Hindawi
Security and Communication Networks
Volume 2021, Article ID 5546334, 12 pages
https://doi.org/10.1155/2021/5546334

*Research Article*

# New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT

**Mourade Azrour**,[1] **Jamal Mabrouki**,[2] **and Rajasekhar Chaganti**[3]

[1]*Computer Sciences Department, Faculty of Sciences and Techniques, Moulay Ismail University, Errachidia, Morocco*
[2]*Laboratory of Spectroscopy, Molecular Modeling Materials Nanomaterial, Water and Environment, CERNE2D, Mohammed V University, Faculty of Science, Rabat, Morocco*
[3]*Expedia Group Inc, Seattle 98119, USA*

Correspondence should be addressed to Mourade Azrour; azrour.mourade@gmail.com

Received 11 March 2021; Accepted 24 April 2021; Published 8 May 2021

Academic Editor: Azeem Irshad



Recently, Internet of Things and cloud computing are known to be emerged technologies in digital evolution. The first one is a large network used to interconnect embedded devices, while the second one refers to the possibility of offering infrastructure that can be used from anywhere and anytime. Due to their ability to provide remote services, IoT and cloud computing are actually integrated in various areas especially in the healthcare domain. However, the user private data such as health data must be secured by enhancing the authentication methods. Recently, Sharma and Kalra projected an authentication scheme for distant healthcare service-based cloud-IoT. Then, authors demonstrated that the proposed scheme is secure against various attacks. However, we prove in this paper that Sharma and Kalra's protocol is prone to password guessing and smart card stolen attacks. Besides, we show that it has some security issues. For that reason, we propose an efficient and secured authentication scheme for remote healthcare systems in cloud-IoT. Then, we prove informally that our projected authentication scheme is secure against multiple attacks. Furthermore, the experimental tests done using Scyther tool show that our proposed scheme can withstand against known attacks as it ensures security requirements.

## 1. Introduction

The Internet of Things (IoT) and cloud computing are revolutionizing many industries such as health and transportation. The IoT is a large network that interconnects objects, computers, and human individuals. These devices are able to sense, process, and communicate data from one end to another one. In addition, cloud computing is a system that allows the customers to access computing resources via the network. The cloud computing provider ensures the protection of a given number of servers that can be used according to the customer needs. Indeed, IoT's growth has been particularly dynamic and has truly revolutionized human personal and professional daily activities. Some of the IoT areas include agricultural [1–4], industrial [5], education [6], healthcare [7–9], and environmental fields [10–13].

Remote patient monitoring relies on computer systems that retrieve health information from individuals in one location and communicate it digitally to health professionals in another location for assessment and advice, as shown in Figure 1. With this kind of service, healthcare provider can continue processing patient's medical data even if the patient stays at home or care facility and reduces the patient readmission rates.

The question of health is systematically at the heart of the human race's concern, even though there are technological advancements in health treatment. Recently, healthcare has taken on great significance, as witnessed by the most recent coronavirus epidemic. Indeed, in areas where the epidemic is spreading, it is increasingly wise to monitor people remotely using health monitoring technology. A variety of monitoring platforms which allow collecting a large volume of healthcare data at the site of treatment, such as patient's vital signs,

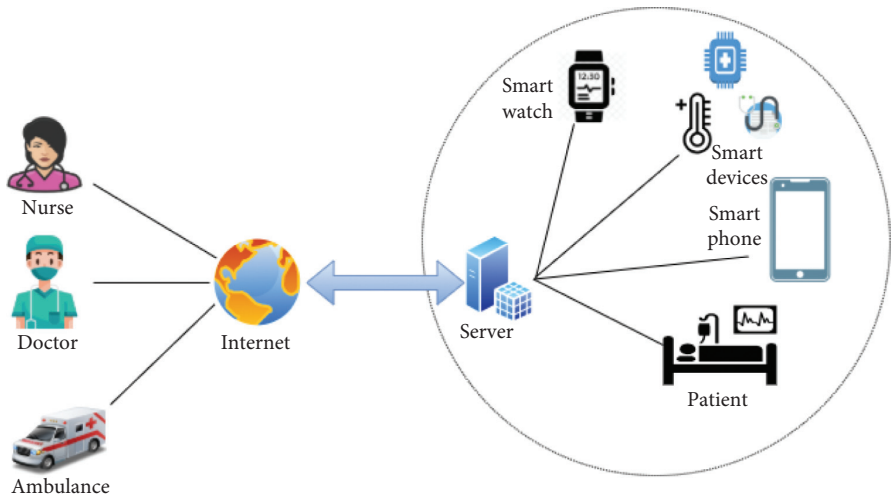


Figure 1: Internet of Thing and healthcare system.

patient's weight, heart rate, blood pressures, blood glucose, blood O2 levels, and electrocardiographs, can be used to monitor the patient's health.

Because of the performance and computing limitations of IoT equipment in handling the massive volume of collected data, it may be appropriate to employ a cloud service to overcome such challenges. Nevertheless, this approach will require warranty of confidentiality, integrity, and security of exchanged data. Subsequently, data are only accessible by authorized entities.

The commonly used solution to assure the confidentiality of data is using such authentication protocols. Hence, Sharma and Kalra [14] proposed a trivial authentication protocol for cloud-IoT-based healthcare system. Formerly, they demonstrated that their proposed protocol is efficient, trivial, and secured against various attacks including DoS attack, man in the middle attack, offline password guessing attack, user impersonation attack, replay attack, and parallel session attack. Authors also use AVISPA tool to evaluate their protocol formally. In this research work, we demonstrated that Sharma and Kalra's scheme poses security vulnerabilities, in particular vulnerable to password guessing attacks. Moreover, some private values described in the protocol are not very secured; it can be obtained basically by any attacker. With the aim to improve the security of cloud-IoT-based healthcare, we propose a new efficient and secured authentication protocol. After proving theoretically that our protocol can resist against various attack, we have done simulation tests under Scyther tool. The obtained results confirm that our scheme can deal against famous attacks, and it guarantees security requirements.

The remaining part of the paper is organized as follows. Related works have been debated in Section 2. Section 3 is reserved for reviewing Sharma and Kalra's protocol. In Section 4, weaknesses of Sharma and Kalra's protocol are discussed. Our proposed efficient and secured authentication scheme is presented in Section 5. In Section 6, informal and formal analyses are detailed. Section 7 provides the performance and comparative analysis. In Section 8, we conclude our paper.

## 2. Related Works

Due to the quick growth and development of various new technologies, personal security, the system for controlling access and the procedures for checking the authenticity of data are gaining significance everyday, particularly since the birth of the IoT. As a consequence, in this section, we discuss some authentication protocols that have been previously presented in literature.

Watro et al. [15] proposed a secured authentication scheme based on RSA for WSNs. Wong et al. [16] proposed authentication scheme that is funded on one way hash functions. This protocol was considered to be secure against many possible attacks, including replay attack, man-in-the-middle attack, forgery attack, and key impersonation attack. Nonetheless, this protocol is proved prone to insider attack and man-in-the-middle attack. As result, Das et al. [17] planned an enhance protocol for gaining more security. Moreover, Xu et al. [18] and Song [19] proposed independently two authentication protocols in 2009 and 2010, respectively. The two proposed protocols are both based on RSA cryptography.

Based on elliptic curve cryptography (ECC), Xu et al. [20] proposed mutual authentication and key convention scheme as a solution of computational problem. Then, he demonstrated that the proposed protocol guarantees confidentiality by using a dynamic identity. Hence, Yan et al. [21] proposed a user authentication system based on biometric detection. However, this protocol cannot resist against replay attacks and is not able to guarantee user anonymity. Furthermore, Mishra et al. proved that Yan's protocol is vulnerable against offline password guessing attacks. Based on those issues, Mishra et al. [22] suggested a new reinforced biometric authentication protocol that uses random digits. Afterword, Tan [23] proposed three-factor mutual authentication protocol.

Yoon and Kim [24] presented user authentication protocol based on a biometric parameter to enhance security of wireless sensors' networks. The proposed scheme is demonstrated secure against some attacks such as DoS attack and sensor impersonation attack.

