sensors

MDPI

Article

A Secure and Lightweight Authentication Protocol for IoT-Based Smart Homes

JiHyeon Oh [1], SungJin Yu [1,2], JoonYoung Lee [1], SeungHwan Son [1], MyeongHyun Kim [1] and YoungHo Park [1,3,*]

[1] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Korea; chldlstnr071@knu.ac.kr (J.O.); darkskiln@knu.ac.kr (S.Y.); harry250@knu.ac.kr (J.L.); sonshawn@knu.ac.kr (S.S.); kimmyeong123@knu.ac.kr (M.K.)
[2] Electronics and Telecommunications Research Institute, Daejeon 34129, Korea
[3] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea
* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-7842

Abstract: With the information and communication technologies (ICT) and Internet of Things (IoT) gradually advancing, smart homes have been able to provide home services to users. The user can enjoy a high level of comfort and improve his quality of life by using home services provided by smart devices. However, the smart home has security and privacy problems, since the user and smart devices communicate through an insecure channel. Therefore, a secure authentication protocol should be established between the user and smart devices. In 2020, Xiang and Zheng presented a situation-aware protocol for device authentication in smart grid-enabled smart home environments. However, we demonstrate that their protocol can suffer from stolen smart device, impersonation, and session key disclosure attacks and fails to provide secure mutual authentication. Therefore, we propose a secure and lightweight authentication protocol for IoT-based smart homes to resolve the security flaws of Xiang and Zheng's protocol. We proved the security of the proposed protocol by performing informal and formal security analyses, using the real or random (ROR) model, Burrows–Abadi–Needham (BAN) logic, and the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. Moreover, we provide a comparison of performance and security properties between the proposed protocol and related existing protocols. We demonstrate that the proposed protocol ensures better security and lower computational costs than related protocols, and is suitable for practical IoT-based smart home environments.

Keywords: smart homes; IoT; authentication; BAN logic; ROR model; AVISPA

Citation: Oh, J.; Yu, S.; Lee, J.; Son, S.; Kim, M.; Park, Y. A Secure and Lightweight Authentication Protocol for IoT-Based Smart Homes. *Sensors* **2021**, *21*, 1488. https://dx.doi.org/10.3390/s21041488

Academic Editor: Sara Comai

Received: 15 January 2021
Accepted: 13 February 2021
Published: 21 February 2021

1. Introduction

With the development of information and communication technologies (ICT) and Internet of Things (IoT), smart home automation systems are receiving a lot of attention. The smart home is a networking environment that connects smart devices (e.g., IoT and sensors) to each other. Based on these smart devices, users can utilize various home services. When the user is inside the home, the user can control all smart devices with a voice commands or applications, granting the user accesses to services such as turning the TV on/off, choosing music, switching lights on/off, and so on. When the user is outside the home, the user can monitor and control various smart devices by checking their status. Thus, users can enjoy a high level of comfort and an increased quality of life through smart home environments.

Generally, smart home environments consist of the user, smart devices, a home gateway, and a registration authority [1–3]. A remote user wants to use the data collected by smart devices. However, smart devices are resource limited in terms of computational power, amount of memory, and bandwidth [4]. For these reasons, smart devices communicate through the home gateway. The home gateway acts as a bridge between smart devices and remote users by providing short and long-distance wireless communication interfaces

that maintain the connectivity with internal smart devices and remote users [5]. Users can remotely operate smart devices with the help of a home gateway using Internet-enabled mobile phones and tablets anytime and anywhere. Thus, the home gateway plays a crucial role by controlling the data exchange. It manages the communication between internal and external surroundings.

Unfortunately, the smart home has security and privacy problems because the sensitive data collected by smart devices are exchanged through wireless networks. If an adversary obtains the data, the adversary will abuse them for his own purposes. Thus, security and privacy are essential elements to providing secure home services. In addition, the exchanged data should meet confidentiality, integrity, and availability standards. Asymmetric and symmetric key cryptosystems are inappropriate for applying to low-capacity devices because they generate high computational costs. Thus, secure and lightweight authentication protocols are necessary to provide security and privacy in IoT-based smart homes.

In 2020, Xiang and Zheng [6] proposed a situation-aware protocol for device authentication in smart grid-enabled smart home environments. Xiang and Zheng claimed that their protocol can withstand impersonation, man-in-the-middle (MITM), and replay attacks. Xiang and Zheng also demonstrated that their protocol can provide data integrity and mutual authentication. However, herein we prove that their protocol does not prevent stolen smart device, impersonation, and session key disclosure attacks, and fails to ensure mutual authentication. They also mentioned that their protocol concentrates on the security of smart grid-enabled smart home environments. However, they proposed an authentication protocol that is only for smart home environments. Thus, we focus on general smart home environments and present a secure and lightweight authentication protocol for IoT-based smart homes that deals with the security drawbacks of Xiang and Zheng's protocol [6]. The proposed protocol is efficient for resource-constrained smart devices because we use only one-way hash functions and XOR operations.

1.1. Contributions

This paper has the following main contributions.

- We analyze the security vulnerabilities of Xiang and Zheng's protocol [6]. To resolve the security drawbacks of their protocol, we propose a secure and lightweight authentication protocol for IoT-based smart homes.
- We demonstrate that our protocol is secure against various kinds of known attacks by reporting on an informal security analysis.
- We conducted formal analysis using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [7–9], Burrows–Abadi–Needham (BAN) logic [10], and the real or random (ROR) model [11]. With the formal analysis, we proved secure mutual authentication, the session key security, and the resistance against MITM and replay attacks of our protocol.
- We provide a comparison of performance and security properties between our protocol and related protocols. The results show that our protocol provides better security and computational costs compared to related protocols.

1.2. Adversary Model

We adopted the widely-used Dolev–Yao (DY) threat model [12–14] and the Canetti and Krawczyk (CK) adversary threat model [15,16] to evaluate the security of the proposed protocol. The capabilities of an adversary $\mathcal{A}$ can be defined as follows.

- $\mathcal{A}$ can eavesdrop, intercept, inject, replay, and modify transmitted messages via a public channel and then $\mathcal{A}$ can perform MITM, replay, impersonation attacks, etc. [17].
- $\mathcal{A}$ can steal the legal user's mobile device or smart device and extract secret credentials stored in the memory by performing the power analysis attack [18–21].
- $\mathcal{A}$ can access short-term keys, long-term keys, and session states of each party.

In addition, we developed some assumptions for our protocol. $\mathcal{A}$ cannot feasibly guess the identity and password of the mobile user simultaneously [22–24]. $\mathcal{A}$ cannot extract the data stored in the home gateway's database, since the home gateway has a secure database.

1.3. Organization

The remaining parts of this paper are structured as follows. In Section 2, we briefly discuss existing proposed protocols in IoT-based smart homes. We suggest the system model of the proposed protocol in Section 3. We review Xiang and Zheng's protocol in Section 4 and analyze security weaknesses of Xiang and Zheng's protocol in Section 5. Section 6 proposes a secure and lightweight authentication protocol for IoT-based smart homes to improve the security drawbacks of Xiang and Zheng's protocol. Section 7 analyzes the security of our protocol through informal and formal analyses with BAN logic, the ROR model, and the AVISPA tool. In Section 8, we present the results of performance and security property comparisons between the proposed protocol and related protocols. Finally, we present the conclusion in Section 9.

2. Related Works

In the last few years, many researchers proposed authentication protocols to provide secure communication between users and smart devices in smart home environments. Santoso and Vun [25] proposed a secure authentication protocol using elliptic curve cryptography (ECC) in IoT-based smart homes. Several authors [26,27] revealed that Santoso and Vun's protocol [25] is vulnerable to privileged-insider and stolen smart card attacks, and fails to achieve user anonymity and untraceability. Dey and Hossian [28] presented a secure session key establishment protocol for smart home environments using public key cryptosystems. Dey and Hossian [28] proved that their protocol achieves resilience against various attacks. Unfortunately, some researchers [29,30] pointed out that Dey and Hossian's protocol [28] has various security drawbacks, such as device compromised and known-key attacks, and is unsuccessful in ensuring anonymity and confidentiality. Shuai et al. [31] suggested an ECC-based anonymous authentication protocol for smart home environments. These protocols [25,28,31] use asymmetric key cryptosystems such as ECC for smart home security. However, in terms of costs, symmetric key cryptosystems are more efficient than asymmetric key cryptosystems for deployment on resource-constrained smart devices.

In view of the computational cost for low capacity devices, many authentication protocols have been proposed using symmetric key cryptosystems in smart home environments. Vaidya et al. [32] proposed a robust authentication protocol to provide secure remote access in home environments using symmetric key cryptosystems. Vaidya et al. [32] claimed that their protocol resists synchronization and stolen smart card attacks, and provides forward secrecy and mutual authentication. However, Kim and Kim [33] demonstrated that Vaidya et al.'s protocol [32] does not resist password guessing and smart card loss attacks, and does not provide forward secrecy. To resolve the security problems in Vaidya et al.'s protocol [32], Kim and Kim [33] proposed an improved authentication protocol. Wazid et al. [34] proposed a symmetric key-based secure remote user authentication protocol to provide future secure communications. Wazid et al. [34] proved that their protocol is secure against other possible known attacks. Lyu et al. [35] pointed out that Wazid et al.'s protocol [34] is not secure against desynchronization and compromised server attacks. Poh et al. [36] proposed a privacy-preserving authentication protocol to support data confidentiality. Unfortunately, Irshad et al. [37] pointed out that Poh et al.'s protocol [36] cannot maintain the privacy of authentication parameters. Although these protocols [32–36] use symmetric key cryptosystems considering the low capacity devices, symmetric key cryptosystems are still unacceptable for smart devices with limited resources in terms of computational costs.

Recently, several lightweight authentication protocols [6,38] have been proposed for smart home environments to solve these problems. Banerjee et al. [38] presented an anonymous and robust authentication protocol for IoT-based smart homes using one-way hash functions, XOR operations, and a fuzzy extractor. Banerjee et al. [38] proved that their protocol resists various attacks. However, AL-Turjman and Deebak [39] pointed out that Banerjee et al.'s protocol [38] does not provide identity protection, traceability, or session secret key agreement. Xiang and Zheng [6] presented a situation-aware protocol for device authentication in smart home environments. Xiang and Zheng [6] claimed that their protocol resists various security threats and ensures data integrity and mutual authentication. However, we prove here that Xiang and Zheng's protocol [6] cannot ensure secure mutual authentication and is vulnerable to stolen smart device, impersonation, and session key disclosure attacks. Therefore, we propose a secure and lightweight authentication protocol for IoT-based smart homes to improve the security flaws of Xiang and Zheng's protocol [6].

3. System Model

Xiang and Zheng [6] claimed that their protocol concentrates on the security of smart grid-enabled smart home environments, but they proposed an authentication protocol that is only for smart home environments. Therefore, we focus on the architecture of general IoT-based smart home environments. The system model is shown in Figure 1.

