our protocol can successfully establish and share a session key between the server and IoT device. $D$ represents an IoT device in the following proof, and $S$ represents the server. The specific proof rules and process are as follows:

6.1.1. BAN Logic Rules

Message-meaning rule (R1) $(D \parallel E \parallel S, D \parallel S | M) \parallel K / D \parallel S \parallel M)$$

Nonce-verification rule (R2) $(D \parallel E \parallel S, D \parallel S | M) \parallel K / D \parallel S \parallel M)$$

Jurisdiction rule (R3) $(D \parallel E \parallel S, D \parallel S | M) \parallel K / D \parallel S \parallel M)$$

Freshness rule (R4) $(D \parallel E \parallel S, D \parallel S | M) \parallel K / D \parallel S \parallel M)$$

Belief rule (R5) $(D \parallel E \parallel S, D \parallel S | M) \parallel K / D \parallel S \parallel M)$$

Session key rule (R6) $(D \parallel E \parallel S, D \parallel S | M) \parallel K / D \parallel S \parallel M)$$
### 6.1.2. Goals

| G1  | $D \vdash D \leftrightarrow S$ |
|-----|--------------------------------|
| G2  | $S \vdash D \leftrightarrow S$ |
| G3  | $D \vdash S \vdash D \leftrightarrow S$ |
| G4  | $S \vdash D \vdash D \leftrightarrow S$ |

### 6.1.3. Idealizing Communication

| M1  | $D \rightarrow S: \{c, SID, u, T\}$ |

### 6.1.4. Initial State Assumptions

| A1  | $D \vdash D \leftrightarrow S$ |
|-----|--------------------------------|
| A2  | $S \vdash D \leftrightarrow S$ |
| A3  | $D \vdash \Upsilon(N, C_i)$ |
| A4  | $S \vdash \Upsilon(ID_i, N_i)$ |
| A5  | $S \vdash (N_i, C_i, D_i)$ |
| A6  | $S \vdash D \Rightarrow (ID_i, N_i)$ |
6.1.5. Detailed Steps. By considering the message M1 and using the seeing rule, we get

\[ S1: S\{ \langle ID_r, N_r \rangle, \langle N_r, C_r \rangle, U \} \]

Using S1, we get the following:

\[ S2: S\{ \langle ID_r, N_r \rangle \} \]

Under the assumption of A2, using S2, R1 can be used to obtain:

\[ S3: S|D| \sim (ID_r) \]

With conclusion R2, using A4 and S3, the following can be obtained:

\[ S4: S|D| \equiv (ID_r) \]

Using A6, R3, and conclusion S4, the following can be obtained:

\[ S5: S|D| \equiv (N_r) \]

According to conclusion S1, the following can be obtained:

\[ S6: S\{ \langle N_r \rangle \} \]

Using A2, R1, and conclusion S6, the following can be obtained:

\[ S7: S|D| \sim (N_r) \]

Using A4, R2, and conclusion S7, the following can be obtained:

\[ S8: S|D| \equiv (N_r) \]

Using A6, R2, and conclusion S3, the following can be obtained:

\[ S9: S|D| \equiv (N_r) \]

Because SK = h(ID_r || N_r || N_r || C_r || R_r), using A5, S5 and S9, we obtain:

\[ S10: S|D| \equiv D^{SK} \] (G2)

Using A4 and R4, we can obtain:

\[ S11: S|D| \equiv D^{SK} \] (G4)

In addition, considering the message M2, we obtain:

\[ S12: D\{ \langle N_r, C_r \rangle, \langle X_1, X_2 \rangle \} \]

Using S12, we get the following:

\[ S13: D\{ \langle N_r, C_r \rangle \} \]

By using A1, S13, and R1, we obtain:

\[ S14: D|S| \sim (N_r, C_r) \]

With conclusion S14, using A3 and applying R2, we obtain:

\[ S15: D|S| \equiv (N_r, C_r) \]

Applying A8, S15 and R3 we obtain:

\[ S16: D|S| \equiv (N_r, C_r) \]

Because SK = h(ID_r || N_r || N_r || C_r || R_r), using A7 and S16, we obtain:

\[ S17: D|S| \equiv D^{SK} \] (G1)

With conclusion S17, using A3 and R4, we can obtain:

\[ S18: U|S| \equiv U^{SK} \] (G3)

6.2. ROR Security Analysis. The real-or-random (ROR) [26] model is a function that randomly maps all possible inputs and outputs. ROR model is a popular security proof method, which can be used for evaluating the security of protocols. [27]. To prove that our proposed protocol offers the necessary security, we use the ROR model [26] to analyze the protocol.