In 2012, He et al. [25] proposed an authentication protocol, which is efficient for actual medical applications that are based on sensor network. Nevertheless, the scheme is prone to forgery attack and password guessing attack. In addition, it is not capable to offer forward privacy service. In 2014, Mishra et al. [26] rely on chaotic map computation for presenting an authentication and key exchanging protocol for healthcare information organisms. However, this scheme is vulnerable to against password guessing attack.

In 2015, Jiang et al. [27] proved that the protocol proposed by Chen et al. [28] is not secured against password guessing attack. Consequently, with the goal to resolve this issue, Jiang et al. projected a different authentication scheme. Nonetheless, the planned solution is still vulnerable to password guessing and user impersonation attack.

In 2019, Azrour et al. [29] demonstrated that Ye et al.'s [30] protocol is not secured and it has some security issues. In the same year, Cheng et al. [31], based on elliptical curve cryptography and biometrics, proposed a public node identity authentication scheme for numerous categories of devices. In 2020, Azrour et al. [32] proposed a new authentication protocol for IoT devices. Then, authors proved formally and informally that their protocol is efficient and can resist against several attacks.

## 3. Evaluation of Sharma and Kalra's Protocol

In the present section, we present a brief review of three main phases of Sharma and Kalra's scheme, namely, registration, login, and authentication phases. Used notations and their significations are described in Table 1.

### 3.1. Registration Phase

Step 1: user $U_i$ selects her/his identity **I d**, password **pw**, and arbitrary number **R**. He/she computes $\mathbf{MPS} = \mathbf{h}(\mathbf{pw} \| \mathbf{R})$ and sends $\langle \mathbf{I\ d}, \mathbf{MPS} \rangle$to the server via secured channel.

Step 2: server checks received Id. If it exists in database, the server requests a new Id. In other case, the server computes $= H(\text{MPS}\|\text{Id})$, $b = A\ (\text{Id}\|K)$, $c = H(K) \oplus H(\text{MPS}\|b)$, and $d = b \oplus H(\text{MPS}\|a)$. Afterwards, server sends back to user $a, c$, and $d$.

Step 3: user saves $\langle a, c, d, R \rangle$ in it smart card.

### 3.2. Login Phase.

Sharma and Kalra's scheme login phase contains three steps:

Step 1: user $U_i$ inserts her/his identity **I d** and password **pw** in the smart device.

Step 2: the smart device calculates $\mathbf{MPS} = \mathbf{h}(\mathbf{pw} \| \mathbf{R})$based on input pw and stored R.

Step 3: the smart device computes $a' = H(\text{MPS} \| \text{Id})$ and compares its value with stored $a$. In this case, it equals the login phase which is a success.

Table 1: Notations and their significations.

| Symbol | Signification |
|---|---|
| $U_i$ | User (medical professional) |
| $\text{Id}_i$ | User's $U_i$ identity |
| $pw$ | User's $U_i$ password |
| $SN_i$ | The sensor node |
| GN | Gateway node |
| $\text{Id}_{\text{sn}}$ | Identity of sensor node |
| CS | Cloud server |
| $X_s$ | Secret key of CS |
| $K_{\text{CS-SN}i}$ | Shared session key between CS and $\text{SN}_i$ |
| $T1, T2, T3, T4$ | The current time |
| A, B, C, D | High entropy random numbers |
| $h$ | Hash function |
| $\oplus$ | XOR operator |
| $\|\|$ | Concatenation operator |

### 3.3. Authentication Phase.

In this phase, the user $U_i$, the sensor, and the gateway node have to authenticate each other mutually and produce the session key. The steps of this phase are

Step 1: the smart device calculates $b = d \oplus H(\text{MPS}\|a)$ and $H(K) = c \oplus H(\text{MPS} \|b)$. It generates timestamp $T_1$ and computes $V_1 = \text{Id} \oplus H(H(K)\|T_1)$. Then, it chooses random $Ni$ and computes $V_2 = N \oplus H(b\| T_1)$ and $V_3 = H(V_1\|V_2\| Ni \|T_1)$. Next, it transfers this message to gateway node (GN) $\langle V_1, V_2, V_3, T_1, \text{Id}_{\text{SN}} \rangle$.

Step 2: after receiving user's massage, the GN verifies the timestamp. Then, it calculates $\text{MID}_{\text{SN}} = \text{Id}_{\text{SN}} \oplus H(H(K\|z_2))$. It generates random number$Nj$, which is used for computing $V_4 = H(X_{\text{GN-SN}} \| T_1\|T_2) \oplus Nj$ and $V_5 = H(\text{Id}_{\text{SN}}\|V_4\|T_1 \|T_2\|Nj)$. The GN then communicates this message $\langle V_1, V_2, V_3, T_1, T_2, \text{MID}_{\text{SN}}, V_4, V_5 \rangle$ to the SN.

Step 3: upon receiving GN message, the SN verifies the timestamp. Then, it calculates $\text{Id}_{\text{SN}}' = \text{MId}_{\text{SN}} \oplus H$ $(H(K\|T_2))$, $X'_{\text{GN-SN}} = (\text{Id}_{\text{SN}})$, and $Nj' = V_4 \oplus H(X'_{\text{GN-SN}} \|T_1\|T_2)$. Formerly, it checks $V_5' = H(\text{Id}_{\text{SN}}\|V_4\|T_1 \| T_2\|Nj')$. If it is correct, the GN is authenticated. Next, SN computes $\text{Id}' = V_1 \oplus H(H(K\|T_1))$, $b' = H(\text{Id}\|K)$, and $Ni' = V_2 \oplus H(b'\|T_1)$. Then, it checks the validity of $V_3' = H(V_1\|V_2\|Ni'\|T_1)$. If it is ok, the user was authenticated. Therefore, the SN computes its parameters $V_6$, $V_7$, $V_8$, and $V_9$ that will be sent to GN. $V_6 = Nj' \oplus H(b'\|T_3)$, $V_7 = Ni' \oplus H(X'_{\text{GN-SN}}\|T_3)$, $V_8 = H(V_6\| b'\|T_3)$, and $V_9 = H(V_7\|X'_{\text{GN-SN}}\|T_3)$.

Step 4: once the GN receives SN response, it verifies the timestamp. Then, it checks the correctness of $V_9' = H(V_7\|X'_{\text{GN-SN}}\| T_3)$. It computes $\prime = V_7 \oplus H$ $(X'_{\text{GN-SN}} \|T_3)$, $\text{Sk} = H(Ni' \oplus Nj)$, and $V_{10} = H(\text{Sk}$

$\| V_6 \| V_8 \| T_3 \| T_4$). Next, the GN sends to the user this message: $\langle V_6, V_8, V_{10}, T_3, T_4 \rangle$.

Step 5: in this step, the user verifies the validity of timestamp. Then, he/she checks the authenticity of $V_8' = H(V_6 \| b' \| T_3)$ and calculates $Nj' = V_6 \oplus H(b' \| T_3)$ and $\mathrm{Sk} = H(Ni \oplus Nj')$. Finally, the user checks the correctness of $V_{10}' = H(\mathrm{Sk} \| V_6 \| V_8 \| T_3 \| T_4)$.

## 4. Weaknesses of Sharma and Kalra's Protocol

In this section, we demonstrate that user authentication scheme for cloud-IoT-based healthcare services suggested by Sharma and Kalra is defenceless against offline password guessing attacks, and it has some security issues.

*4.1. User Password Guessing Attack.* Sharma and Kalra proved that their protocol can resist against offline password guessing attack even if the smart card is stolen. In opposition to this, we can prove her that an adversary can guess user's password. To do that, he/she has to steal the user's smart card and then recover the value of $R_1$ and $a_i$. Afterword, the adversary runs the dictionary attack to guess the correct password. As it is clear in Figure 2, adversary selects a guessed password from passwords dictionary. Then, he/she computes the value of $\mathrm{MPS}'' \leftarrow h(\mathrm{pw} \| R_1)$ and the value of $h(\mathrm{MPS}'' \| \mathrm{Id}_i)$; if the second value equals $a_i$, the guessed password is correct. Otherwise, the adversary selects another password until discovering the correct one.