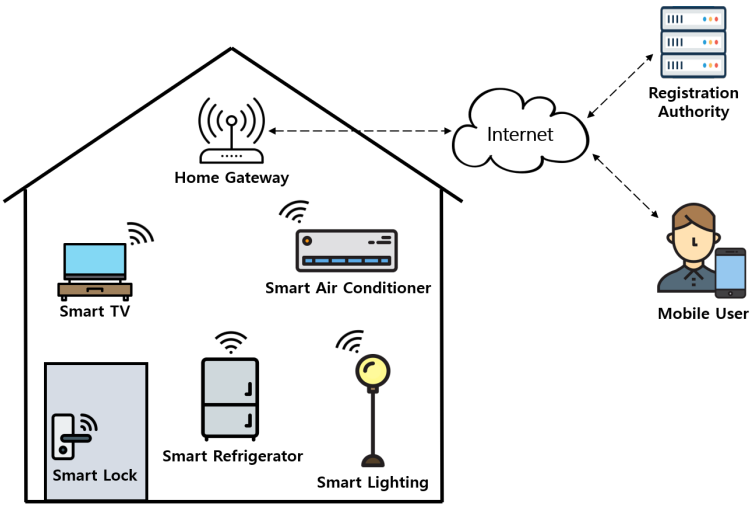

Figure 1. System model for IoT-based smart homes.

The proposed system is composed of a mobile user (*MU*), a smart device (*SD*), a home gateway (*HGW*), and a registration authority (*RA*). *RA* and *HGW* are trusted entities in smart home environments. *RA* is responsible for initializing the system and registering *MU* and *SD*. *MU* first needs to register at *RA* to utilize services. *SD* and *HGW* also need to register at *RA* for providing home services. After receiving the registration request message from *MU* and *SD*, *RA* stores the information of each entity in the mobile device of *MU* and in the memory of *SD*. *RA* also stores all information required for the authentication of the *MU* and *SD* in *HGW*'s database. Then, the *MU* and *SD* perform the mutual authentication and session key agreement with the help of the *HGW*. With this session key, *MU* and *SD* can utilize secure smart home services.

4. Review of Xiang and Zheng's Protocol

This section reviews Xiang and Zheng's protocol [6]. Xiang and Zheng proposed an authentication protocol according to the security risk level in smart home environments. Their protocol consists of smart device registration, and authentication and key agreement phases. The notation of this paper is described in Table 1.

4.1. Smart Device Registration Phase

At the registration phase, RA generates an identity ID_{SD} and a random number r_{RA} for SD and computes $S_i = h(ID_{SD}||r_{RA})$. Then, RA sends $\{ID_{SD}, S_i\}$ to SD and $\{ID_{SD}, r_{RA}\}$ to HGW through a secure channel.

Table 1. Notation.

Notation	Description
MU	Mobile user
HGW	Home gateway
SD	Smart device
RA	Registration authority
ID_{MU}	Identity of MU
ID_G	Identity of HGW
ID_{SD}	Identity of SD
PID_{MU}	Pseudo identity of MU
PID_{SD}	Pseudo identity of SD
PW_{MU}	Password of MU
K_{RA}	Master key of RA
K_{SD}	Secret key of SD
K_{MUG}	Shared secret key between MU and HGW
K_{GSD}	Shared secret key between HGW and SD
$r_{MU}, r_{RA}, r_{SD}, RN_{MU}, RN_G, RN_{SD}$	Random number
SK	Session key between MU and SD
$h(\cdot)$	One-way hash function
$E_K(\cdot)/D_K(\cdot)$	Symmetric encryption/decryption using key K
$\oplus$	XOR operation
$\|\|$	Concatenation operation
T	Timestamp
ΔT	Maximum transmission delay
$HE_{i,L}/HE_{i,H}$	Message header at the low/low security risk

4.2. Authentication and Key Agreement Phase

After the registration, SD sends the message $MSG_1 = [HE_1||ID_{SD}]$ to HGW in the authentication and key agreement phase. $HE_1 =' SD - AUTH'$ is a message header of MSG_1. Upon getting MSG_1, HGW receives the current situation from the smart home system regarding whether the security risk level is low or high. According to the security risk level, the authentication phase is divided into low security risk and high security risk.

4.2.1. Low Security Risk

When HGW receives a low-security-risk level report, the authentication phase is described below.

Step 1: HGW computes $S_i^* = h(ID_{SD}^*||r_{RA})$ and extracts current timestamp T_1. Then HGW calculates $C_{1,L} = (ID_G||T_1) \oplus S_i^*$ and $C_{2,L} = h(HE_{2,L}||ID_G||T_1||S_i^*)$. Finally, HGW sends $MSG_{2,L} = [HE_{2,L}||C_{1,L}||C_{2,L}]$ to SD, where $HE_{2,L} =' HGW - LOW'$ is the header of the message $MSG_{2,L}$ through an insecure channel.

Step 2: Upon receiving the message $MSG_{2,L}$ at timestamp T_1', SD knows the current security risk level is low from the message header. SD also computes $C_{2,L}^* = h(HE_{2,L}^*||ID_G^*||T_1^*||S_i)$ and checks if $|T_1' - T_1^*| \leq \Delta T$ and $C_{2,L}^* \stackrel{?}{=} C_{2,L}$. If it is not equal, the authentication process will be aborted. Then, SD computes $A_i = h(ID_G^*||h(ID_{SD}||S_i))$ and extracts the current timestamp T_2. SD also computes $C_{3,L} = (ID_{SD}||T_2) \oplus A_i$ and $C_{4,L} = h(HE_{3,L}||ID_{SD}||T_2||A_i)$. Finally, SD sends $MSG_{3,L} = [HE_{3,L}||C_{3,L}||C_{4,L}]$ to HGW, where $HE_{3,L} =' SD - LOW'$ is the header of the message $MSG_{3,L}$. SD computes the session key $SK = h(T_1^*||T_2||S_i||A_i)$ for the future data communication.

Step 3: After receiving $MSG_{3,H}$ at timestamp T_2', HGW computes $A_i^* = h(ID_G||h(ID_{SD}||S_i^*))$, $(ID_{SD}^*||T_2^*) = C_{3,L} \oplus A_i^*$, and $C_{4,L}^* = h(HE_{3,L}^*||ID_{SD}^*||T_2^*||A_i^*)$. Then, HGW checks if $|T_2' - T_2^*| \leq \Delta T$ and $C_{4,H}^* \stackrel{?}{=} C_{4,H}$. If it is correct, HGW computes the session key $SK = h(T_1||T_2^*||S_i^*||A_i^*)$ and adds ID_{SD} to the trusted device list.

4.2.2. High Security Risk

If HGW receives a situation report detailing that the current security risk level is high, the authentication phase contains the following steps.

Step 1: HGW computes $S_i^* = h(ID_{SD}^*||r_{RA})$, and generates a random number RN_G. After that, HGW extracts a current timestamp T_1, and computes $C_{1,H} = E_{S_i^*}(ID_G||T_1||RN_G)$ and $C_{2,H} = h(HE_{2,H}||ID_G||T_1||RN_G)$. Then, HGW sends the message $MSG_{2,H} = [HE_{2,H}||C_{1,H}||C_{2,H}]$ to SD, where $HE_{2,H} =' HGW - HIGH'$ is the message header of $MSG_{2,H}$ through a public channel.

Step 2: After getting $MSG_{2,H}$ at timestamp T_1', SD knows the security risk level is high from the header of $MSG_{2,H}$. SD then computes $(ID_G^*||T_1^*||RN_G^*) = D_{S_i}(C_{1,H}^*)$ and $C_{2,H} = h(HE_{2,H}^*||ID_G^*||T_1^*||RN_G^*)$. After that, SD checks whether $|T_1' - T_1^*| \leq \Delta T$ and $C_{2,H}^* \stackrel{?}{=} C_{2,H}$. If the check is failed, the authentication process will be terminated. Otherwise, SD computes $A_i = h(ID_G^*||h(ID_{SD}||S_i))$ and generates a random number RN_{SD}. Then, SD extracts the current timestamp T_2, and computes $C_{3,H} = E_{A_i}(ID_{SD}||T_2||RN_{SD})$ and $C_{4,H} = h(HE_{3,H}||ID_{SD}||T_2||RN_{SD})$. Finally, SD sends the message $MSG_{3,H} = [HE_{3,H}||C_{3,H}||C_{4,H}]$ to HGW, where $HE_{3,H} =' SD - HIGH'$ is the message header of $MSG_{3,H}$, and computes the session key $SK = h(T_1^*||T_2||S_i||A_i||RN_{SD}||RN_G^*)$.

Step 3: Upon receiving $MSG_{3,H}$ at timestamp T_2', HGW computes $A_i^* = h(ID_G||h(ID_{SD}||S_i^*))$, $(ID_{SD}^*||T_2^*||RN_{SD}^*) = D_{A_i^*}(C_{3,H})$, and $C_{4,H}^* = h(HE_{3,H}^*||ID_{SD}^*||T_2^*||RN_{SD}^*)$. Then, HGW checks whether $|T_2' - T_2^*| \leq \Delta T$ and $C_{4,H}^* \stackrel{?}{=} C_{4,H}$. If it is correct, HGW computes the session key $SK = h(T_1||T_2^*||S_i^*||A_i^*||RN_{SD}^*||RN_G)$ and adds ID_{SD} to the trusted device list.

5. Cryptanalysis of Xiang and Zheng's Protocol

In this section, we discuss the security flaws of Xiang and Zheng's protocol. We demonstrate that their protocol is vulnerable to various attacks and does not perform secure mutual authentication.

5.1. Stolen Smart Device Attack

We suppose that an adversary $\mathcal{A}$ can obtain secret credentials $\{ID_{SD}, S_i\}$ of SD using the power analysis according to Section 1.2. Xiang and Zheng's protocol sends the authentication request message $MSG_1 = [HE_1||ID_{SD}]$ as plaintext. $\mathcal{A}$ can obtain HE_1 from $[HE_1||ID_{SD}]$ of the previous session. Then, $\mathcal{A}$ can make the message MSG_1 anytime and perform various attacks with secret credentials. In conclusion, their protocol does not prevent the stolen smart device attack.

5.2. Impersonation Attack

According to Section 1.2, $\mathcal{A}$ can perform an impersonation attack at low and low-security-risk levels. The detailed processes are below.

5.2.1. Low Security Risk

$\mathcal{A}$ can perform the impersonation attack with the following steps.

Step 1: With the obtained secret credentials $\{ID_{SD}, S_i\}$ from SD and HE_1 from the previous session, $\mathcal{A}$ can send the message $MSG_1 = [HE_1||ID_{SD}]$.

Step 2: Upon getting MSG_1, HGW computes $S_i^* = h(ID_{SD}^*||r_{RA})$ and extracts the current timestamp T_1. After that, HGW computes $C_{1,L} = (ID_G||T_1) \oplus S_i^*$ and $C_{2,L} = h(HE_{2,L}||ID_G||T_1||S_i^*)$, and sends the message $MSG_{2,L} = [HE_{2,L}||C_{1,L}||C_{2,L}]$.

Step 3: After receiving $MSG_{2,L}$, $\mathcal{A}$ computes $(ID_G^*||T_1^*) = C_{1,L} \oplus S_i$ and $C_{2,L}^* = h(HE_{2,L}^*||ID_G^*||T_1^*||S_i)$. Then, $\mathcal{A}$ verifies the validity of T_1^* and $C_{2,L}^*$. If it is equal, $\mathcal{A}$ computes $A_i = h(ID_G^*||h(ID_{SD}||S_i))$ and generates the current timestamp T_2. After that, $\mathcal{A}$ computes $C_{3,L} = (ID_{SD}||T_2) \oplus A_i$ and $C_{4,L} = h(HE_{3,L}||ID_{SD}||T_2||A_i)$. Finally, $\mathcal{A}$ sends the message $MSG_{3,L} = [HE_{3,L}||C_{3,L}||C_{4,L}]$ to HGW and computes the session key $SK = h(T_1^*||T_2||S_i||A_i)$.