6.2.1. ROR Model. The protocol involves two entities, D_1 and S. The symbols \( H_D^x \) and \( H_S^y \) are used to represent \( x_{th} D_1 \) and \( y_{th} S \), respectively, while Z is used to encapsulate the two entities, \( Z = \{H_D^x, H_S^y\} \). Furthermore, in the analysis process, \( \mathcal{A} \) is required to submit multiple queries, which are as follows:

- Execute (Z): by performing this operation, \( \mathcal{A} \) can obtain the information transmitted by \( D_1 \) and \( S \) through the public channel during the login authentication phase.
- Send (Z, M): with the help of this query condition, \( \mathcal{A} \) can initiate requests to \( D_1 \) and \( S \), which implies that \( \mathcal{A} \) can send certain false messages to both \( D_1 \) and \( S \).
- Corrupt (Z): this operation can help \( \mathcal{A} \) obtain the message \( \{ID_r, A, B, V\} \in D_1 \)’s memory.
- Reveal (Z): \( \mathcal{A} \) can obtain the shared key \( K \) of \( D_1 \) and \( S \) in the mutual authentication phase.
- Test (Z): \( \mathcal{A} \) verifies the correctness of the session key \( SK \) established by the two entities in the protocol by tossing a coin \( C \).

Theorem: assuming that \( \mathcal{A} \) desires to obtain the session key \( SK = h(ID_r || N_r || N_r || C_r || R_r) \) in the authentication phase of \( D_1 \) and \( S \), the advantage \( \mathcal{A} \) has to successfully obtain the session key within the polynomial time \( t \),

\[ A \cdot dv_{\mathcal{A}}^{KF} \leq (2 \max \{E, q_s, q_h, q_k, q_v\} \cdot \frac{1}{2^{|H|}} + 2A \cdot dv_{\mathcal{A}}^{\mathcal{A}}) \]

Here, \( b \) represents the password length of \( D_1 \) during the login phase, \( E \) and \( x' \) represent two constants, \( q_h \) represents the number of hash functions in the protocol, \( |H| \) represents the range of Hash functions, and \( q_k \) represents the number of Send functions.

6.2.2. Security Proof

Proof: in the proof process, we defined four games \( GM_i \) (i = 0,1,2,3) to prove the security of the protocol, of which \( SUCG(GM_i) (i = 0,1,2,3) \) represents the probability of winning the game. The specific description of the process is as follows.

\( GM_0 \): in the initial game, \( \mathcal{A} \) must compete with legal \( D \) and \( S \). At the beginning of the game, \( \mathcal{A} \) does not perform any query operations; therefore, \( A \cdot dv_{\mathcal{A}}^{KF} = 2\Pr [SUCG_{GM_0} - 1] \).

\( GM_1 \): in the second game, \( \mathcal{A} \) executes the Execute query operation to obtain the messages \( \{c, U, T_1\} \) and \( \{G, W, T_2\} \) transmitted by \( S \) and \( D \) through the public
channel, and $\mathcal{A}$ needs to perform Reveal and Test operations to verify whether the session key contains long-term keys and randomness, which can be easily determined. Therefore, messages transmitted via the public channel must be monitored continuously to ensure that $\mathcal{A}$ cannot obtain this information. Thus, we can conclude $\Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_1} = \Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_2}$.