*4.2. Impersonation Attack.* Sharma and Kalra demonstrated that their proposed protocol can resist against numerous attacks including impersonation attack. However, in this section, we demonstrate that the user impersonation attack is still operative in Sharma and Kalra's authentication scheme. Accept that an adversary has obtained the contents of a smartcard. He/she can execute the pervious attack to get user's parameters (login and password); then, he/she makes a forged authentication request.

The pirate inserts stolen smart card. Next, he/she enters the deduced $\mathrm{ID}'$ and $\mathrm{Pw}'$. Subsequent, the smart device computes the values of $\mathrm{MPS}' = h(\mathrm{Pw}' \| R)$ and $a' = H(\mathrm{MPS}' \| \mathrm{Id}')$. Then, it verifies if $a' \stackrel{?}{=} a$. The equality will be true because the guessed parameters are verified in the previous attack. Afterward, the smart device will execute the authentication phase. It computes $b' = d \oplus H(\mathrm{MPS}' \| a')$ and $H(K) = c \oplus H(\mathrm{MPS}' \| b')$. It generates timestamp $T_1$ and computes $V_1 = \mathrm{Id} \oplus H(H(K) \| T_1)$. Then, it chooses random $Ni$ and computes $V_2 = N \oplus H(b \| T_1)$ and $V_3 = H(V_1 \| V_2 \| Ni \| T_1)$. Next, it transfers this message to gateway node (GN) $< \langle V_1, V_2, V_3, T_1, \mathrm{Id}_{\mathrm{SN}} \rangle$.

The remainder of the protocol will run normally. The gateway node and sensor node will authenticate the pirate as the valid user and then share with them the session key successfully. Therefore, the pirate can execute user impersonation attack successfully if he has stolen the smart card contents.

Algorithm1 offline_PW_Guessing

```
Begin
  input : Id_i, R_1, a_i, D_PW (Password_dictionary)
  output : PW

for pw' in D_PW do
   MPS'' ← h (pw'||R_1)
   if a_i = h (MPS''||Id_i) then
        return pw'
   end if
end for
end.
```

Figure 2: Algorithm for guessing password offline.

*4.3. Other Security Issues.* Authentication phase is an essential and main phase in all authentication protocols. Indeed, it is a step that assures verification of user identity, allowing authorization and constructing session keys. In healthcare field, it is a key for establishing a secured connection with the remote server and for protecting patient private data. Therefore, it must receive much importance. In our cryptanalysis of Sharma and Kalra authentication protocol for cloud-IoT-based healthcare service, we have observed that there are double serious mistakes. Initially, the sensor node utilizes value of K which is the secret master key of gateway node (GN). Normally, the secret master key of gateway node is a private key. Accordingly, it must be a secret and should not be known to anyone. Otherwise, all systems are at risk and all secret messages will be discovered by any attacker. Secondly, authors propose that the secret shared between the gateway node and sensor node ($\mathrm{SN}_i$) is $X_{\mathrm{GN-SN}_i}$ which is computed as $X_{\mathrm{GN-SN}_i} = h(\mathrm{Id}_{\mathrm{SN}_i})$. However, the value of $\mathrm{Id}_{\mathrm{SN}_i}$ is clearly (not encrypted or hashed) in the first message sent by user ($U_i$) to gateway node (GN). Consequently, if an adversary intercepts this message, he can get easily $\mathrm{Id}_{\mathrm{SN}_i}$. Then, he is able to compute $X_{\mathrm{GN-SN}_i}$.

## 5. Our Proposed Scheme

In this section, we present our new efficient and secured authentication protocol for remote healthcare systems in cloud-IoT. The proposed scheme entails five phases, including system setup phase, new sensor registration phase, user registration phase, login and authentication phase, and password changing phase.

*5.1. System Setup Phase.* In this phase, the superadmin chooses secret key of cloud server $X_s$ and one way hash function $h$. Finally, the server publishes $h$ and saves its private key secretly.

*5.2. New Sensor Registration Phase.* To implement a newly connected sensor node ($\mathrm{SN}_i$) in an already functioning healthcare system, the cloud server has to randomly select both specific $\mathrm{Id}_{\mathrm{sn}}$ and $K_{\mathrm{CS-SN}i}$ as the identification and unique key of the added sensor, respectively. The gateway subsequently uploads $\mathrm{Id}_{\mathrm{sn}}$ and $\mathrm{SK} = h(\mathrm{Id}_{\mathrm{sn}} \| K_{\mathrm{CS-SN}i})$ data to

the sensor memory before running it. In addition, it saves $\mathrm{Id}_{\mathrm{sn}}$ and $\mathrm{HSK} = \mathrm{SK} \oplus h(X_s \| \mathrm{Id}_{\mathrm{SN}})$ in its local database for eventual future utilization.

*5.3. User Registration Phase.* In order to create a count in the cloud server, a medical professional has to perform registration steps with the cloud server. The details of this phase are illustrated in Figure 3.

Step 1: the medical professional selects freely appropriate identity $\mathrm{Id}_i$ and suitable password $\mathrm{pw}_i$. Afterword, he picks arbitrarily two numbers $a$ and $b$. Next, he calculates $\mathrm{MID} = h(\mathrm{Id}_i \| a)$ and $\mathrm{MPW} = h(\mathrm{Id}_i | i | b)$. The two last values are transferred to the cloud server using a secured canal.

Step 2: the cloud server CS picks c randomly and computes $V = h(\mathrm{MID} \| X_s) \oplus h(\mathrm{MPW} \| c)$. Next, the CS saves MID and $c$ in local database and transmits $V$ to medical professional.

Step 3: the medical professional memorizes that information $(V, a, b, \mathrm{MID})$ in a smart card.

*5.4. Login and Authentication Phase.* Once medical professional has accomplished successfully the registration process, he can connect to any sensor node. For doing that, the medical professional must accomplish the login phase by inserting his smart card. Subsequently, the authentication phase is executed. After successful login and authentication, the medical professional is allowed to interact with medical sensor nodes in real time. The steps of login and authentication phase are described below and are depicted in Figure 4.

Step_Auth1: $\mathbf{U} \longrightarrow \mathbf{CS} : \{\mathbf{V}_1, \mathbf{MI\ D}, \mathbf{A}, \mathbf{Id}_{\mathrm{SN}}, \mathbf{T}_1\}$

Firstly, medical professional user types his/her $\mathrm{Id}_i$ and $\mathrm{pw}_i$, and the smart card verifies user's identity by checking $\mathrm{MID} \stackrel{?}{=} h(\mathrm{Id}_i \| a)$. If it is not OK, the process stops. Otherwise, the smart card picks randomly an integer A. Then, it computes $x = V \oplus h(h(\mathrm{Id}_i | \mathrm{pw}_i | b) \| c)$ and $V_1 = h(x \| A)$. Subsequently, it sends to the cloud server this message $\{V_1, \mathrm{MID}, A, \mathrm{Id}_{\mathrm{SN}}, T_1\}$.

Step_Auth2: $\mathbf{CS} \longrightarrow \mathbf{SN} : \{\mathbf{V}_2, \mathbf{B}, \mathbf{MI\ D}, \mathbf{T}_2\}$

After receiving user's communication, the cloud server checks the timestamp $T_2 - T_1 \leq T$. Then, it computes $w_1 = h(\mathrm{MID} \| X_s)$ and verifies if $V_1 \stackrel{?}{=} h(w_1 \| A)$. In the case it is validate, the cloud server generates randomly an integer $B$. Hence, it calculates $w_2 = \mathrm{HSK} \oplus h(\mathrm{Id}_{\mathrm{sn}} \| X_s)$, $\mathrm{HID} = h(\mathrm{MID} \| \mathrm{Id}_{SN})$, and $V_2 = h(\mathrm{HID} \| w_2 \| T_2 \| B)$ Finally, the cloud server forwards this message to the sensor node $\{\mathbf{V}_2, \mathbf{B}, \mathbf{MI\ D}, \mathbf{T}_2\}$.