Step 4: Upon getting $MSG_{3,L}$, HGW computes $A_i^* = h(ID_G||h(ID_{SD}||S_i^*))$, $(ID_{SD}^*||T_2^*) = C_{3,L} \oplus A_i^*$, and $C_{4,L}^* = h(HE_{3,L}^*||ID_{SD}^*||T_2^*||A_i^*)$. After that, HGW checks the validity of T_2^* and $C_{4,L}^*$. If it is equal, HGW computes $SK = h(T_1||T_2^*||S_i^*||A_i^*)$.

Thus, $\mathcal{A}$ can impersonate SD successfully, and Xiang and Zheng's protocol cannot prevent the impersonation attack at the low-security-risk level.

5.2.2. High Security Risk

With the obtained secret credentials $\{ID_{SD}, S_i\}$, $\mathcal{A}$ can disguise as SD, and the detailed steps are below.

Step 1: $\mathcal{A}$ can send $MSG_1 = [HE_1||ID_{SD}]$ to HGW using obtained secret credentials $\{ID_{SD}, S_i\}$ and HE_1.

Step 2: Upon getting MSG_1, HGW calculates $S_i^* = h(ID_{SD}^*||r_{RA})$ and generates a random number RN_G. After that, HGW extracts the current timestamp T_1, and computes $C_{1,H} = E_{S_i^*}(ID_G||T_1||RN_G)$ and $C_{2,H} = h(HE_{2,H}||ID_G||T_1||RN_G)$. Then, HGW sends $MSG_{2,H} = [HE_{2,H}||C_{1,H}||C_{2,H}]$.

Step 3: After receiving $MSG_{2,H}$, $\mathcal{A}$ computes $(ID_G^*||T_1^*||RN_G) = D_{S_i}(C_{1,H}^*)$ and $C_{2,H}^* = h(HE_{2,H}^*||ID_G^*||T_1^*||RN_G^*)$. Then, $\mathcal{A}$ verifies the validity of T_1^* and $C_{2,H}^*$. If all checks pass, $\mathcal{A}$ computes $A_i^* = h(ID_G^*||h(ID_{SD}||S_i))$, generates a random number RN_{SD}, and extracts the current timestamp T_2. After that, $\mathcal{A}$ computes $C_{3,H} = E_{A_i}(ID_{SD}||T_2||RN_{SD})$, $C_{4,H} = h(HE_{3,H}||ID_{SD}||T_2||RN_{SD})$, and $SK = h(T_1^*||T_2||S_i||A_i||RN_{SD}||RN_G^*)$. Finally, $\mathcal{A}$ sends $MSG_{3,H} = [HE_{3,H}||C_{3,H}||C_{4,H}]$ to HGW.

Step 4: Upon getting $MSG_{3,H}$, HGW computes $A_i^* = h(ID_G||h(ID_{SD}||S_i^*))$, $(ID_{SD}^*||T_2^*||RN_{SD}^*) = D_{A_i^*}(C_{3,H})$, and $C_{4,H}^* = h(HE_{3,H}^*||ID_{SD}^*||T_2^*||RN_{SD}^*)$. Then, HGW checks the validity of T_2^* and $C_{4,H}^*$. If it is equal, HGW computes $SK = h(T_1||T_2^*||S_i^*||A_i^*||RN_{SD}^*||RN_G)$.

In conclusion, Xiang and Zheng's protocol cannot prevent the impersonation attack at the low-security-risk level because $\mathcal{A}$ can impersonate SD successfully.

5.3. Session Key Disclosure Attack

As mentioned in Section 1.2, $\mathcal{A}$ can extract secret credentials $\{ID_{SD}, S_i\}$. In addition, according to Section 5.2, $\mathcal{A}$ can obtain the session key between SD and HGW at the both low-security-risk and high-security-risk levels. With the obtained session key, $\mathcal{A}$ can communicate with HGW and misinform HGW for $\mathcal{A}$'s own purpose. Therefore, Xiang and Zheng's protocol is vulnerable to the session key disclosure attack.

5.4. Mutual Authentication

Xiang and Zheng claimed that their protocol supports the mutual authentication between SD and HGW because S_i and A_i cannot be obtained from the eavesdropped messages. However, in accordance with Section 5.2, $\mathcal{A}$ can generate an authentication request message $MSG_1 = [HE_1||ID_{SD}]$ and calculate session key $SK = h(T_1||T_2||S_i||A_i)$ and $SK = h(T_1||T_2||S_i||A_i||RN_{SD}||RN_G)$ at low security and low security phases, respectively. Thus, Xiang and Zheng's protocol does not satisfy secure mutual authentication between SD and HGW.

6. Proposed Protocol

In this section, we present a secure and lightweight authentication protocol for IoT-based smart homes to improve the security drawbacks of Xiang and Zheng's protocol [6]. The proposed protocol consists of four phases: initialization, registration, authentication and key agreement, and password update.

6.1. Initialization Phase

Before SD and HGW are deployed in the smart home, RA generates a master key K_{RA}. HGW has a unique identity ID_G, and SD has a unique identity ID_{SD} and secret key K_{SD}.

6.2. Registration Phase

The detailed registration phases for the smart device and user are below.

6.2.1. Smart Device Registration Phase

To provide home services to MU, SD must register at RA. We indicate the registration phase of SD and RA in Figure 2, and detailed steps are described below.

Step 1: SD generates a random number r_{SD} and computes $PID_{SD} = h(ID_{SD}||r_{SD})$. Then, SD sends $\{PID_{SD}, r_{SD}\}$ to RA through a secure channel.

Step 2: Upon getting the message, RA generates r_{RA} and computes $K_{GSD} = h(PID_{SD}||K_{RA}||r_{RA})$. Then, RA stores $\{PID_{SD}, K_{GSD}, r_{SD}\}$ in HGW's database and sends $\{K_{GSD}\}$ to SD over a secure channel. After that, RA makes PID_{SD} public.

Step 3: After receiving the message, SD computes $B_1 = r_{SD} \oplus h(ID_{SD}||K_{SD})$ and $B_2 = K_{GSD} \oplus h(r_{SD}||K_{SD})$. Then, SD stores $\{B_1, B_2, PID_{SD}\}$ in the memory.

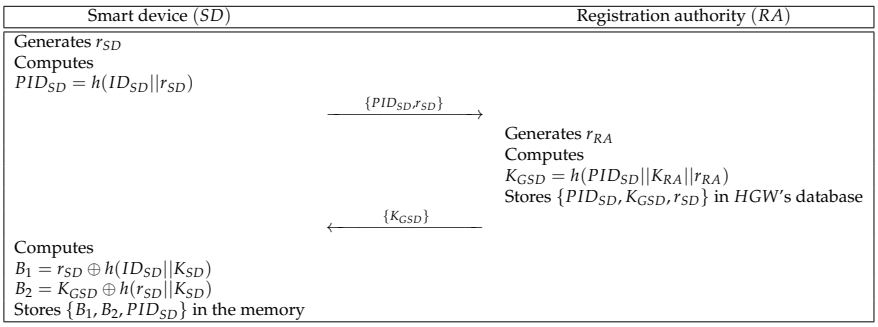

Figure 2. Smart device registration phase of the proposed protocol.

6.2.2. Mobile User Registration Phase

MU must register at RA to use the data transmitted from SD. Figure 3 shows the registration phase of MU and RA. This phase is described as follows.

Step 1: MU selects identity and password $\{ID_{MU}, PW_{MU}\}$ and generates a random number r_{MU}. Then, MU computes $PID_{MU} = h(ID_{MU}||r_{MU})$ and sends $\{PID_{MU}\}$ to RA through a secure channel.

Step 2: Upon receiving the message, RA computes $K_{MUG} = h(PID_{MU}||K_{RA}||r_{RA})$ and $RID_{MU} = h(PID_{MU}||K_{MUG})$. Then, RA stores $\{PID_{MU}, RID_{MU}, K_{MUG}\}$ in HGW's database and sends $\{K_{MUG}, RID_{MU}\}$ to MU via a secure channel.

Step 3: After receiving the message, MU computes $HPW_{MU} = h(PW_{MU}||r_{MU})$, $A_1 = r_{MU} \oplus h(ID_{MU}||PW_{MU})$, $A_2 = h(ID_{MU}||PW_{MU}||r_{MU}||HPW_{MU})$, $A_3 = RID_{MU} \oplus h(r_{MU}||HPW_{MU})$, and $A_4 = K_{MUG} \oplus h(RID_{MU}||HPW_{MU})$. Then, MU stores $\{A_1, A_2, A_3, A_4, PID_{MU}\}$ in the mobile device.

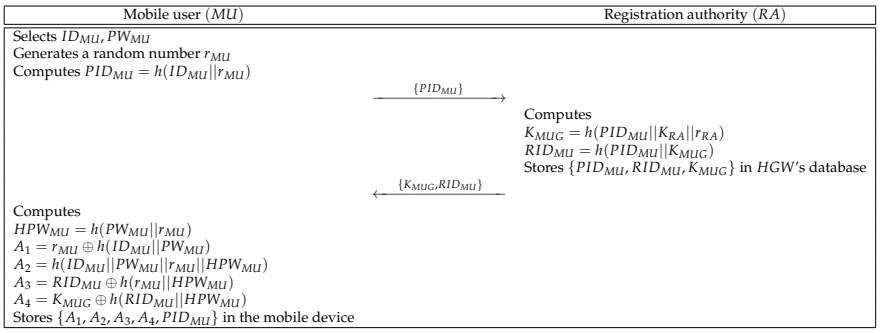

Figure 3. Mobile user registration phase of the proposed protocol.

6.3. Authentication and Key Agreement Phase

To utilize secure home services, MU and SD establish a session key with the help of HGW. We indicate the detailed steps below, and a summarized version of this phase is in Figure 4.

Step 1: MU inputs identity and password $\{ID_{MU}, PW_{MU}\}$ and computes $r_{MU} = A_1 \oplus h(ID_{MU}||PW_{MU})$, $HPW_{MU} = h(PW_{MU}||r_{MU})$, and $A_2^* = h(ID_{MU}||PW_{MU}||r_{MU}||HPW_{MU})$. Then, MU checks if $A_2^* \stackrel{?}{=} A_2$. If this condition is satisfied, MU generates a random nonce RN_{MU} and computes $RID_{MU} = A_3 \oplus h(r_{MU}||HPW_{MU})$, $K_{MUG} = A_4 \oplus h(RID_{MU}||HPW_{MU})$, $M_1 = h(PID_{MU}||RID_{MU}||K_{MUG}) \oplus (RN_{MU}||PID_{SD})$, $C_1 = h(ID_{MU}||RN_{MU}) \oplus h(K_{MUG}||RN_{MU})$, and $V_{MU} = h(PID_{MU}||RID_{MU}||RN_{MU}||PID_{SD}||K_{MUG})$. After that, MU sends $\{PID_{MU}, M_1, C_1, V_{MU}\}$ to HGW through a public channel.