$\mathcal{GM}_1$: in this game, we simulate an attack. During the attack, $\mathcal{A}$ continues to submit $Sen_d$ queries. In addition, $\mathcal{A}$ also obtains the information exchanged between $S$ and $D_i$ during the authentication phase. However, to obtain the session key of both parties, $\mathcal{A}$ must know the identity of $D_i$ and the random number of $S$. However, it is impossible for $\mathcal{A}$ to obtain the identity of $D_i$ because $\mathcal{A}$ only has access to the identity after symmetric encryption. Moreover, because of the existence of PUFs, obtaining $Di$ and $also obtains the information exchanged between $A$ must be monitored continuously to ensure that $\mathcal{A}$ cannot obtain this information. Thus, we can conclude $\Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_1} = \Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_2}$.

$\mathcal{GM}_2$: in the last game, $\mathcal{A}$ uses Send and Corrupt to query the information $\{ID_i, A, B, V\}$ stored in the memory of $D_i$. $\mathcal{A}$ attempts to obtain the random numbers $N_i$ and $N_j$ used in the session key, but to obtain $N_i$, it must obtain the key $K$ that encrypts the identity of $D_i$. After $\mathcal{A}$ performs these operations, we obtain the following relationship:

$$\Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_1} - \Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_2} \leq \max\left\{\frac{E' \cdot q_x^2 \cdot q_j}{2^b}\right\}.$$  

(28)

It is well known that when we toss a coin with a uniform texture, the probability of getting a heads or tails is (1/2); thus, the probability of $\mathcal{A}$ guessing the correct session key is

$$\Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_d} = \frac{1}{2}.\quad (29)$$

Based on the aforementioned drawn conclusions, we can derive the relationship:

$$\frac{1}{2} A_{dv_{PF}} = \Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_d} - \frac{1}{2} = \Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_d} - \Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_1} \leq \Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_1} - \Pr_{\mathcal{Succ}_d}^{\mathcal{GM}_1}.$$  

(30)

Subsequently, we can obtain

$$A_{dv_{PF}} \leq \frac{2 \max\left\{\frac{E' \cdot q_x^2 \cdot q_j}{2^b}\right\} + q_h}{|H| + 2A_{dv_{PF}}}.$$  

(31)

6.3. Security Analysis. In this section, we evaluate the proposed protocol to prove its ability to withstand a privileged insider attack, known temporary information disclosure attack, stolen verification attack, physical IoT device capture attack, and a perfect forward security and IoT device simulation attack.

6.3.1. Ability to Withstand Privileged Insider Attack. If $\mathcal{A}$ was to succeed in obtaining the registration information $RID_i$ that was sent by $D_i$ to $S$ during the registration phase, the value of $N_i$ could be calculated based on the pseudo-identity of $D_i$, while the parameter $c$ could be intercepted on the public channel. Furthermore, the parameters $N_i$ and $C_i$ can be derived from $N_j$ and $V$; however, the real identity of $D_i$ is only available after being encrypted by a symmetric encryption algorithm, which $\mathcal{A}$ is unable to decrypt. Therefore, even if $\mathcal{A}$ was to obtain the registration information sent by $D_i$ to $S$ during the registration phase, it would not succeed in obtaining the session key for communication between $D_i$ and $S$. Therefore, the proposed protocol offers protection against privileged insider attacks.

6.3.2. Ability to Withstand Known Temporary Information Disclosure Attack. We assume that $\mathcal{A}$ obtains the temporary information $N_i$ generated by $D_i$ during the login authentication phase. Consequently, $\mathcal{A}$ can also easily obtain the pseudo-identity of $D_i$, that is $RID_i$, based on the information on the public channel. However, the session key of $D_i$ and $S$ is based on the real identity of $D_i$ and is composed of $ID_i$, such that $\mathcal{A}$ cannot obtain the key $K$, which would be necessary to decrypt the real identity. Therefore, despite $\mathcal{A}$ having obtained $N_i$, it would not be able to access the value of the session key. In addition, we assume that $\mathcal{A}$ obtains the temporary information $N_j$ generated by $S$ in the authentication phase, which is used by $D_i$ to verify the legality of $S$ by encrypting $A$. As $\mathcal{A}$ does not know the values of the parameters $A$ and $B$, obtaining temporary information from $S$ would not allow $\mathcal{A}$ to crack the session key. In summary, our proposed protocol can effectively resist known temporary information exposure attacks.
6.3.3. Ability to Withstand Stolen Verification Attack. We assume that $\mathcal{A}$ obtains certain parameters $\{RI_{Di}, C_{Di}, R_{S}\}$ stored in the memory of $S$, which $\mathcal{A}$ can use to calculate the value of $R_{S}$. The session key of $D_{i}$ and $S$ is set by $D_{i}$. The real identity $ID_{Di}$ is composed of random numbers $N_{Di}, N_{S}, C_{Di}$, and $R_{i}$ generated by $D_{i}$ and $S$ during the authentication phase and thus cannot be obtained. Therefore, even if the information in the $S$ memory was to be obtained, it would not be possible to successfully launch the attack. Thus, our proposed protocol provides resistance against stolen verification attacks.