Step_Auth3: $\mathbf{SN} \longrightarrow \mathbf{CS}$: $\{\mathbf{V}_3, \mathbf{C}, \mathbf{HI\ D}, \mathbf{Id}_{\mathbf{SN}}, \mathbf{T}_3\}$

Once the sensor node SN obtains the message sent by CS. First of all, it checks the authenticity of timestamp $T_3 - T_2 \leq \Delta T$. Next, it calculates the value of $\mathrm{HID}' = h(\mathrm{MID} \| \mathrm{Id}_{SN})$. Then, it checks whether $V_2 \stackrel{?}{=} h(\mathrm{HID}\prime \| \mathrm{SK} \| \mathrm{T}_2 \| \mathrm{B})$ is valid or not. If it is OK, the sensor node SN chooses random integer $C$ and computes $V_3 = h(|\mathrm{MID}||\mathrm{Id}_{\mathrm{SN}}||\mathrm{SK}|\mathrm{T}_3 \| C)$ which is sent back to the cloud server with other values $\{\mathbf{V}_3, \mathbf{C}, \mathbf{HI\ D}, \mathbf{Id}_{\mathbf{SN}}, \mathbf{T}_3\}$.

Step_Auth4: $\mathbf{CS} \longrightarrow \mathbf{U}$: $\{\mathbf{V}_4, \mathbf{D}, \mathbf{Id}_{\mathbf{SN}}, \mathbf{T}_4\}$

When sensor's response is reaching to the cloud server, this last one verifies the validity of timestamp $T_4 - T_3 \leq \Delta T$. Subsequently, it checks if $V_3 \stackrel{?}{=} h(\mathrm{MID} \| w_2 \| T_3 \| C)$ is correct or not. In the case that it is reasonable, and the cloud server picks randomly an integer $D$. At that moment, it computes $V_4 = h(|w_1||\mathrm{MID}||\mathrm{Id}_{\mathrm{SN}}||T_4 \| D)$ and the session key $S_{\mathrm{Key}} = h(w_1 \| \mathrm{MID} \| \mathrm{Id}_{\mathrm{SN}})$. After that, the cloud server sends back the response to the medical professional user $\{\mathbf{V}_4, \mathbf{D}, \mathbf{Id}_{\mathbf{SN}}, \mathbf{T}_4\}\}$.

Step_Auth5:

After reception of cloud server reply, the medical professional user checks the correctness of the timestamp $T_5 - T_4 \leq \Delta T$. Hence, it authenticates the cloud server's message by checking if $V_4 \stackrel{?}{=} h(|x|||\mathrm{MID}|| \mathrm{Id}_{\mathrm{SN}} \| T_4 \| D)$ is true or false. In the case that it is OK, the medical professional user generates the session key $S_{\mathrm{Key}} = h(x||\mathrm{MID}||\mathrm{Id}_{\mathrm{SN}})$.

*5.5. Password Changing Phase.* Naturally, our proposed authentication protocol gives to the medical professional user the possibility to alter his/her password spontaneously. This operation can be completed in a public channel. The steps of this phase are illustrated in Figure 5. Besides, they are detailed in the following.

Step_Chang1: $\mathbf{U} \longrightarrow \mathbf{CS} : \{\mathbf{M}_u\}$

In this step, medical professional user $U$ types in it login and password ($\mathrm{Id}_i$ and $\mathrm{pw}_i$) . Afterwards, he/she verifies $\mathrm{MID} \stackrel{?}{=} h(\mathrm{Id}_i \| a)$. If it is all right, the user selects freely his/her new password $\mathrm{pw}_i^*$ and chooses two arbitrary numbers $a^*$and $b^*$. Next, he/she computes $\mathrm{MID}^* = h(\mathrm{Id}_i \| a^*)$ and $\mathrm{MPW}^* = h(\mathrm{Id}_i || \mathrm{pw}_i^* | b^*|)$. Finally, he/she encrypts the message $M_u = E_{SK}(\mathrm{MPW}||\mathrm{MPW}^*|| \mathrm{MID}||\mathrm{MID}^*||V)$ which will be sent to the cloud server.

Step_Chang2: $\longrightarrow \mathbf{U} : \{\mathbf{M_s}\}$

After receiving user's request the server decrypts the received message $M_u' = D_{SK}(\mathrm{MPW}||\mathrm{MPW}^*||\mathrm{MID} ||\mathrm{MID}^*||V)$. Then, it checks whether $V \stackrel{?}{=} h(\mathrm{MID} \| X_s) \oplus h(\mathrm{MPW} \| c)$ is correct or not. If it is OK, the server selects randomly $c^*$. Then, it replaces MID and $c$ by $\mathrm{MID}^*$ and $c^*$, respectively. Next, it computes$V^* = h(\mathrm{MID}^* \| X_s) \oplus h(\mathrm{MPW}^* \| c^*)$ and $M_s = E_{\mathrm{SK}}(V^*)$. Finally, the server sends back to the user the message $M_s$.

Step_Chang3

Once the server response was received by the user, this last one decrypts the message $M_s' = D_{\mathrm{SK}}(V^*)$ and replaces $V, a, b$, and MID by $V^*$, $a^*, b^*$ and $\mathrm{MID}^*$ respectively.

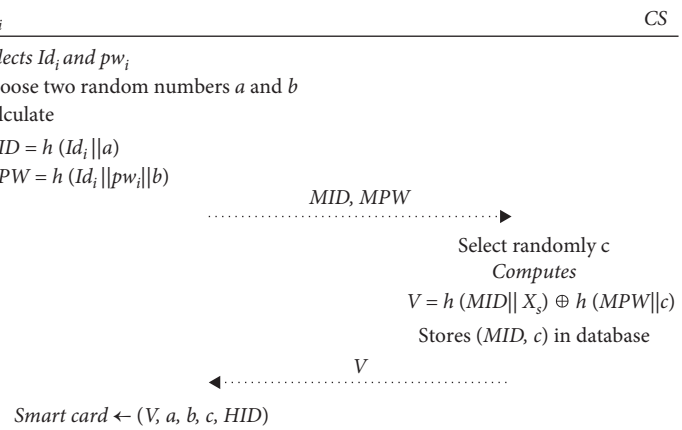


Figure 3: Registration phase.

## 6. Informal and Formal Analyses

In this section, we will analyze the security of our proposed protocol against numerous security attacks. As well as, we present its security properties, including mutual authentication, data integrity, user anonymity, session key exchanging, and forward secrecy (Table 2). The security of our proposed scheme is proved by both informal and formal analysis.

### 6.1. Informal Analyses

*6.1.1. Session Key Exchanging.* In our proposed protocol, the session key is generated by the user and cloud server as $S_{\text{Key}} = h(x||\text{MID}||\text{Id}_{\text{SN}})$, where $x = V \oplus h(h(|\text{Id}_i||\text{pw}_i||b)\|c) = h(\text{MID}\|X_s)$. Thanks to the reason that $X_s$ and MID are secret, and the session key cannot be known at the end of login and authentication phase except for the medical professional user and the cloud server. As a result, we can say that the proposed scheme guarantees session key secrecy.

*6.1.2. Mutual Authentication.* In a public unsecured channel, each entity has authenticated each other before authentic communication takes place. So, owing to the advantages of mutual authentication in a network environment, our planned protocol reassures mutual authentication. Hence, the cloud server authenticates both the medical professional user and the sensor node. To verify users authenticity in Step 2, the server checks the correctness of received$V_1$. For checking the sensor identity, in Step 4, the cloud server verifies the exactness of $V_3 \stackrel{?}{=} h(\text{MID}\|w_2\|T_3\|C)$. Additionally, the medical professional user is able to authenticate the cloud server identity in Step 5, through examination of the legitimacy of $V_4 \stackrel{?}{=} h(|x||\text{MID}||\text{Id}_{\text{SN}}\|T_4\|D)$. Hereafter, in Step 3, the sensor node also authenticates cloud servers' message by checking the accuracy of $V_2 \stackrel{?}{=} h(\text{HID}'\|\text{SK}\|T_2\|B)$.