Step 2: Upon getting the message, HGW retrieves RID_{MU} and K_{MUG} corresponding to PID_{MU}, and computes $(RN_{MU}^*||PID_{SD}^*) = M_1 \oplus h(PID_{MU}||RID_{MU}||K_{MUG})$ and $V_{MU}^* = h(PID_{MU}||RID_{MU}||RN_{MU}^*||PID_{SD}^*||K_{MUG})$. HGW checks if $V_{MU}^* \stackrel{?}{=} V_{MU}$. If it is equal, HGW retrieves K_{GSD} and r_{SD} corresponding to PID_{SD}. Then, HGW generates a random nonce RN_G and computes $M_2 = h(RN_{MU}||RN_G)$, $M_3 = h(PID_{SD}||K_{GSD}||r_{SD}) \oplus M_2$, $h(ID_{MU}||RN_{MU}) = C_1 \oplus h(K_{MUG}||RN_{MU})$, $C_2 = (h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)) \oplus h(K_{GSD}||r_{SD})$, and $V_{MUG} = h(PID_{MU}||M_2||K_{GSD})$. Finally, HGW sends $\{PID_{MU}, M_3, C_2, V_{MUG}\}$ to SD.

Step 3: After receiving the message, SD computes $r_{SD} = B_1 \oplus h(ID_{SD}||K_{SD})$, $K_{GSD} = B_2 \oplus h(r_{SD}||K_{SD})$, $M_2^* = M_3 \oplus h(PID_{SD}||K_{GSD}||r_{SD})$, and $V_{MUG}^* = h(PID_{MU}||M_2^*||K_{GSD})$. SD checks if $V_{MUG}^* \stackrel{?}{=} V_{MUG}$. If this condition is valid, SD generates a random nonce RN_{SD}. Then, SD computes $(h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)) = C_2 \oplus h(K_{GSD}||r_{SD})$, $SK = h(h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)||h(ID_{SD}||RN_{SD}))$, $M_4 = h(PID_{SD}||K_{GSD}||r_{SD}) \oplus h(ID_{SD}||RN_{SD})$, and $V_{SD} = h(PID_{MU}||PID_{SD}||M_2^*||h(ID_{SD}||RN_{SD})||K_{GSD})$. Finally, SD sends $\{M_4, V_{SD}\}$ to HGW.

Step 4: Upon receiving the message, HGW computes $h(ID_{SD}||RN_{SD}) = M_4 \oplus h(PID_{SD}||K_{GSD}||r_{SD})$ and $V_{SD}^* = h(PID_{MU}||PID_{SD}||M_2||h(ID_{SD}||RN_{SD})||K_{GSD})$. HGW checks if $V_{SD}^* \stackrel{?}{=} V_{SD}$. Then, HGW computes $SK = h(h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)||ID_{SD}||RN_{SD}))$, $PID_{MU}^{new} = h(PID_{MU}||RN_{MU})$, and $RID_{MU}^{new} = h(PID_{MU}^{new}||K_{MUG})$, and computes $M_5 = h(RID_{MU}||RN_{MU}) \oplus (h(ID_G||RN_G)||h(ID_{SD}||RN_{SD})||PID_{MU}^{new})$ and $V_{GSD} = h(PID_{MU}||RN_{MU}||h(ID_G||RN_G)||h(ID_{SD}||RN_{SD})||PID_{MU}^{new}||K_{MUG})$. HGW stores $\{PID_{MU}, RID_{MU}\}$ with $\{PID_{MU}^{new}, RID_{MU}^{new}\}$ in HGW's database. Finally, HGW sends $\{M_5, V_{GSD}\}$ to MU.

Step 5: After receiving the message, MU computes $PID_{MU}^{new} = h(PID_{MU}||RN_{MU})$, $(h(ID_G||RN_G)||h(ID_{SD}||RN_{SD})||PID_{MU}^{new}) = M_5 \oplus h(RID_{MU}||RN_{MU})$ and $V_{GSD}^* = h(PID_{MU}||RN_{MU}||h(ID_G||RN_G)||h(ID_{SD}||RN_{SD})||PID_{MU}^{new}||K_{MUG})$. MU checks if $V_{GSD}^* \stackrel{?}{=} V_{GSD}$. After that, MU computes $SK = h(h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)||h(ID_{SD}||$

RN_{SD})). Then, MU updates $RID_{MU}^{new} = h(PID_{MU}^{new}||K_{MUG})$, $A_3^{new} = RID_{MU}^{new} \oplus h(r_{MU}||HPW_{MU})$, and $A_4^{new} = K_{MUG} \oplus h(RID_{MU}^{new}||HPW_{MU})$. Then, MU replaces $\{A_3, A_4, PID_{MU}\}$ to $\{A_3^{new}, A_4^{new}, PID_{MU}^{new}\}$ in the mobile device. MU computes $M_6 = h(SK||PID_{MU}^{new})$ and sends M_6 to HGW.

Step 6: After receiving the message from MU, HGW computes $M_6^* = h(SK||PID_{MU}^{new})$ and checks if $M_6^* \stackrel{?}{=} M_6$. If it is correct, HGW deletes $\{PID_{MU}, RID_{MU}\}$ in the database.

Figure 4. Authentication and key agreement phase of the proposed protocol.

6.4. Password Update Phase

MU can update the password individually. In Figure 5, we represent the password update phase and the detailed steps are below.

Step 1: MU inputs identity and old password $\{ID_{MU}, PW_{MU}^{old}\}$ to the mobile device over a secure channel.

Step 2: Mobile device computes $r_{MU} = A_1 \oplus h(ID_{MU}||PW_{MU}^{old})$, $HPW_{MU} = h(PW_{MU}^{old}||r_{MU})$, and $A_2^* = h(ID_{MU}||PW_{MU}^{old}||r_{MU}||HPW_{MU})$. Then, the mobile device checks whether $A_2^* \stackrel{?}{=} A_2$. If this condition is met, the mobile device sends the authentication message to MU.

Step 3: Upon receiving the authentication message, MU inputs the new password PW_{MU}^{new} to the mobile device.

Step 4: After getting the new password, the mobile device computes $RID_{MU} = A_3 \oplus h(r_{MU}||HPW_{MU})$, $K_{MUG} = A_4 \oplus h(RID_{MU}||HPW_{MU})$, $HPW_{MU}^{**} = h(PW_{MU}^{new}||r_{MU})$,

$A_1^{**} = r_{MU} \oplus h(ID_{MU}||PW_{MU}^{new})$, $A_2^{**} = h(ID_{MU}||PW_{MU}^{new}||r_{MU}||HPW_{MU}^{**})$, $A_3^{**} = RID_{MU} \oplus h(r_{MU}||HPW_{MU}^{**})$, and $A_4^{**} = K_{MUG} \oplus h(RID_{MU}||HPW_{MU}^{**})$. Finally, the mobile device replaces $\{A_1, A_2, A_3, A_4, PID_{MU}\}$ with $\{A_1^{**}, A_2^{**}, A_3^{**}, A_4^{**}, PID_{MU}\}$.

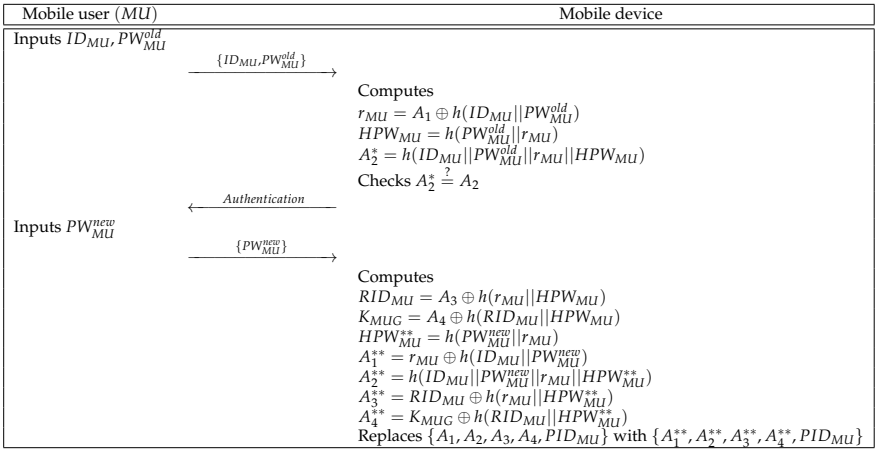

Figure 5. Password update phase of the proposed protocol.

7. Security Analysis

This section shows informal and formal security analyses of our protocol using BAN logic, the ROR model, and the AVISPA tool. Through theses analyses, we demonstrate that the proposed protocol prevents various kinds of known attacks.

7.1. Informal Security Analysis

We performed informal analysis to describe how our protocol withstands various attacks and supports perfect forward secrecy and mutual authentication.

7.1.1. Mobile User Impersonation Attack

According to Section 1.2, an adversary $\mathcal{A}$ can have the lost/stolen mobile device of a legal user MU, and extract secret credentials $\{A_1, A_2, A_3, A_4, PID_{MU}\}$ using the power analysis [18,19]. With these values, $\mathcal{A}$ can try to impersonate MU by intercepting transmitted messages through an insecure channel. However, $\mathcal{A}$ cannot send a valid authentication request message $\{M_1, C_1, V_{MU}\}$ because $\mathcal{A}$ cannot calculate $\{HPW_{MU}, RID_{MU}, K_{MUG}\}$ without the knowledge of the MU's real identity ID_{MU}, password PW_{MU}, and a random nonce RN_{MU}. Hence, the proposed protocol resists the mobile user impersonation attack.

7.1.2. Home Gateway Impersonation Attack

Suppose that an adversary $\mathcal{A}$ intercepts messages $\{PID_{MU}, M_3, C_2, V_{MUG}\}$ and $\{M_5, V_{GSD}\}$ over an insecure channel. $\mathcal{A}$ can try to calculate the other valid messages $\{PID_{MU}, M_3', C_2', V_{MUG}'\}$ and $\{M_5', V_{GSD}'\}$. However, $\mathcal{A}$ cannot compute messages, because $\mathcal{A}$ has no knowledge of the MU's real identity ID_{MU} and a random nonce RN_{MU}. In addition, $\mathcal{A}$ does not know HGW's real identity ID_G, a random nonce RN_G, and the shared secret key K_{GSD}. Thus, the proposed protocol withstands the home gateway impersonation attack.

7.1.3. Smart Device Impersonation Attack

An adversary $\mathcal{A}$ can try to impersonate SD using the exchanged message $\{M_4, V_{SD}\}$. According to Section 1.2, $\mathcal{A}$ can extract stored values in the lost/stolen smart device. However, $\mathcal{A}$ cannot compute the message because $\mathcal{A}$ does not know the SD's unique identity ID_{SD}, secret key K_{SD}, and a random nonce RN_{SD}. Therefore, our protocol prevents the smart device impersonation attack.

7.1.4. Session Key Disclosure Attack

In accordance with Section 1.2, an adversary $\mathcal{A}$ can extract secret credentials $\{A_1, A_2, A_3, A_4, PID_{MU}\}$ and $\{B_1, B_2, PID_{SD}\}$ of MU and SD, respectively. To calculate the session key, $\mathcal{A}$ should know real identities and random nonces of MU, HGW, and SD. However, $\mathcal{A}$ cannot obtain $\{ID_{MU}, ID_G, ID_{SD}\}$ and $\{RN_{MU}, RN_G, RN_{SD}\}$ from transmitted messages because these are encrypted with secret keys $\{K_{MUG}, K_{GSD}, K_{SD}\}$. Thus, the proposed protocol withstands the session key disclosure attack.