6.3.4. Physical IoT Device Capture Attack. Assuming that $D_{i}$ is physically captured by $\mathcal{A}$, the latter can obtain the information $\{ID_{Di}, B, A, V\}$ stored in the memory of $D_{i}$ according to the power analysis. However, $\mathcal{A}$ cannot obtain the information generated by $D_{i}$ and $S$ during the login authentication phase. Furthermore, the random numbers $N_{Di}$ and $N_{S}$, and the key-value pairs $C_{Di}$ and $R_{Di}$ used by the PUF are also not available. Therefore, $\mathcal{A}$ cannot carry out the attack despite capturing $D_{i}$ and obtaining the information in the memory. Thus, our protocol can resist physical IoT device capture attacks.

6.3.5. Perfect Forward Security. Consider that $\mathcal{A}$ obtains the shared key $K$ of $D_{i}$ and $S$ in the predeployment phase; however, the key $K$ is only used to encrypt the identity $ID_{Di}$ of $D_{i}$. Although the value of the shared key $K$ is known, the identity of $D_{i}$, that is $ID_{Di}$, cannot be obtained by $\mathcal{A}$. Therefore, $\mathcal{A}$ cannot crack the session key $SK_{Di}$ of $D_{i}$ and $S$, and thus the $K$ obtained in this manner is of no value. In summary, the proposed protocol has perfect forward security.

6.3.6. IoT Device Simulation Attack. Assume that $\mathcal{A}$ captures an IoT device and attempts to tamper with certain information in the memory, thereby establishing a session key with $S$. First, $\mathcal{A}$ must log in after obtaining $D_{i}$. However, during the login phase, the login password of $D_{i}$ must be known, which is not stored in the memory of $D_{i}$. Therefore, $\mathcal{A}$ cannot log in successfully and thus cannot simulate the operation of networked devices. Therefore, our proposed protocol can effectively resist IoT device simulation attacks.

7. Security and Performance Comparisons

In this section, we present a comparison of the proposed protocol with existing protocols [12, 13, 20, 28–30] in related fields. We compared the performance of the protocol considering its running time and communication cost. In addition, protocols were compared in terms of security. Comprehensive performance and security analyses prove that the proposed protocol has significant advantages in both respects. The specific evaluation and comparison process is as follows.

We drew on the experimental environment of a published protocol [12]. As physical IoT devices and servers are involved in the protocol, artificial devices can be used to implement the encryption operation of the IoT devices in the authentication phase. Moreover, a desktop system can be used to implement the encryption operation on the server. In the case of the specific implementation process, only the running time of the hash function is considered in the protocol authentication phase, whereas the connection operation and the XOR operation are ignored for the moment. According to the previous experimental results [12], the single hash operation time for using artificial equipment for the realization of $D_{i}$ is 1.063 ms, whereas that for realizing the encryption operation on $S$ is 0.0027 ms. Furthermore, the communication cost was determined by only considering the cost incurred during the login authentication phase. We specify the identity of $D_{i}$, XOR operation, timestamp, symmetric encryption, symmetric decryption, connection operation, and memory occupied using the hash function in the transmission process as 160, 160, 128, 128, 160, 160, and 256 bits, respectively. $T_{h}$ represents the time consumed by a single hash operation, and $T_{p}$ represents the time consumed by a single dot multiplication operation.