*6.1.3. Data Integrity.* It is very essential, while forwarding information between the various IoT terminals, to ensure that the data are correct and belong to the authenticated sender. Besides, it is very indispensable to ensure that the data have not been falsified by the authorities during the transfer by invoking fraudulent acts or forgery attacks. In the proposed protocol, if we suppose that an attacker captures $V_1$, $V_2$, $V_3$, or $V_4$. Then, he tries to alter their values. However, the receivers will detect this modification using the timestamp. In addition, the modification timestamps will be identified since the timestamps are embedded into $V_1$, $V_2$, $V_3$, and $V_4$.

*6.1.4. DoS Attack.* Our proposed authentication protocol can resist against DoS attack. Accordingly, the user is able to know if her/his message has passed the authentication phase or not, especially, after getting server's response which can be validated or rejected message. With goals to verify the freshness of received message, our protocol uses timestamps. Furthermore, arbitrary numbers are produced in every stage and in every session. Besides, since the duplicated messages are unacceptable, the attacker is not able to perform the DoS attack. Thus, our suggested method can resist against DoS attack.

*6.1.5. No Verification Table.* According to the proposed protocol, the confidential information of each user, including the password, is not stored by the cloud server or the sensors. In case an attacker succeeds in the hacking cloud server or node, he/she would not be actually capable of obtaining the password checking data. Therefore, the hacker will not be able to retrieve any authenticating details.

*6.1.6. Off-Line Password Guessing Attack.* Assume that an attacker has got the smart card. Then, he/she extracts all stored parameters in it. If he/she wants to guess the password, he/she cannot, since the only value that contains password is $V = h(\text{MID}\|X_s) \oplus h(\text{MPW}\|c)$. Therefore, the attacker has to know the value of $X_s$ and $\text{Id}_i$. However, $\text{Id}_i$ is encrypted using one-way hash function, and $X_s$ is server's private key. Consequently, we can conclude that our proposed scheme is secure against offline password guessing attack.

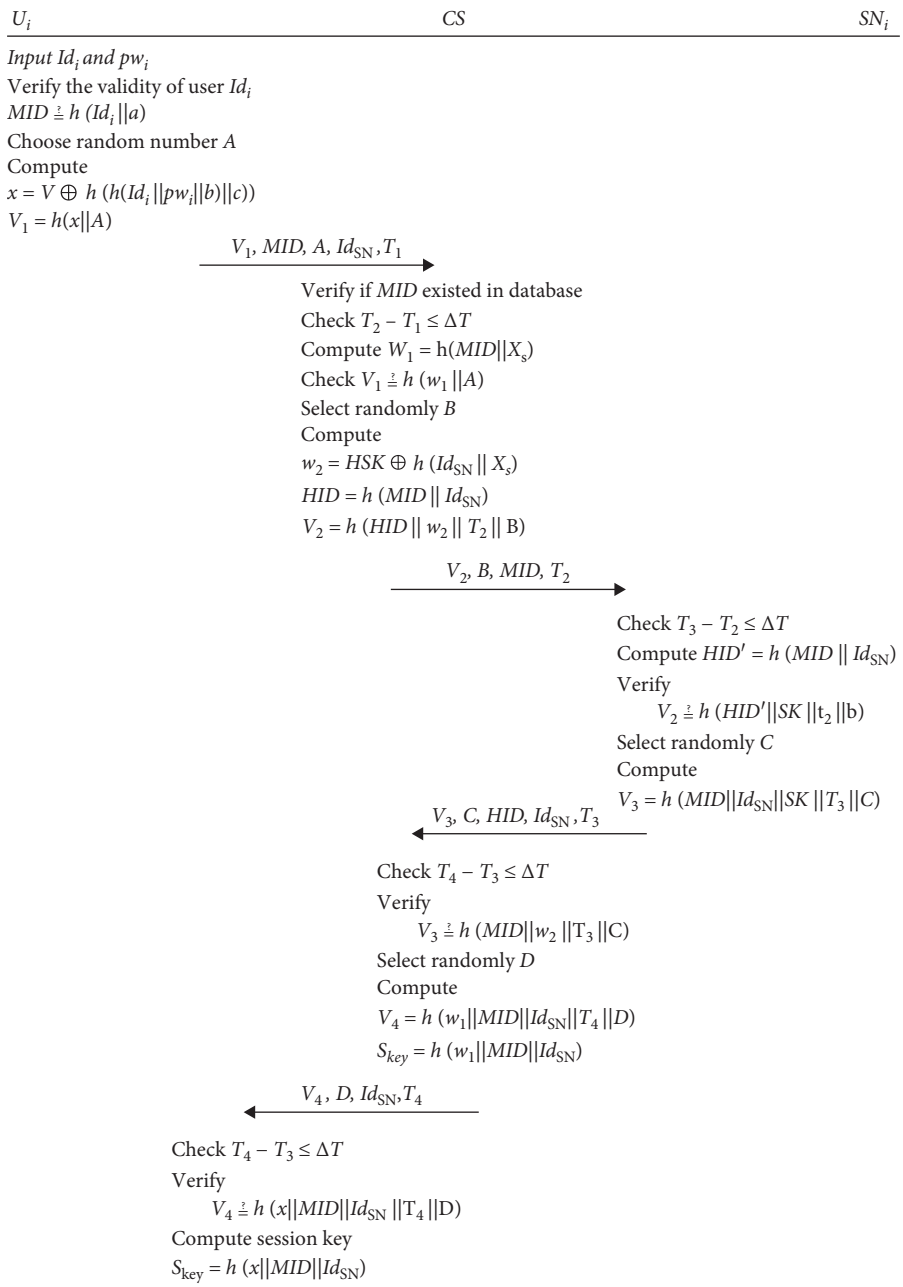


Figure 4: Login and authentication phase.

*6.1.7. Replay Attack.* Assume that an adversary replays the old message to the server. In our proposed protocol, the cloud server discovers that this message is not fresh. Originally, the cloud server checks the validity of timestamp $T_2 - T_1 \leq \Delta T$; in the case that it is noted valid, the session stops. The same thing happens after receiving sensor node's message $T_4 - T_3 \leq \Delta T$. The sensor node and user use $T_3 - T_2 \leq \Delta T$ and $T_5 - T_4 \leq \Delta T$, respectively, to check the newness of cloud server's message. Consequently, our proposed protocol can withstand against replay attack.

*6.1.8. Insider Attack.* Our proposed scheme can resist against privileged insider attack. Assume that a malicious or pirate has an access the registration data {MID,c }. Even if we have those data, the attacker can neither guess the password nor initiate any kind of counterfeit attack. Furthermore, he/she must fight the secrecy of one way hash function if he/she wants to have just the user Id. On the contrary, for initiating impersonation attack, the attacker must have access to cloud server's secret key. Consequently, our proposed protocol can resist against privileged insider attack.

*6.1.9. Perfect Forward Secrecy.* In our proposed protocol, the session key $S_{Key}$ is computed as $S_{Key} = h(w_1||MID||Id_{SN})$, where $w_1 = h(MID||X_s)$. The session key contains server's secret key $X_s$ and users MID that depend on user's encrypted

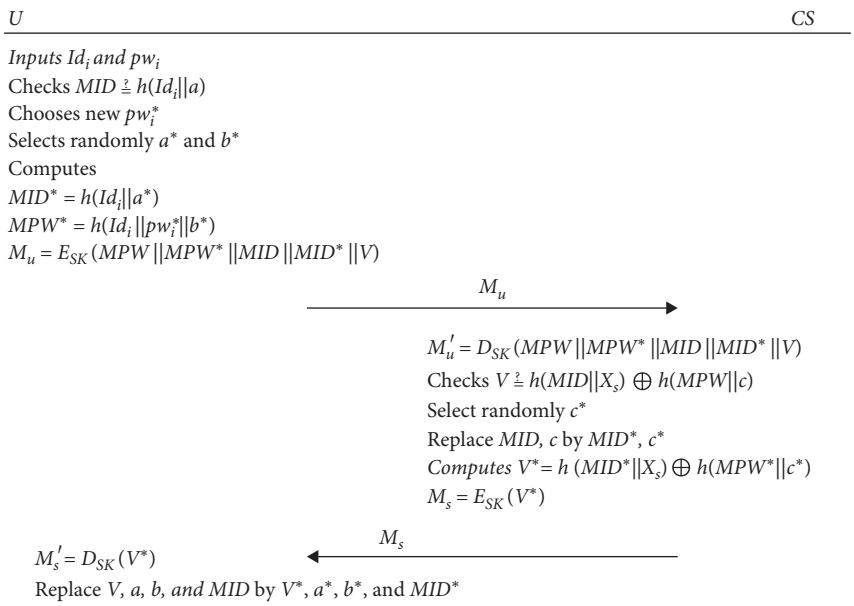


Figure 5: Password changing phase.