7.1.5. Replay and MITM Attack

We assume that an adversary $\mathcal{A}$ intercepts and resends the previous authentication request message $\{PID_{MU}, M_1, C_1, V_{MU}\}$ to HGW for the purpose of disguising MU. HGW detects RN_{MU} is not fresh by checking the validity of V_{MU}. In addition, even if $\mathcal{A}$ tries to modify the authentication request message, $\mathcal{A}$ cannot modify $\{M_1, C_1, V_{MU}\}$ without the knowledge of the MU's real identity ID_{MU}, password PW_{MU}, a random nonce RN_{MU}, and shared secret key K_{MUG}. In conclusion, our protocol prevents replay and MITM attacks.

7.1.6. Offline Guessing Attack

After extracting the information from the MU's mobile device, $\mathcal{A}$ can obtain $A_1 = r_{MU} \oplus h(ID_{MU}||PW_{MU})$, $A_2 = h(ID_{MU}||PW_{MU}||r_{MU}||HPW_{MU})$, $A_3 = RID_{MU} \oplus h(r_{MU}||HPW_{MU})$, and $A_4 = K_{MUG} \oplus h(RID_{MU}||HPW_{MU})$. All of these values are encrypted with ID_{MU} and PW_{MU}. If $\mathcal{A}$ wants to compromise the security of our protocol, $\mathcal{A}$ needs to guess both ID_{MU} and PW_{MU}. However, it is a computationally infeasible problem to $\mathcal{A}$ according to Section 1.2. As a result, our protocol resists the offline guessing attack.

7.1.7. Stolen Smart Device Attack

Assume that an adversary $\mathcal{A}$ obtains SD and extracts secret credentials $\{B_1, B_2, PID_{SD}\}$ stored in the memory through the power analysis attack [20,21]. Although $\mathcal{A}$ obtains these values, $\mathcal{A}$ cannot get sensitive information of SD because all information stored in the memory is masked with SD's unique identity ID_{SD} and secret key K_{SD}. Thus, the proposed protocol withstands the stolen smart device attack.

7.1.8. Privileged-Insider Attack

In this attack, a privileged-insider adversary $\mathcal{A}$ is able to get PID_{MU} during the MU's registration phase. Then, $\mathcal{A}$ can extract secret credentials $\{A_1, A_2, A_3, A_4, PID_{MU}\}$ stored in the mobile device. However, since $\mathcal{A}$ does not know the MU's real identity ID_{MU}, password PW_{MU}, and a random number r_{MU}, $\mathcal{A}$ cannot calculate the session key $SK = h(h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)||h(ID_{SD}||RN_{SD}))$. Hence, our protocol prevents the privileged-insider attack.

7.1.9. Known Session-Secret Temporary Information Attack

An adversary $\mathcal{A}$ can obtain session specific random nonces $\{RN_{MU}, RN_G, RN_{SD}\}$ to conduct the known session-secret temporary information attack under the CK-adversary model. Even if $\mathcal{A}$ knows these secrets, $\mathcal{A}$ cannot calculate the session key $SK = h(h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)||h(ID_{SD}||RN_{SD}))$, because SK consists of MU, HGW, and SD's identities. Thus, our protocol withstands the known session-secret temporary information attack.

7.1.10. Desynchronization Attack

A desynchronization attack is when an adversary $\mathcal{A}$ can modify and block the transmitted messages to make MU, HGW, and SD unable to authenticate in the future. Assume that $\mathcal{A}$ tries to modify the messages for desynchronizing the next session. However, as mentioned in Section 7.1.5, $\mathcal{A}$ cannot modify the exchanged messages because $\mathcal{A}$ has no knowledge about MU's secret credentials. In addition, we assume that $\mathcal{A}$ blocks the transmitted messages to disturb the synchronization. HGW calculates PID_{MU}^{new}, generates

a verification message $\{M_5, V_{GSD}\}$ using PID_{MU}^{new}, and sends it to MU. HGW stores the PID_{MU}^{new} with PID_{MU}, and MU checks V_{GSD}. If the V_{GSD} is correct, MU updates PID_{MU}^{new}. MU sends the message M_6 to HGW to describe that authentication is complete. Then, HGW checks the validation of M_6. If M_6 is validated, HGW deletes the old PID_{MU} and RID_{MU}. Otherwise, HGW stores them. Through these things, MU and HGW always have synchronized values. Consequently, a desynchronization attack is impossible in our protocol.

7.1.11. Perfect Forward Secrecy

We assume that an adversary $\mathcal{A}$ knows long-term secret keys $\{K_{RA}, K_{MUG}, K_{GSD}, K_{SD}\}$. $\mathcal{A}$ can try to calculate the session key $SK = h(h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)||h(ID_{SD}||RN_{SD}))$. However, $\mathcal{A}$ cannot affect on the confidentiality of past communications because SK is composed of the random nonces $\{RN_{MU}, RN_G, RN_{SD}\}$ which is generated for each session. Thus, the proposed protocol provides the perfect forward secrecy.

7.1.12. Mutual Authentication

At the authentication and key agreement phase, MU, HGW, and SD check the message validity. MU checks the validity of V_{GSD}^*, HGW verifies $V_{MU}^* \stackrel{?}{=} V_{MU}$ and $V_{SD}^* \stackrel{?}{=} V_{SD}$, and SD checks whether $V_{MUG}^* \stackrel{?}{=} V_{MUG}$. If the values are correct, each entity authenticates each other. Therefore, our protocol achieves the mutual authentication.

7.1.13. Anonymity and Untraceability

An adversary $\mathcal{A}$ can obtain exchanged messages in the authentication and key agreement phase. However, $\mathcal{A}$ cannot obtain real identities of MU, HGW, and SD because these are dependent on $\{r_{MU}, RN_G, r_{SD}\}$. In addition, MU and HGW update PID_{MU} to $PID_{MU}^{new} = h(PID_{MU}||RN_{MU})$ for every session. It makes all messages are dynamic at every session. Consequently, the proposed protocol provides anonymity and untraceability.

7.2. BAN Logic

We performed the formal security analysis with BAN logic to evaluate the secure mutual authentication of the proposed protocol [10,40]. We present the notation of BAN logic in Table 2.

Table 2. BAN logic notation.

Notation	Description
$skey$	Secret key
$W\| \equiv S$	W **believes** statement S
$\#S$	Statement S is **fresh**
$W \triangleleft S$	W **receives** statement S
$W\| \sim S$	W once **said** S
$W \Rightarrow S$	W **controls** statement S
$< S >_T$	Statement S is **combined** with secret statement T
$\{S\}_{skey}$	Statement S is **masked** by $skey$
$W \stackrel{skey}{\longleftrightarrow} N$	W and N **share** $skey$ to communicate with each other
$W \stackrel{skey}{\rightleftharpoons} N$	$skey$ is known only to W, N, and trusted principals of W and N

7.2.1. Rules

We describe the rules of BAN logic in the following.

- Message meaning rule (MMR):

$$\frac{W| \equiv W \stackrel{skey}{\longleftrightarrow} N, W \triangleleft \{S\}_{skey}}{W| \equiv N| \sim S}$$

- Nonce verification rule (NVR):

$$\frac{W| \equiv \#(S), W| \equiv N| \sim S}{W| \equiv N| \equiv S}$$

- Jurisdiction rule (JR):

$$\frac{W| \equiv N| \Rightarrow S, W| \equiv N| \equiv S}{W| \equiv S}$$

- Freshness rule (FR):

$$\frac{W| \equiv \#(S)}{W| \equiv \#(S,T)}$$

- Belief rule (BR):

$$\frac{W| \equiv (S,T)}{W| \equiv S}$$

7.2.2. Goals

The following are the main goals to demonstrate that our protocol satisfies the secure mutual authentication.

Goal 1: $MU| \equiv (MU \overset{SK}{\longleftrightarrow} SD)$.

Goal 2: $MU| \equiv SD| \equiv (MU \overset{SK}{\longleftrightarrow} SD)$.

Goal 3: $SD| \equiv (MU \overset{SK}{\longleftrightarrow} SD)$.

Goal 4: $SD| \equiv MU| \equiv (MU \overset{SK}{\longleftrightarrow} SD)$.

7.2.3. Assumptions

We assume the following to initiate states of the proposed protocol.

A_1**:** $HGW| \equiv (MU \overset{SK}{\longleftrightarrow} HGW)$

A_2**:** $HGW| \equiv \#(RN_{MU})$

A_3**:** $SD| \equiv (HGW \overset{K_{GSD}}{\longleftrightarrow} SD)$

A_4**:** $SD| \equiv \#(RN_G)$

A_5**:** $HGW| \equiv (HGW \overset{K_{GSD}}{\longleftrightarrow} SD)$

A_6**:** $HGW| \equiv \#(RN_{SD})$

A_7**:** $MU| \equiv (MU \overset{K_{MUG}}{\longleftrightarrow} HGW)$

A_8**:** $MU| \equiv \#(RN_G)$

A_9**:** $MU| \equiv HGW| \Rightarrow MU \overset{h(ID_G||RN_G)||h(ID_{SD}||RN_{SD})}{\rightleftharpoons} SD$

A_{10}**:** $SD| \equiv HGW| \Rightarrow (MU \overset{h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)}{\rightleftharpoons} SD)$

A_{11}**:** $MU| \equiv SD| \Rightarrow (MU \overset{SK}{\longleftrightarrow} SD)$

A_{12}**:** $SD| \equiv MU| \Rightarrow (MU \overset{SK}{\longleftrightarrow} SD)$

7.2.4. Idealized Forms

We present ideal forms of our protocol as below.

M_1**:** $MU \rightarrow HGW : (PID_{MU}, RID_{MU}, RN_{MU})_{K_{MUG}}$

M_2**:** $HGW \rightarrow SD : (PID_{MU}, h(ID_{MU}||RN_{MU}), h(ID_G||RN_G), PID_{SD}, r_{SD})_{K_{GSD}}$

M_3**:** $SD \rightarrow HGW : (PID_{MU}, PID_{SD}, h(ID_{MU}||RN_{MU}), h(ID_{SD}||RN_{SD}))_{K_{GSD}}$

M_4**:** $HGW \rightarrow MU : (RID_{MU}, h(ID_{MU}||RN_{MU}), h(ID_G||RN_G), h(ID_{SD}||RN_{SD}))_{K_{MUG}}$

7.2.5. Proof

We conducted the BAN logic test, and detailed steps are described as follows.