7.1. Security Comparisons. Security analysis: although the time and communication performance of the proposed protocol was not analyzed, we performed a security analysis of the protocol. We compared the proposed protocol with related protocols and proved that our protocol offers the required security by evaluating whether the protocols can resist certain attacks during the login authentication phase. The main attacks included in the comparison were: A1—privileged insider attack, A2—IoT device capture attack, A3—stolen verification attack, A4—IoT device simulation attack, A5—perfect forward security, and A6—desynchronization attack. The results in Table 2 confirm that our protocol can resist various attacks and has significant advantages over other protocols in terms of security. In the table, “Yes” implies that the protocol can resist the attack, whereas “No” indicates that it cannot.

| Protocols | A1 | A2 | A3 | A4 | A5 | A6 |
|-----------|----|----|----|----|----|----|
| Alzahrani and Mahmood [12] | No | No | No | Yes | Yes | Yes |
| Kumari et al. [20] | Yes | Yes | Yes | Yes | Yes | No |
| Chikouche et al. [13] | Yes | Yes | Yes | Yes | No | Yes |
| Chaudhry et al. [28] | Yes | Yes | Yes | Yes | No | Yes |
| Melki et al. [29] | Yes | Yes | Yes | No | No | Yes |
| Panda et al. [30] | Yes | Yes | Yes | Yes | No | Yes |
| Ours | Yes | Yes | Yes | Yes | Yes | Yes |

7.2. Performance Comparisons. Time consumption cost analysis: time consumption cost represents the time consumed by the encryption operations used in the authentication phase of $D_{i}$ and $S$. We compared the proposed protocol with existing protocols [12, 13, 20, 28–30] in related fields. In the login and authentication phase of the proposed protocol, $D_{i}$ uses four hash functions; thus, the time consumed by the IoT device is $4 \times 1.063 = 4.252$ ms, while $S$ uses five hash functions, resulting in a time consumption of $5 \times 0.0027 = 0.0135$ ms by $S$ on the desktop system. Thus, the total time consumed by our proposed protocol is $4.252 + 0.0135 = 4.2655$ ms. Furthermore, the time consumed by each entity of other related protocols and the total time consumed are shown in Figure 4. In addition, Table 3 shows the time consumed by each protocol more intuitively.
Table 3: Computational cost of the proposed protocol.

| Protocols                  | $D_T$                      | $S_T$                      | Total computation  |
|----------------------------|----------------------------|----------------------------|--------------------|
| Ours                       | $4 T_h \approx 4.252 ms$   | $5 T_H \approx 0.0135 ms$  | $4.2655 ms$        |
| Alzahrani and Mahmood [12] | $3 T_h \approx 3.189 ms$   | $5 T_H \approx 0.0135 ms$  | $3.2025 ms$        |
| Kumari et al. [20]         | $3 T_h + 6 T_P \approx 9.861 ms$ | $4 T_h + 6 T_P \approx 0.0354 ms$ | $9.8964 ms$        |
| Chikouche et al. [13]      | $5 T_h + 5 T_P \approx 10.874 ms$ | $4 T_h + 5 T_P \approx 0.0313 ms$ | $10.9053 ms$       |
| Chaudhry et al. [28]       | $7 T_h + 2 T_P + 2 T_b \approx 12.107 ms$ | $6 T_h + 2 T_P + 2 T_b \approx 0.0298 ms$ | $12.1368 ms$       |
| Melki et al. [29]          | $5 T_h \approx 5.314 ms$   | $5 T_H \approx 0.0135 ms$  | $5.3275 ms$        |
| Panda and Chattopadhyay [30]| $4 T_h + 5 T_P \approx 9.812 ms$ | $5 T_h + 5 T_P \approx 0.034 ms$ | $9.8460 ms$        |
Communication cost analysis: communication cost refers to the information transmitted between $D_1$ and $S$ during the login authentication phase. First, $D_1$ transmits the information $\{c, V, T_1\}$ to $S$, and $S$ sends the information $\{G, W, T_2\}$ to $D$ after authenticating it as legitimate $D$. As $c = RID_1 \oplus N_1$ and $u = h(B \oplus N_1 \oplus A \oplus T_1)$, the communication cost consumed by $D$ in the authentication phase is $160 + 256 + 160 = 576$ bits. In addition, $G = N_2 \oplus (N_1 \oplus C)$ and $W = h(B \oplus N_1 \| N_2)$, and thus the communication cost consumed by $S$ in the authentication phase is $160 + 256 = 416$ bits. Considering these results, we can conclude that the overall communication cost incurred by the proposed protocol during the login authentication phase is 992 bits. Figure 5 shows the communication costs incurred by various entities of other related protocols [12, 13, 20, 28–30]. Furthermore, Table 4 presents a more intuitive comparison of the communication costs incurred by our proposed protocol and other protocols.