Table 2: Security feature's comparison.

| | Kumar et al. [33] | He et al [34] | Amin et al [35] | Sharma and Kalra [14] | Ours |
|---|---|---|---|---|---|
| Mutual authentication | ✓ | ✓ | ✓ | ✓ | ✓ |
| Data integrity | ✓ | ✓ | ✓ | ✓ | ✓ |
| No verification table | ✗ | ✗ | ✓ | ✓ | ✓ |
| Session key exchanging | ✓ | ✓ | ✓ | ✓ | ✓ |
| DoS attack | ✓ | ✓ | ✓ | ✓ | ✓ |
| Perfect forward secrecy | ✗ | ✗ | ✗ | ✓ | ✓ |
| Off-line password guessing attack | ✗ | ✗ | ✗ | ✗ | ✓ |
| Replay attack | ✗ | ✗ | ✗ | ✓ | ✓ |
| Insider attack | ✗ | ✗ | ✗ | ✓ | ✓ |

✓: secured against attack. ✗: not secured against attack.

$Id_i$. In other words, the session key $S_{Key}$ depends on secure parameters which are not accessible to the attacker. Consequently, our proposal assures perfect forward secrecy.

## 7. Formal Analyses

*7.1. Security Examination Using Scyther.* In this section, we initially clarify the usefulness of Scyther tool [36], which was useful for formal security examination of our proposed scheme. Formerly, we showed gained outcomes by using this tool. Absolutely, it is software for automatically checking of security protocols. It is based on a reverse analysis method. The requests correspond to the knowledge of the participants (source and destination) and also to the wishes of a possible hacker. Symmetric and asymmetric encryption, hash functions, and encryption keys are also embedded in Scyther.

Our planned protocol is then written in the security protocol description language (SPDL). This specification allows us to specify different roles of the medical professional user, the cloud server, and the sensor node. In each role, event sequences are embedded including sending, receiving, declarations, and complaints. According to Figure 6, which details the obtained results, we can notice that our protocol is secure against many attacks. Besides, it meets the necessary security-related fundamentals.

*7.2. Security Verification Using Random Oracle Model.* Actually, the random oracle model is used for formal security analysis. In this analysis, we verify that a given attacker is not in measure to recover important secret values including Id, Pw, $V_1$Idi, and session key $S_{Key}$. In our study, the similar procedure presented in [37–39] is adopted. The random oracle is detailed in the following.

Reveal: it produces the input of a hash function; let $\lambda$ be absolutely from a specified hash output $\psi$, where $\psi = f(\lambda)$.

*Formula 1.* Let us suppose that an attacker has stool user's smart device; in addition, he has the knowledge about the transactions $\{V_1 - V_4, T_1 - T_4, Id_{SN}, MID\}$ that has been transmitted over an untrusted network. While, $h$ is the hash function that is understood as a random oracle, and the introduced protocol is secured against the attacker to retrieve the Id, Pw, $V_1$, and the session key $S_{Key}$ of legitimate user $U$.

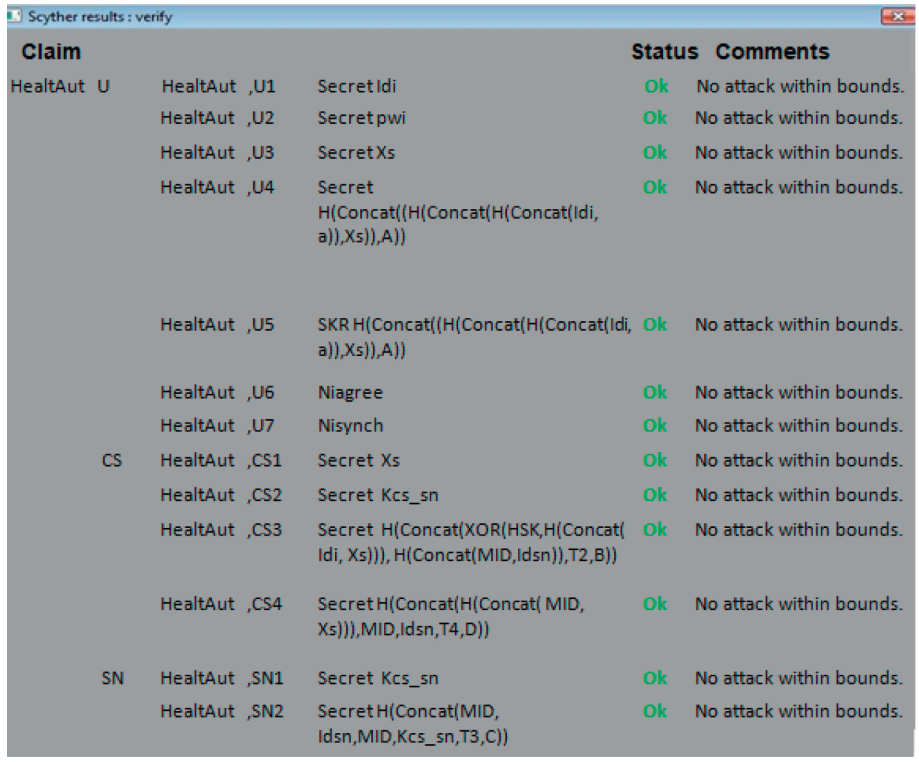


Figure 6: Experimental test results.

*Proof.* Suppose that the attacker $\mathscr{A}$ has obtained user's parameters Id and Pw by using smart card data $\{V, a, b, \text{MID}\}$ and the intercepted messages $\{V_1 - V_4, T_1 - T_4, \text{Id}_{\text{SN}}, \text{MID}\}$, and the tentative algorithm $\text{Tent1}^{\text{hash}}_{\mathscr{A},\text{RPMAP}}$ defines the possibility of achievement $\text{Ach}_1$ for $\text{Exp1}^{\text{hash}}_{\mathscr{A},\text{RPMAP}}$ by means of the following specified function:

$$\text{Ach}_1 = \left|\text{Pb} \cdot \left[\text{Tent}\,1^{\text{hash}}_{\mathscr{A},\text{RPMAP}} = 1 - 1\right]\right|. \quad (1)$$

$\text{Pb}[E]$ denotes the probability for a given event E. In this experiment, we define the advantageous condition as $F_{Av1}(t_1, rq_1) = \text{Max}_{\mathscr{A}}\{\text{Ach1}\}$, where Max is determined based on totally adversaries taking the execution time $t_1$ and $\text{rq}_1$ is the total maximum requests transmitted to the reveal oracle. Our presented protocol is secure against the adversary to discover Id, Pw, and $V_1$ if $F_{Av1}(t_1, \text{rq}_1) \le \varepsilon$ for a small value of $\varepsilon > 0$. The trial model $\text{Tent}\,1^{\text{hash}}_{\mathscr{A},\text{RPMAP}}$ indicates that if the hacker is able to compute the inverse of the hash function, it can recover user's parameters Id, Pw, and $V_1$. Nevertheless, it is impossible to determine the reverse of this one-way hash function in polynomial period, such that $F_{Av1}(t_1, rq_1) \le \varepsilon$ for a small value of $\varepsilon > 0$. Accordingly, we can say that our proposed scheme is safe from the attacker for obtaining Id, Pw, $V_1$, and $S_{\text{Key}}$. □

*Formula 2.* If we suppose that the hash function behaves as random oracle and the attacker have intercepted the message forwarded on unsecured channel $\{V_1 - V_4, T_1 - T_4, \text{Id}_{\text{SN}}, \text{MID}\}$, the attacker may not be able to calculate user's session key $S_{\text{Key}}$.