Step 1: From M_1, we can obtain S_1.

$$S_1 : HGW \triangleleft (PID_{MU}, RID_{MU}, RN_{MU})_{K_{MUG}}$$

Step 2: Using S_1 and A_1 with MMR, we can get S_2.

$$S_2 : HGW| \equiv MU| \sim (PID_{MU}, RID_{MU}, RN_{MU})_{K_{MUG}}$$

Step 3: S_3 can obtained using S_2 and A_2 with FR.

$$S_3 : HGW| \equiv \#(PID_{MU}, RID_{MU}, RN_{MU})_{K_{MUG}}$$

Step 4: Using S_2 and S_3 with NVR, we can get S_4.

$$S_4 : HGW| \equiv MU| \equiv (PID_{MU}, RID_{MU}, RN_{MU})_{K_{MUG}}$$

Step 5: We can obtain S_5 from M_2.

$$S_5 : SD \triangleleft (PID_{MU}, h(ID_{MU}||RN_{MU}), h(ID_G||RN_G), PID_{SD}, r_{SD})$$

Step 6: S_6 can obtained using S_5 and A_3 with MMR.

$$S_6 : SD| \equiv HGW| \sim (PID_{MU}, h(ID_{MU}||RN_{MU}), h(ID_G||RN_G), PID_{SD}, r_{SD})_{K_{GSD}}$$

Step 7: Utilizing S_6 and A_4 with FR, we can get S_7.

$$S_7 : SD| \equiv \#(PID_{MU}, h(ID_{MU}||RN_{MU}), h(ID_G||RN_G), PID_{SD}, r_{SD})_{K_{GSD}}$$

Step 8: For obtaining S_8, we can use S_6 and S_7 with NVR.

$$S_8 : SD| \equiv HGW| \equiv (PID_{MU}, h(ID_{MU}||RN_{MU}), h(ID_G||RN_G), PID_{SD}, r_{SD})_{K_{GSD}}$$

Step 9: From M_3, we can obtain S_9.

$$S_9 : HGW \triangleleft (PID_{MU}, PID_{SD}, h(ID_{MU}||RN_{MU}), h(ID_{SD}||RN_{SD}))_{K_{GSD}}$$

Step 10: For getting S_{10}, we can utilize S_9 and A_5 with MMR.

$$S_{10} : HGW| \equiv SD| \sim (PID_{MU}, PID_{SD}, h(ID_{MU}||RN_{MU}), h(ID_{SD}||RN_{SD}))_{K_{GSD}}$$

Step 11: For obtaining S_{11}, we can use A_6 and S_{10} with FR.

$$S_{11} : HGW| \equiv \#(PID_{MU}, PID_{SD}, h(ID_{MU}||RN_{MU}), h(ID_{SD}||RN_{SD}))_{K_{GSD}}$$

Step 12: Using S_{10} and S_{11} with NVR, we can get S_{12}.

$$S_{12} : HGW| \equiv SD| \equiv (PID_{MU}, PID_{SD}, h(ID_{MU}||RN_{MU}), h(ID_{SD}||RN_{SD}))_{K_{GSD}}$$

Step 13: We can get S_{13} from M_4.

$$S_{13} : MU \triangleleft (RID_{MU}, h(ID_{MU}||RN_{MU}), h(ID_G||RN_G), h(ID_{SD}||RN_{SD}))_{K_{MUG}}$$

Step 14: S_{14} can obtained using S_{13} and A_7 with MMR.

$$MU| \equiv HGW| \sim (RID_{MU}, h(ID_{MU}||RN_{MU}), h(ID_G||RN_G), h(ID_{SD}||RN_{SD}))_{K_{MUG}}$$

Step 15: S_{15} can obtained using S_{14} and A_8 with FR.

$$S_{15} : MU| \equiv \#(RID_{MU}, h(ID_{MU}||RN_{MU}), h(ID_G||RN_G), h(ID_{SD}||RN_{SD}))_{K_{MUG}}$$

Step 16: Using S_{14} and S_{15} with NVR, we can get S_{16}.

$$S_{16} : MU| \equiv HGW| \equiv (RID_{MU}, h(ID_{MU}||RN_{MU}), h(ID_G||RN_G), h(ID_{SD}||RN_{SD}))_{K_{MUG}}$$

Step 17: Since the session key is $SK = h(h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)||h(ID_{SD}||RN_{SD}))$, we can obtain S_{17} from S_{12}, S_{16}, and A_9.

$$S_{17} : MU| \equiv SD| \equiv (MU \overset{SK}{\longleftrightarrow} SD) \quad \textbf{(Goal 2)}$$

Step 18: From S_4, S_8, and A_{10}, we can get S_{18}.

$$S_{18} : SD| \equiv MU| \equiv (MU \overset{SK}{\longleftrightarrow} SD) \quad \textbf{(Goal 4)}$$

Step 19: S_{19} can obtained from S_{17} and A_{11}.

$$S_{19} : MU| \equiv (MU \overset{SK}{\longleftrightarrow} SD) \quad \textbf{(Goal 1)}$$

Step 20: S_{20} can obtained using S_{18} and A_{12}.

$$S_{20} : SD| \equiv (MU \overset{SK}{\longleftrightarrow} SD) \quad \textbf{(Goal 3)}$$

Therefore, MU, HGW, and SD can perform the secure mutual authentication in our protocol.

7.3. ROR Model

The session key security of the proposed protocol is demonstrated using the ROR model [11]. We interpret the ROR model before proving the session key security of the proposed protocol. In the authentication and key agreement phase of the proposed protocol, we have three participants $\mathcal{P}^t$, which are mobile user $\mathcal{P}^{t_1}_{MU}$, home gateway $\mathcal{P}^{t_2}_{HGW}$, and smart device $\mathcal{P}^{t_3}_{SD}$. These are instances t_1, t_2, and t_3 for MU, HGW, and SD, respectively. $\mathcal{A}$ can eavesdrop, intercept, or modify transmitted messages through an insecure channel. In addition, $\mathcal{A}$ can simulate active and passive attacks by executing various queries defined in the ROR model, such as *Execute*, *CorruptMD*, *Reveal*, *Send*, and *Test* queries. Detailed instructions of the queries are below.

- $Execute(\mathcal{P}^{t_1}_{MU}, \mathcal{P}^{t_2}_{HGW}, \mathcal{P}^{t_3}_{SD})$: $\mathcal{A}$ performs this query to obtain transmitted messages over a public channel between MU, HGW, and SD.
- $CorruptMD(\mathcal{P}^{t_1}_{MU})$: This query represents that $\mathcal{A}$ can extract sensitive information stored in the mobile device of MU.
- $Reveal(\mathcal{P}^t)$: This query is that $\mathcal{A}$ reveals the current session key SK between $\mathcal{P}^{t_1}_{MU}$ and $\mathcal{P}^{t_3}_{SD}$. If an adversary $\mathcal{A}$ cannot reveal the session key SK between $\mathcal{P}^{t_1}_{MU}$ and $\mathcal{P}^{t_3}_{SD}$ using the $Reveal(\mathcal{P}^t)$ query, then SK is secure.
- $Send(\mathcal{P}^t, M)$: With this query, $\mathcal{A}$ can send the message M to $\mathcal{P}^t$ and receive a response message.
- $Test(\mathcal{P}^t)$: Before the start of the game, a fair coin fc is tossed and the result becomes only known to $\mathcal{A}$. $\mathcal{A}$ uses this result to make a decision of the *Test* query. If $\mathcal{A}$ runs

the *Test* query and the session key SK is fresh, $\mathcal{P}^t$ returns SK for $fc = 1$ or a random number for $fc = 0$. Otherwise, it returns a null ($\perp$).

After $\mathcal{A}$ performs the *Test* query on $\mathcal{P}^t$, $\mathcal{A}$ must distinguish the result value. $\mathcal{A}$ uses the output of the *Test* query for checking the consistency of the random bit fc. $\mathcal{A}$ wins the game when the guessed bit fc' is equal to fc. Moreover, all participants have access to a collision-resistant cryptographic one-way hash function $h(\cdot)$. We model $h(\cdot)$ as a random oracle, *Hash*.

7.3.1. Security Proof

We prove the session key security of the proposed protocol using Zipf's law [41].

Theorem 1. *$\mathcal{A}$ can break the session key security of the proposed protocol. We denote the advantage of $\mathcal{A}$ running in polynomial time as $Adv_{\mathcal{A}}$. Then, we obtain the following.*

$$Adv_{\mathcal{A}} \leq \frac{q_h^2}{|Hash|} + 2\{C \cdot q_{send}^s\}$$

Here, q_h is the number of Hash queries, $|Hash|$ is the range space of the hash function $h(\cdot)$, and q_{send} is the number of Send queries. In addition, C and s denote Zipf's parameters [41].

Proof. The proof of Theorem 1 is similar as presented in [42,43]. We prove the session key security through a sequence of four games, GM_i, where $i \in [0,3]$. $Succ_{\mathcal{A},i}$ indicates the event that $\mathcal{A}$ wins GM_i by guessing the random bit fc correctly. We denote the advantage of $\mathcal{A}$ winning the game GM_i as $Pr[Succ_{\mathcal{A},GM_i}]$. In the following, we describe each game.

- GM_0: This game allows $\mathcal{A}$ to execute the real attack against the proposed protocol. $\mathcal{A}$ chooses a random bit fc at the beginning of GM_0. Then, we obtain the following in accordance with this game.

$$Adv_{\mathcal{A}} = |2Pr[Succ_{\mathcal{A},GM_0}] - 1| \quad (1)$$

- GM_1: In this game, $\mathcal{A}$ runs the $Execute(\mathcal{P}_{MU}^{t_1}, \mathcal{P}_{HGW}^{t_2}, \mathcal{P}_{SD}^{t_3})$ query and eavesdrops transmitted messages $\{PID_{MU}, M_1, C_1, V_{MU}\}$, $\{PID_{MU}, M_3, C_2, V_{MUG}\}$, $\{M_4, V_{SD}\}$, and $\{M_5, V_{GSD}\}$. Then, $\mathcal{A}$ executes *Reveal* and *Test* queries to validate whether the derived session key is real or not. In our protocol, the session key is constructed as $SK = h(h(ID_{MU}||RN_{MU})||h(ID_G||RN_G)||h(ID_{SD}||RN_{SD}))$. To derive the session key, $\mathcal{A}$ needs to know the identities and random nonces of MU, HGW, and SD. Consequently, there are no instances in which $\mathcal{A}$ increases GM_1's winning probability. Therefore, GM_0 and GM_1 turn out to be indistinguishable, and we can obtain the following.

$$Pr[Succ_{\mathcal{A},GM_1}] = Pr[Succ_{\mathcal{A},GM_0}] \quad (2)$$

- GM_2: To obtain the session key, $\mathcal{A}$ performs *Hash* and *Send* queries in this game. $\mathcal{A}$ can perform an active attack by modifying exchanged messages. However, all exchanged messages are constructed with secret credentials and random nonces, and protected using one-way hash function $h(\cdot)$. In addition, $\mathcal{A}$ is difficult to derive secret credentials and random nonces because it is a computationally infeasible problem according to the property of $h(\cdot)$. Hence, we can get the following result through the use of birthday paradox [44].

$$|Pr[Succ_{\mathcal{A},GM_2}] - Pr[Succ_{\mathcal{A},GM_1}]| \leq \frac{q_h^2}{2|Hash|} \quad (3)$$

- GM_3: In the final game GM_3, $\mathcal{A}$ can try to get the session key with the $CorruptMD$ query. By the $CorruptMD$ query, $\mathcal{A}$ can extract sensitive values $\{A_1, A_2, A_3, A_4\}$ stored in the mobile device of MU. Sensitive values are expressed as $A_1 = r_{MU} \oplus$