Based on the aforementioned analysis of the protocol, it is evident that the proposed protocol is superior to other related protocols in terms of time and security performance.

### 8. Conclusion

The development of IoT has increasingly focused attention on security issues related to IoT communication. Alzahrani et al. proposed an identity-based authentication and key exchange protocol to address the key exchange network problem experienced by IoT devices and servers. However, we identified many security vulnerabilities in their protocol, and the IoT device login phase was absent. Therefore, we designed a two-factor encryption protocol using the symmetric key method based on an identity password. Using the ROR model, we proved that our proposed protocol could resist various attacks. In addition, we observed that our protocol is significantly advantageous in terms of time and communication cost through comparison with other related protocols. Therefore, future developments in the IoT industry are anticipated to benefit from our protocol, which is expected to provide more efficient security for smart devices. Owing to the possibility of improving the communication efficiency of this protocol, we will enhance this protocol in future work to help provide more efficient communication efficiency.

### Data Availability

No data were used to support this study.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### References

[1] H. Xiong, X. Huang, M. Yang, L. Wang, and S. Yu, “Unbounded and efficient revocable attribute-based encryption with adaptive security for cloud-assisted internet of things,” *IEEE Internet of Things Journal*, vol. 9, 2021.

[2] C.-T. Li, C.-C. Lee, C.-Y. Weng, and C.-M. Chen, “Towards secure authenticating of cache in the reader for RFID-based IoT systems,” *Peer-to-Peer Networking and Applications*, vol. 11, no. 1, pp. 198–208, 2018.

[3] H. Xiong, Y. Zhao, Y. Hou et al., “Heterogeneous signcryption with equality test for IoT environment,” *IEEE Internet of Things Journal*, vol. 8, 2020.

[4] C. M. Chen, X. Deng, W. Gan, J. Chen, and S. H. Islam, “A secure blockchain-based group key agreement protocol for IoT,” *The Journal of Supercomputing*, vol. 77, pp. 1–23, 2021.

[5] L. McRae, K. Ellis, and M. Kent, “Internet of things (IoT): education and technology,” *Relatsh. between Educ. Technol. students with Disabil. Learnne*, Res, pp. 1–37, 2018.

[6] S. A. Chaudhry, J. Nebhen, K. Yahya, and F. Al-Turjman, “A privacy enhanced authentication scheme for securing smart grid infrastructure,” *IEEE Transactions on Industrial Informatics*, 2021.

[7] H. Yang, W. Lee, and H. Lee, “IoT smart home adoption: the importance of proper level automation,” *Journal of Sensors*, vol. 2018, 2018.

[8] T. Y. Wu, L. Yang, J. N. Luo, L. Ming-Taiex, and J. Wu, “A Provably Secure Authentication and Key Agreement Protocol In cloud-based smart healthcare environments,” *Security and Communication Networks*, vol. 2021, 2021.

[9] S. A. Chaudhry, A. Irshad, J. Nebhen et al., “An anonymous device to device access control based on secure certificate for internet of medical things systems,” *Sustainable Cities and Society*, vol. 75, Article ID 103322, 2021.

[10] T.-Y. Wu, Z. Lee, L. Yang, C.-M. Chen, and R. Tso, “A provably secure authentication and key exchange protocol in vehicular ad hoc networks,” *Security and Communication Networks*, vol. 2021, pp. 1–17, 2021.

[11] P. Wang, C. M. Chen, S. Kumari, M. Shojaifar, R. Tafazolli, and Y. N. Liu, “HDMA: hybrid D2D message authentication scheme for 5G-enabled VANETs,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, 2020.