*Proof.* If an attacker that intercepts the transferred message $\{V_1 - V_4, T_1 - T_4, \text{Id}_{\text{SN}}, \text{MID}\}$ tries to generate user's session key $S_{\text{Key}}$, the probability of achievement $\text{Ach}_1$ in this calculation is defined by the tentative algorithm $\text{Tent2}^{\text{hash}}_{\mathscr{A},\text{RPMAP}}$ as

$$\text{Ach}_2 = \left|\text{Pb} \cdot \left[\text{Tent}\,2^{\text{hash}}_{\mathscr{A},\text{RPMAP}} = 1 - 1\right]\right|. \quad (2)$$

$\text{Pb}[E]$denotes the probability for a given event E. In this experiment, we define the advantageous function as $F_{Av2}(t_2, rq_2) = \text{Max}_{\mathscr{A}}\{\text{Ach2}\}$, where Max is determined based on totally adversaries taking the execution time $t_1$ and $rq_1$ is the total maximum requests transmitted to the reveal oracle. Our presented protocol is secure against the adversary to discover $S_{\text{Key}}$ if $F_{Av2}(t_2, rq_2) \le \varepsilon$ for a small value of $\varepsilon > 0$. The trial model $\text{Tent2}^{\text{hash}}_{\mathscr{A},\text{RPMAP}}$ indicates that if the hacker is able to compute the inverse of the hash function, it can recover user's parameters Id, Pw, and $V_1$. Nevertheless, it is impossible to determine the reverse of this one-way hash function in legal period, such that $F_{Av2}(t_2, rq_2) \le \varepsilon$ for a small value of $\varepsilon > 0$. Accordingly, we can close that our proposed scheme is safe from the attacker for computing $S_{Key}$ (Algorithm 1 and 2 ). □

## 8. Performance and Comparative Analysis

This section details the results of the performance analysis of our protocol. In the first place, we display the performance of our proposed protocol in point of view of the ability to resist against security attacks. Secondly, our protocol is compared

Begin
(1) Intercept the transmitted values $\{\mathbf{V}_1, \mathbf{V}_4, \mathbf{MI\ D}, \mathbf{A}, \mathbf{Id_{SN}}, \mathbf{T}_1, \mathbf{T}_4\}$
(2) Call reveal oracle on $\mathbf{V}_4$ for getting value of $\{(x\|MI\ D\|\mathrm{Id}_{\mathrm{SN}}\|T_4\|D)\}$ *as* $(x\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}}\|T_4\|D)) \leftarrow reveal1\ (\mathbf{V}_4)$
(3) Call reveal oracle on $\mathbf{V}_1$ for getting value of $\{(x\|A)\}$ *as* $(x\|A) \leftarrow reveal1\ (\mathbf{V}_1)$
(4) Calculate $x' = V \oplus h((h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c))$
(5) If $(x' \stackrel{?}{=} x)$, then
(6) Extract the parameters $\{\mathbf{V}, \mathbf{a}, \mathbf{b}, \mathbf{MI\ D}\}$ from the mobile device.
(7) Calculate $h(h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c) = V \oplus x$
(8) Call reveal oracle on input $h(h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c$ to discover $\{h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c\}$ as $(h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c \leftarrow reveal1(h(h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\|c)$.
(9) Call reveal oracle on input $h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)$ to discover $\{(\mathrm{Id}_i\|\mathrm{pw}_i\|b)\}$ as $(\mathrm{Id}_i\|\mathrm{pw}_i\|b) \leftarrow reveal1\ (h(\mathrm{Id}_i\|\mathrm{pw}_i\|b)$.
(10) Ten, compute MID' $= h(\mathrm{Id}_i\|a)$
(11) If (MID′$\stackrel{?}{=}$ MID), then
Accept $\mathrm{Id}_i'$, $\mathrm{pw}_i'$, and $a'$ as valid identity, password, and random number.
Return (true)
(12) Else
Return (false)
(13) End if
End.

Algorithm 1: Tent $1^{\mathrm{hash}}_{A,\mathrm{RPMAP}}$.

Begin
(14) Intercept the transmitted values $\{\mathbf{V}_1, \mathbf{V}_4, \mathbf{MI\ D}, \mathbf{A}, \mathbf{Id_{SN}}, \mathbf{T}_1\}$
(15) Call reveal oracle on $\mathbf{V}_4$ for getting value of $\{(x\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}}\|T_4\|D)\}$ *as* $(x\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}}\|T_4\|D) \leftarrow reveal1\ (\mathbf{V}_4)$
(16) Call reveal oracle on $\mathbf{V}_1$ for getting value of $\{(x\|A)\}$ *as* $(x\|A) \leftarrow reveal1\ (\mathbf{V}_1)$
(17) Calculate $w_1 = h(\mathrm{MID}\|X_s)$
(18) Generate the session key $S_{\mathrm{key}}' = h(w_1\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}})$
(19) Compute $V_4' = (x\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}}\|T_4\|D)$
(20) If $(V_4' \stackrel{?}{=} \mathbf{V}_4)$, then
Accept $S_{\mathrm{key}}' = h(w_1\|\mathrm{MID}\|\mathrm{Id}_{\mathrm{SN}})$ as effective user's session key.
Return (true)
(21) Else
Return (false)
(22) End if
End.

Algorithm 2: Tent $2^{\mathrm{hash}}_{A,\mathrm{RPMAP}}$.

to other related ones according to the computational complexity. Hence, the results of the first comparison are illustrated in Table 2. As it is very clear, we can realize that our protocol can resist against various attacks, and it is able to guarantee several security requirements including perfect forward secrecy, session key exchange, and mutual authentication.

As we have mentioned above, the calculation charges of our proposed scheme is compared to other correlated protocols specifically Alzahrani et al. [40], Li et al. [41], Azrour et al. [32], and Sharma and Kalra [14]. In this calculation, very weedy procedures such as string concatenation operation and XoR procedure are ignored. The sign $T_h$ personifies the time charge of one way hash operation, whereas $T_{\mathrm{ED}}$ characterizes the time complexity of symmetric key operations and $T_{\mathrm{pm}}$ denotes the computational charge of elliptic curve point multiplication.

In our protocol, the medical professional user calculates $6T_h$ and the cloud server computes $8T_h$, while the sensors node calculates $3T_h$. Consequently, the whole calculation complexity of our scheme is only $17T_h$. As displayed in Table 3, one can remark that our scheme is based only on one-way hash functions which do not consume much time if it is compared to the symmetric key operations. In case we compare the number of time that $T_h$ uses, we will find that our protocol uses only 17. Hence, we confirm that our proposed protocol is appropriate for remote healthcare applications based one cloud-IoT.

Table 3: Comparative analysis.

| | Alzahrani et al [40] | Li et al. [41] | Sharma and Kalra [14] | Azrour et al. [32] | Ours |
|---|---|---|---|---|---|
| User | $1T_{ED} + 11T_h$ | $2T_{ED} + 6T_h$ | $11T_h$ | $5T_h$ | $6T_h$ |
| Server/gateway | $7T_h$ | $6T_{ED} + 7T_h$ | $7T_h$ | $6T_h + 4T_{pm}$ | $8T_h$ |
| Sensor node | $1T_{ED} + 5T_h$ | $2T_{ED} + 5T_h$ | $5T_h$ | $2T_h + 2T_{pm}$ | $3T_h$ |
| Total | $2T_{ED} + 23T_h$ | $10T_{ED} + 18T_h$ | $23T_h$ | $13T_h + 6T_{pm}$ | $17T_h$ |

## 9. Conclusions

Modern technologies are currently having a great impact on the healthcare world. Thus, healthcare professionals can have access to patient confidential data online, which mandates strong authentication protocols for home patient monitoring. So, it is necessary to consider a lightweight authentication scheme to guarantee secure communication between the healthcare system-based cloud-IoT. In the present study, we firstly demonstrated that the protocol proposed by Sharma and Kalra is vulnerable and exposes some security issues. Then, we have proposed our authentication protocol to mitigate the prior work vulnerable issues. Afterwards, we have demonstrated informally that our protocol can resist against various attacks and can provide security requirement. In addition, the simulation done under Scyther tools confirms that our protocol is formally secured and meets security fundamentals.