$h(ID_{MU}||PW_{MU})$, $A_2 = h(ID_{MU}||PW_{MU}||r_{MU}||HPW_{MU})$, $A_3 = RID_{MU} \oplus h(PID_{MU}||HPW_{MU})$, and $A_4 = K_{MUG} \oplus h(RID_{MU}||HPW_{MU})$. Since $\mathcal{A}$ has no knowledge of ID_{MU} and PW_{MU}, $\mathcal{A}$ cannot derive secret values r_{MU} and K_{MUG} from the extracted values. Besides, it is a computationally infeasible task for $\mathcal{A}$ to guess ID_{MU} and PW_{MU} simultaneously. In conclusion, GM_2 and GM_3 are indistinguishable. By utilizing Zipf's law, the following result can be obtained.

$$|Pr[Succ_{\mathcal{A},GM_3}] - Pr[Succ_{\mathcal{A},GM_2}]| \leq C \cdot q_{send}^s \tag{4}$$

As all games have been run, $\mathcal{A}$ must guess the bit for winning the game. Therefore, we can obtain the following result.

$$Pr[Succ_{\mathcal{A},GM_3}] = \frac{1}{2} \tag{5}$$

From Equations (1) and (2), we obtain the result as follows.

$$\frac{1}{2}Adv_{\mathcal{A}} = |Pr[Succ_{\mathcal{A},GM_0} - \frac{1}{2}]| = |Pr[Succ_{\mathcal{A},GM_1} - \frac{1}{2}]|. \tag{6}$$

With Equations (5) and (6), we derive the below equation.

$$\frac{1}{2}Adv_{\mathcal{A}} = |Pr[Succ_{\mathcal{A},GM_1}] - Pr[Succ_{\mathcal{A},GM_3}]|. \tag{7}$$

By using the triangular inequality, we can have the following result with Equations (4), (5), and (7).

$$\begin{aligned} \frac{1}{2}Adv_{\mathcal{A}} &= |Pr[Succ_{\mathcal{A},GM_1}] - Pr[Succ_{\mathcal{A},GM_3}]| \\ &\leq |Pr[Succ_{\mathcal{A},GM_1}] - Pr[Succ_{\mathcal{A},GM_2}]| \\ &\quad + |Pr[Succ_{\mathcal{A},GM_2}] - Pr[Succ_{\mathcal{A},GM_3}]| \\ &\leq \frac{q_h^2}{2|Hash|} + C \cdot q_{send}^s \end{aligned} \tag{8}$$

Finally, by multiplying both sides of Equation (8) by two, we can obtain the required result.

$$Adv_{\mathcal{A}} \leq \frac{q_h^2}{|Hash|} + 2\{C \cdot q_{send}^s\} \tag{9}$$

Therefore, we prove Theorem 1. □

7.4. AVISPA Tool

We utilized the AVISPA tool [7–9] to verify the security of our protocol against MITM and replay attacks. The AVISPA tool uses a role based language, High-Level Protocols Specification Language (HLPSL), to specify actions of each protocol participant [45]. For the security analysis, the HLPSL is entered and translated into the Intermediate Format (IF) in the AVISPA tool. If the IF becomes the input of the back-end, the back-end outputs the security analysis result as the Output Format (OF). The back-end of the AVISPA tool consists of four components, including SAT-based Model-Checker (SATMC), Tree-Automata-based Protocol Analyzer (TA4SP), On-the-Fly-Model-Checker (OFMC), and CL-based Attack Searcher (CL-AtSe). If the OF is SAFE for the back-end, the proposed protocol prevents MITM and replay attacks. We use OFMC and CL-AtSe for the proposed protocol, since SATMC and TA4SP do not support XOR operations.

7.4.1. Specifications of the Proposed Protocol

We set up the session, environment, and security goals using the HLPSL language. Details of these are shown in Figure 6. In *session* and *environment*, we specify instances of each role and construct the whole protocol session. In addition, we state the security goals of the proposed protocol. *secrecy* is used to check secret values are explicitly undisclosed and *authentication* is used to verify the validity of secret values between entities. Through *secrecy* and *authentication*, we can confirm that the proposed protocol is resistant to MITM and replay attacks.

```
role session(MU, SD, HGW, RA : agent, SKmura, SKsdra : symmetric_key, H: hash_func)

def=
local SN1, SN2, SN3, SN4, RV1, RV2, RV3, RV4 : channel(dy)
composition
user(MU, SD, HGW, RA, SKmura, SKsdra, H, SN1, RV1)
/\ device(MU, SD, HGW, RA, SKmura, SKsdra, H, SN2, RV2)
/\ gate(MU, SD, HGW, RA, SKmura, SKsdra, H, SN3, RV3)
/\ register(MU, SD, HGW, RA, SKmura, SKsdra, H, SN4, RV4)
end role

role environment()
def=
const mu, sd, hgw, ra : agent,
skmura, sksdra: symmetric_key,
h: hash_func,
idmu, idsd, idg, kra, ksd, kmug, kgsd, pidmu, pidsd: text,
mu_ hgw _rnmu, hgw _sd_rng, sd_ hgw _rnsd, hgw _mu_rng, hgw _mu_rnsd: protocol_id,
sp1, sp2, sp3, sp4: protocol_id

intruder_knowledge = {mu, sd, hgw, ra, pidmu, pidsd, h}
composition
session(mu, sd, hgw, ra, skmura, sksdra, h) /\session(i, sd, hgw, ra, skmura, sksdra, h)
/\session(mu, i, hgw, ra, skmura, sksdra, h) /\ session(mu, sd, i, ra, skmura, sksdra, h)
/\session(mu, sd, hgw, i, skmura, sksdra, h)

end role

security goals
secrecy_of sp1, sp2, sp3, sp4
authentication_on mu_hgw_rnmu
authentication_on hgw _sd_rng
authentication_on sd_ hgw _rnsd
authentication_on hgw _mu_rng
authentication_on hgw _mu_rnsd
end goal

environment()
```

Figure 6. Roles of session, environment, and security goals.

As shown in Figure 7, if the registration process is started at state 0, MU generates identity ID_{MU} and password PW_{MU}, and calculates PID_{MU} at state 1. Then, MU sends the registration request message $\{PID_{MU}\}$ to RA. After receiving secret values $\{K_{MUG}, RID_{MU}\}$ from RA, MU updates the state from 1 to 2. Then, MU stores secret values encrypted with the ID_{MU} and PW_{MU} in the mobile device. Then, MU transmits the authentication request message $\{PID_{MU}, M_1, C_1, V_{MU}\}$ to HGW. Upon receiving the message $\{M_5, V_{GSD}\}$ in state 2, MU updates the state from 2 to 3 and checks $V^*_{GSD} \stackrel{?}{=} V_{GSD}$. If the condition is met, MU authenticates HGW successfully. Then, MU computes M_6 and sends it to HGW. The roles of HGW, SD, and RA are similar to the roles of MU.

```
role user(MU, SD, HGW, RA: agent, SKmura, SKsdra : symmetric_key, H: hash_func, SND, RCV : channel(dy))

played_by MU
def=
local State: nat,
   IDmu, PWmu, Rmu, PIDmu, Kmug, Kra, Rra, RIDmu, HPWmu, A1, A2, A3, A4, IDsd, Rsd, PIDsd, Kgsd, B1, B2, Ksd: text,
   RNmu, M1, C1, Vmu, IDg, RNg, M2, M3, C2, Vmug, RNsd, SK, M4, Vsd, M5, Vgsd, PIDmunew, RIDmunew, A3new, A4 new, M6: text

const sp1, sp2, sp3, sp4, mu_hgw_rnmu, hgw_sd_rng, sd_hgw_rnsd, hgw_mu_rng, hgw_mu_rnsd: protocol_id
init State := 0
transition

%%%%%%%%%%%%%Registration phase
1. State = 0 /\ RCV(start) =|>
State' := 1 /\ Rmu' := new() /\ PIDmu' := H(IDmu.Rmu')
   /\ SND({PIDmu'}_SKmura)
      /\ secret({IDmu, PWmu}, sp1, {MU})

2. State = 1 /\ RCV ({H(H(IDmu.Rmu').Kra.Rra).H(H(IDmu.Rmu').H(H(IDmu.Rmu').Kra.Rra)).Rsd'}_SKmura)=|>
 State' := 2 /\ HPWmu' := H(PWmu.Rmu')
      /\ A1' := xor(Rmu',H(IDmu.PWmu)) /\ A2' := H(IDmu.PWmu.Rmu'.HPWmu')
      /\ A3' := xor(H(H(IDmu.Rmu').H(H(IDmu.Rmu').Kra.Rra)),H(Rmu'.HPWmu'))
      /\ A4' := xor(H(H(IDmu.Rmu').Kra.Rra), H(H(H(IDmu.Rmu').H(H(IDmu.Rmu').Kra.Rra)).HPWmu'))

%%%%%%%%%%%%%Authentication & Key agreement phase
      /\ RNmu' := new()
      /\ M1' := xor(H(H(IDmu.Rmu').H(H(IDmu.Rmu').H(H(IDmu.Rmu').Kra.Rra)).H(H(IDmu.Rmu').Kra.Rra)),H(RNmu'.H(IDsd.Rsd')))
   /\ C1' := xor(H(IDmu.RNmu'),H(H(H(IDmu.Rmu').Kra.Rra).RNmu'))
   /\ Vmu' := H(H(IDmu.Rmu').H(H(IDmu.Rmu').H(H(IDmu.Rmu').Kra.Rra)).RNmu'.H(IDsd.Rsd').H(H(IDmu.Rmu').Kra.Rra))
      /\ SND(H(IDmu.Rmu').M1'.C1'.Vmu')
      /\ witness(MU,GWN,mu_hgw_rnmu,RNmu')

3. State = 2 /\ RCV(xor(H(H(H(IDmu.Rmu').H(H(IDmu.Rmu').Kra.Rra)).RNmu'),(H(IDg.RNg').H(IDsd.RNsd').H(H(IDmu.Rmu')
. RNmu'))).H(H(IDmu.Rmu').RNmu'.H(IDg.RNg').H(IDsd.RNsd').H(H(IDmu.Rmu').Kra.Rra))) =|>
State' := 3 /\ SK' := H(H(IDmu.RNmu').H(IDg.RNg').H(IDsd.RNsd'))
            /\ M6' := H(SK'. H(H(IDmu.Rm').RNmu'))
            /\ SND(M6')
            /\ request(GWN,MU,hgw_mu_rng,RNg')
            /\ request(GWN,MU,hgw_mu_rnsd,RNsd')
end role
```

Figure 7. Roles of *MU*.