[12] B. A. Alzahrani and K. Mahmood, “Provably privacy preserving authentication solution for internet of things environment,” *IEEE Access*, vol. 9, 2021.

[13] N. Chikouche, P.-L. Cayrel, E. H. M. Mboup, and B. O. Boidje, “A privacy-preserving code-based authentication protocol for
Internet of Things,” *The Journal of Supercomputing*, vol. 75, no. 12, pp. 8231–8261, 2019.

[14] X. Sun, S. Men, C. Zhao, and Z. Zhou, “A security authentication scheme in machine-to-machine home network service,” *Security and Communication Networks*, vol. 8, no. 16, pp. 2678–2686, 2015.

[15] P. Gope and B. Sikdar, “Lightweight and privacy-preserving two-factor authentication scheme for IoT devices,” *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2018.

[16] A. A. Alamr, F. Kausar, J. Kim, and C. Seo, “A secure ECC-based RFID mutual authentication protocol for internet of things,” *The Journal of Supercomputing*, vol. 74, no. 9, pp. 4281–4294, 2018.

[17] A. Maarof, M. Senhadji, Z. Labbi, and M. Belkasmi, “Authentication protocol for securing internet of things,” in *Proceedings of the Fourth International Conference*, pp. 1–7, USA, June 2018.

[18] S. Hussain, S. A. Chaudhry, O. A. Alomari, M. H. Alsharif, M. K. Khan, and N. Kumar, “Amassing the security: an ECC-based authentication scheme for Internet of drones,” *IEEE Systems Journal*, vol. 15, 2021.

[19] S. Kalra and S. K. Sood, “Secure authentication scheme for IoT and cloud servers,” *Pervasive and Mobile Computing*, vol. 24, pp. 210–223, 2015.

[20] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, “A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers,” *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6428–6453, 2018.

[21] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[22] I. Agadakos, C. Y. Chen, M. Campanelli et al., “Jumping the air gap: modeling cyber-physical attack paths in the Internet-of-Things,” in *Proceedings of the 2017 Workshop*, pp. 37–48, USA, November 2017.

[23] S. Y. Kim, H. Kim, and D. H. Lee, “An Efficient Id-Based Mutual Authentication Secure against Privileged-Insider Attack,” in *Proceedings of the 2015 5th International Conference on IT Convergence and Security (ICITCS)*, pp. 1–4, IEEE, Kuala Lumpur, Malaysia, August 2015.

[24] Q. Jiang, S. Zeadally, J. Ma, and D. He, “Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks,” *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[25] K. A. Rahman, K. S. Balagani, and V. V. Phoha, “Making impostor pass rates meaningless: a case of snoop-forgereplay attack on continuous cyber-behavioral verification with keystrokes,” in *Proceedings of the CVPR 2011 WORKSHOPS*, pp. 31–38, IEEE, Colorado Springs, CO, USA, August 2011.

[26] M. Abdalla, P.-A. Fouque, and D. Pointcheval, “Password-based authenticated key exchange in the three-party setting,” in *Proceedings of the Public Key Cryptography - PKC 2005*, pp. 65–84, Berlin, Heidelberg, January 2005.

[27] S. Banerjee, V. Odelu, A. K. Das et al., “A provably secure and lightweight anonymous user authenticated session key exchange scheme for internet of things deployment,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8739–8752, 2019.

[28] S. A. Chaudhry, I. L. Kim, S. Rho, M. S. Farash, and T. Shon, “An improved anonymous authentication scheme for distributed mobile cloud computing services,” *Cluster Computing*, vol. 22, no. 1, pp. 1595–1609, 2019.

[29] R. Melki, H. N. Noura, and A. Chehab, “Lightweight multi-factor mutual authentication protocol for IoT devices,” *International Journal of Information Security*, vol. 19, no. 6, pp. 679–694, 2020.

[30] P. K. Panda and S. Chattopadhyay, “A secure mutual authentication protocol for IoT environment,” *Journal of Reliable Intelligent Environments*, vol. 6, no. 2, pp. 79–94, 2020.