## Data Availability

Experimental results, obtained using Scyther tool, are available and will be shared with authors at https://sites.Google.com/umi.ac.ma/azrour.

## Conflicts of Interest

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

## References


[1] D. Pivoto, P. D. Waquil, E. Talamini, C. P. S. Finocchio, V. F. Dalla Corte, and G. de Vargas Mores, "Scientific development of smart farming technologies and their application in Brazil," *Information Processing in Agriculture*, vol. 5, no. 1, pp. 21–32, 2018.

[2] P. Visconti, N. I. Giannoccaro, R. d. Fazio, S. Strazzella, and D. Cafagna, "IoT-oriented software platform applied to sensors-based farming facility with smartphone farmer app," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 3, pp. 1095–1105, 2020.

[3] G. I. Hapsari, G. Andriana Mutiara, L. Rohendi, and A. Mulia, "Wireless sensor network for monitoring irrigation using XBee Pro S2C," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1345–1356, 2020.

[4] M. S. U. Chowdury, T. B. Emran, S. Ghosh et al., "IoT based real-time river water quality monitoring system," *Procedia Computer Science*, vol. 155, pp. 161–168, 2019.

[5] Z. Mohd Yusoff, Z. Muhammad, M. S. I. Mohd Razi, N. F. Razali, and M. H. C. Hashim, "IOT-Based smart street lighting enhances energy conservation," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, p. 528, 2020.

[6] J. Poushter, "Smartphone ownership and internet usage continues to climb in emerging economies," *Pew Research Center*, vol. 9, no. 4, pp. 1–44, 2016.

[7] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.

[8] M. S. Hossain, G. Muhammad, and A. Alamri, "Smart healthcare monitoring: a voice pathology detection paradigm for smart cities," *Multimedia Systems*, vol. 25, no. 5, pp. 565–575, 2019.

[9] Y.-T. Park, "Emerging new era of mobile health technologies," *Healthcare Informatics Research*, vol. 22, no. 4, pp. 253-254, 2016.

[10] J. Mabrouki, M. Azrour, G. Fattah, D. Dhiba, and S. E. Hajjaji, "Intelligent monitoring system for biogas detection based on the Internet of Things: mohammedia, Morocco city landfill case," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 10–17, 2021.

[11] J. Mabrouki, M. Azrour, D. Dhiba, Y. Farhaoui, and S. E. Hajjaji, "IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 25–32, 2021.

[12] J. Mabrouki, M. Azrour, Y. Farhaoui, and S. El Hajjaji, "Intelligent system for monitoring and detecting water quality," in *Big Data and Networks Technologies*, Y. Farhaoui, Ed., vol. vol. 81, pp. 172–182, Springer International Publishing, Cham, Switzerland, 2020.

[13] J. Mabrouki, M. Azrour, and S. El Hajjaji, "Use of internet of things for monitoring and evaluation water's quality: comparative study," *International Journal of Cloud Computing*, 2021, In press.

[14] G. Sharma and S. Kalra, "A lightweight user authentication scheme for cloud-IoT based healthcare services," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 43, no. 1, pp. 619–636, 2019.

[15] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, and P. Kruus, "TinyPK: securing sensor networks with public key technology," in *Proceedings Of the 2nd ACM Workshop on Security Of Ad Hoc and Sensor Networks*, pp. 59–64, San Diego, CA, USA, 2004.

[16] K. H. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," *Sensor Networks, Ubiquitous, and Trustworthy Computing*, vol. 1, p. 8, 2006.

[17] M. L. Das, "Two-factor user authentication in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1086–1090, 2009.

[18] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.

[19] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards & Interfaces*, vol. 32, no. 5-6, pp. 321–325, 2010.

[20] X. Xu, Z. P. Jin, H. Zhang, and P. Zhu, "A dynamic ID-based authentication scheme based on ECC for telecare medicine information systems," *In Applied Mechanics And Materials*, vol. 457, pp. 861–866, 2014.

[21] X. Yan, W. Li, P. Li, J. Wang, X. Hao, and P. Gong, "A secure biometrics-based authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 37, no. 5, pp. 1–6, 2013.

[22] D. Mishra, S. Mukhopadhyay, S. Kumari, M. K. Khan, and A. Chaturvedi, "Security enhancement of a biometric based authentication scheme for telecare medicine information systems with nonce," *Journal of Medical Systems*, vol. 38, no. 5, pp. 1–11, 2014.

[23] Z. Tan, "A user anonymity preserving three-factor authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 3, pp. 1–9, 2014.

[24] E.-J. Yoon and C. Kim, "Advanced biometric-based user authentication scheme for wireless sensor networks," *Sensor Letters*, vol. 11, no. 9, pp. 1836–1843, 2013.

[25] D. He, C. Chen, S. Chan, J. Bu, and A. V. Vasilakos, "ReTrust: attack-resistant and lightweight trust management for medical sensor networks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 4, pp. 623–632, 2012.

[26] D. Mishra, J. Srinivas, and S. Mukhopadhyay, "A secure and efficient chaotic map-based authenticated key agreement scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 10, p. 120, 2014.

[27] Q. Jiang, J. Ma, G. Li, and X. Li, "Improvement of robust smart-card-based password authentication scheme," *International Journal of Communication Systems*, vol. 28, no. 2, pp. 383–393, 2015.

[28] Z.-Y. Cheng, Y. Liu, C.-C. Chang, and S.-C. Chang, "An improved protocol for password authentication using smart cards," vol. 22, no. 4, 2012.

[29] M. Azrour, M. Ouanan, Y. Farhaoui, and A. Guezzaz, "Authentication Protocol for Internet of Things," *Studies in Big Data*, vol. 53, pp. 67–74, 2019.

[30] N. Ye, Y. Zhu, R.-c. Wang, R. Malekian, and L. Qiao-min, "An efficient authentication and access control scheme for perception layer of internet of things," *Applied Mathematics & Information Sciences*, vol. 8, no. 4, pp. 1617–1624, 2014.

[31] X. Cheng, Z. Zhang, F. Chen et al., "Secure identity authentication of community medical internet of things," *IEEE Access*, vol. 7, pp. 115966–115977, 2019.

[32] M. Azrour, J. Mabrouki, A. Guezzaz, and Y. Farhaoui, "New enhanced authentication protocol for Internet of Things," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 1–9, 2021.

[33] P. Kumar, S.-G. Lee, and H.-J. Lee, "E-SAP: efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks," *Sensors*, vol. 12, no. 2, pp. 1625–1647, 2012.

[34] D. He, N. Kumar, J. Chen, C.-C. Lee, N. Chilamkurti, and S.-S. Yeo, "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Systems*, vol. 21, no. 1, pp. 49–60, 2015.

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

[36] C. J. F. Cremers, "The scyther tool: verification, falsification, and analysis of security protocols," *Computer Aided Verification*, pp. 414–418, 2008, In press.

[37] N. Koblitz and A. J. Menezes, "The random oracle model: a twenty-year retrospective," *Designs, Codes and Cryptography*, vol. 77, no. 2-3, pp. 587–610, 2015.

[38] J.-S. Coron, J. Patarin, and Y. Seurin, "The random oracle model and the ideal cipher model are equivalent," *In Advances in Cryptology*, Springer, Berlin, Germany, pp. 1–20, 2008.

[39] M. Bellare and P. Rogaway, "Random oracles are practical," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, New York, NY, USA, 1993.

[40] B. A. Alzahrani, A. Irshad, K. Alsubhi, and A. Albeshri, "A secure and efficient remote patient-monitoring authentication protocol for cloud-IoT," *International Journal of Communication Systems*, vol. 33, no. 11, p. e4423, 2020.

[41] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," *Security and Communication Networks*, vol. 9, no. 15, pp. 2643–2655, 2016.