7.4.2. Result of AVISPA

We use OFMC and CL-AtSe for XOR operations to show the security analysis result. The OFMC estimates that the proposed protocol withstands the MITM attack, and CL-AtSe assesses our protocol is resistant to the replay attack. Figure 8 shows the OF of OFMC and CL-AtSe back-ends for the proposed protocol. The output shows that the proposed protocol is SAFE in OFMC and CL-AtSe back-ends. Thus, our protocol successfully satisfies the specified security goals. In other words, our protocol withstands MITM and replay attacks.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/jh.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.01s
  searchTime: 36.54s
  visitedNodes: 0 nodes
  depth: 14 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/span/testsuite/results/jh.if

GOAL
  As Specified
BACKEND
  CL-AtSe

STATISTICS
  Analysed   : 3124 states
  Reachable  : 624 states
  Translation: 0.23 seconds
  Computation : 0.02 seconds
```

Figure 8. Results of analysis using OFMC and CL-AtSe.

8. Performance and Security Analyses

This section shows the comparison results of the proposed protocol with similar protocols [6,31,34,38], including computational and communication costs, and security properties.

8.1. Computational Costs

The computational costs are analyzed for our protocol and related existing protocols [6,31,34,38]. For comparison, we refer to the work [46]. T_m, T_R, T_h, and T_s denote the execution times of an ECC point multiplication (≈7.3529 ms), fuzzy extractor function (≈7.3529 ms), a hash function (≈0.0004 ms), and symmetric key encryption/decryption (≈0.1303 ms), respectively. Table 3 contains the result of the computational costs comparison. Although the proposed protocol has a slightly higher computational cost than the low-security-risk path of Xiang and Zheng's protocol [6], our protocol provides more robust security. Moreover, the proposed protocol has a lower computational cost compared with the other related protocols, except for the low-security-risk path of Xiang and Zheng's protocol [6].

Table 3. Computational costs comparison.

Protocol	Total	Computational Costs
Shuai et al. [31]	$3T_m + 16T_h$	22.0651 ms
Wazid et al. [34]	$25T_h + 1T_R + 4T_s$	7.8841 ms
Banerjee et al. [38]	$26T_h + 1T_R$	7.3633 ms
Xiang and Zheng [6]	Low-security risk: $11T_h$	0.0044 ms
	High-security risk: $11T_h + 4T_s$	0.5256 ms
Ours	$42T_h$	0.0168 ms

8.2. Communication Costs

The communication cost of our protocol is compared to those costs of other related protocols [6,31,34,38]. Referring to the paper [31], we define that an ECC point, symmetric key encryption/decryption, hash function, random number, identity, and timestamp are 320, 256, 160, 160, 128, and 32 bits. We estimate the message header as Internet Protocol version 4 (IPv4) packet header, 4 bits. In the authentication and key agreement phase of the proposed protocol, exchanged messages $\{PID_{MU}, M_1, C_1, V_{MU}\}$, $\{PID_{MU}, M_3, C_2, V_{MUG}\}$, $\{M_4, V_{SD}\}$, $\{M_5, V_{GSD}\}$, and M_6 need 640, 640, 320, 20, and 160 bits, respectively. Consequently, our protocol has 2080 bits as the total communication cost. In Table 4, we show the results of the communication costs comparison. Although our protocol has a higher communication cost than some of the existing protocols [6,31,38], it provides more efficient computational costs and security.

Table 4. Communication costs comparison.

Protocol	Communication Costs	Number of Messages
Shuai et al. [31]	(960 + 320 + 320 + 320) = 1920 bits	4
Wazid et al. [34]	(480 + 960 + 512 + 1408) = 3360 bits	4
Banerjee et al. [38]	(448 + 320 + 320 + 320) = 1408 bits	4
Xiang and Zheng [6]	Low-security risk: (132 + 324 + 324) = 780 bits	3
	High-security risk: (132 + 676 + 676) = 1484 bits	3
Ours	(640 + 640 + 320 + 320 + 160) = 2080 bits	5

8.3. Security Properties

In Table 5, we present security properties of the proposed protocol and those of models by Shuai et al. [31], Wazid et al. [34], Banerjee et al. [38], and Xiang and Zheng [6]. In contrast with the other protocols [6,31,34,38], our protocol prevents more attacks. Thus, the proposed protocol meets more security requirements compared to related protocols.

Table 5. Security properties.

Security Properties	[31]	[34]	[38]	[6]	Ours
Impersonation attack	○	○	○	×	○
Session key disclosure attack	○	○	○	×	○
Replay attack	○	○	○	○	○
MITM attack	○	○	○	○	○
Off-line guessing attack	×	○	○	○	○
Stolen smart device attack	-	-	-	×	○
Privileged-insider attack	○	○	○	×	○
Known session-secret temporary information attack	-	-	○	×	○
Desynchronization attack	○	×	-	×	○
Perfect forward secrecy	×	×	-	×	○
Mutual authentication	○	○	○	×	○
Anonymity	○	×	×	×	○
Untraceability	○	○	×	×	○

○: Secure. ×: Insecure. -: Not considered.

9. Conclusions

We proved that Xiang and Zheng's protocol does not perform secure mutual authentication. We also discovered that their protocol is vulnerable to impersonation, stolen smart device, and session key disclosure attacks. To deal with the security threats to Xiang and Zheng's protocol, we proposed a secure and lightweight authentication protocol for IoT-based smart homes. We demonstrated that the proposed protocol is secure against various attacks, including impersonation, replay, MITM, and session key disclosure attacks. We performed the BAN logic test to show that our protocol ensures secure mutual authentication. Furthermore, we demonstrated that the proposed protocol provides session key security and resists replay and MITM attacks by utilizing the ROR model and the AVISPA tool. We compared our protocol with associated existing protocols in terms of security properties, and computational and communication costs. In conclusion, our protocol provides better security and low computational costs. When we consider all perspectives of security and costs, our protocol is suitable for practical IoT-based smart home environments. In the future, we will develop a better protocol and implement it in an actual environment.

Author Contributions: Conceptualization, J.O.; formal analysis, J.L., S.S. and M.K.; investigation, S.Y.; methodology, J.O.; software, S.Y. and J.L.; supervision, Y.P.; validation, S.S., M.K. and Y.P.; writing—original draft, J.O.; writing—review and editing, S.Y., J.L., S.S., and Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under grant 2020R1I1A3058605, and in part by the BK21 FOUR project funded by the Ministry of Education, Korea under grant 4199990113966.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Shin, S.; Kwon, T. A lightweight three-factor authentication and key agreement scheme in wireless sensor networks for smart homes. *Sensors* **2019**, *19*, 2012. [CrossRef] [PubMed]
2. Naoui, S.; Elhdhili, M.E.; Saidane, L.A. Lightweight and secure password based smart home authentication protocol: LSP-SHAP. *J. Netw. Syst. Manag.* **2019**, *27*, 1020–1042. [CrossRef]
3. Baruah, B.; Dhal, S. A two-factor authentication scheme against FDM attack in IFTTT based smart home system. *Comput. Secur.* **2018**, *77*, 21–35. [CrossRef]

4. Kumar, P.; Gurtov, A.; Iinatti, J.; Ylianttila, M.; Sain, M. Lightweight and secure session-key establishment scheme in smart home environments. *IEEE Sens. J.* **2015**, *16*, 254–264. [CrossRef]
5. Kumar, P.; Braeken, A.; Gurtov, A.; Iinatti, J.; Ha, P.H. Anonymous secure framework in connected smart home environments. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 968–979. [CrossRef]
6. Xiang, A.; Zheng, J. A situation-aware scheme for efficient device authentication in smart grid-enabled home area networks. *Electronics* **2020**, *9*, 989. [CrossRef]
7. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 10 November 2020).
8. SPAN: A Security Protocol Animator for AVISPA. Available online: http://www.avispa-project.org/ (accessed on 10 November 2020).
9. Mandal, S.; Bera, B.; Sutrala, A.K.; Das, A.K.; Choo, K.R.; Park, Y. Certificateless-signcryption-based three-factor user access control scheme for IoT environment. *IEEE Internet Things J.* **2020**, *7*, 3184–3197. [CrossRef]
10. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
11. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password based authenticated key exchange in the three-party setting. In *Public Key Cryptgraphy*; Springer: Les Diablerets, Switzerland, 2005; pp. 65–84.
12. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
13. Lee, J.; Yu, S.; Kim, M.; Park, Y.; Das, A.K. On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks. *IEEE Access* **2020**, *8*, 107046–107062. [CrossRef]
14. Yu, S.; Lee, J.; Lee, K.; Park, K.; Park, Y. Secure authentication protocol for wireless sensor networks in vehicular communications. *Sensors* **2018**, *18*, 3191. [CrossRef]
15. Canetti, R.; Krawczyk, H. Universally composable notions of key exchange and secure channels. In *International Conference on the Theory and Applications of Cryptographic Thechniques (EUROCRYPT'02)*; Springer: Amsterdam, The Netherlands, 2002; pp. 337–351.
16. Wazid, M.; Bagga, P.; Das, A.K.; Shetty, S.; Rodrigues, J.J.P.C.; Park, Y. AKM-IoV: Authenticated key management protocol in fog computing-based internet of vehicles deployment. *IEEE Internet Things J.* **2019**, *6*, 8804–8817. [CrossRef]
17. Yu, S.; Lee, J.; Park, Y.; Park, Y.; Lee, S.; Chung, B. A secure and efficient three-factor authentication protocol in global mobility networks. *Appl. Sci.* **2020**, *10*, 3565. [CrossRef]
18. Roy, S.; Chatterjee, S.; Das, A.K.; Chattopadhyay, S.; Kumar, N.; Vasilakos, A.V. On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services. *IEEE Access* **2017**, *5*, 25808–25825. [CrossRef]
19. Park, K.; Park, Y.; Park, Y.; Das, A.K. 2PAKEP: Provably secure and efficient two-party authenticated key exchange protocol for mobile environment. *IEEE Access* **2018**, *6*, 30225–30241. [CrossRef]
20. Chaudhry, S.A.; Alhakami, H.; Baz, A.; Al-Turjman, F. Securing demand response management: A certificate-based access control in smart grid edge computing infrastructure. *IEEE Access* **2020**, *8*, 101235–101243. [CrossRef]
21. Park, K.; Noh, S.; Lee, H.; Das, A.K.; Kim, M.; Park, Y.; Wazid, M. LAKS-NVT: Provably secure and lightweight authentication and key agreement scheme without verification table in medical internet of things. *IEEE Access* **2020**, *8*, 119387–119404. [CrossRef]
22. Ul Haq, I.; Wang, J.; Zhu, Y. Secure two-factor lightweight authentication protocol using self-certified public key cryptography for multi-server 5G networks. *J. Netw. Comput. Appl.* **2020**, *161*, 102660. [CrossRef]
23. Amin, R.; Islam, S.H.; Biswas, G.P.; Khan, M.K.; Kumar, N. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Gener. Comput. Syst.* **2018**, *80*, 483–495. [CrossRef]
24. Chandrakar, P.; Om, H. A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC. *Comput. Commun.* **2017**, *110*, 26–34. [CrossRef]
25. Santoso, F.K.; Vun, N.C.H. Securing IoT for smart home system. In Proceedings of the 2015 International Symposium on Consumer Electronics (ISCE), Madrid, Spain, 24–26 June 2015; pp. 1–2.
26. Fakroon, M.; Alshahrani, M.; Gebali, F.; Traore, I. Secure remote anonymous user authentication scheme for smart home environment. *Internet Things* **2020**, *9*, 100158. [CrossRef]
27. Banerjee, S.; Odelu, V.; Das, A.K.; Chattopadhyay, S.; Rodrigues, J.J.P.C.; Park, Y. Physically secure lightweight anonymous user authentication protocol for internet of things using physically unclonable functions. *IEEE Access* **2019**, *7*, 85627–85644. [CrossRef]
28. Dey, S.; Hossian, A. Session-key establishment and authentication in a smart home network using public key cryptography. *IEEE Sens. Lett.* **2019**, *3*, 7500204. [CrossRef]
29. Gaba, G.S.; Kumar, G.; Monga, H.; Kim, T.; Kumar, P. Robust and lightweight mutual authentication scheme in distributed smart environments. *IEEE Access* **2020**, *8*, 69722–69733. [CrossRef]
30. Kumar, P.; Chouhan, L. A privacy and session key based authentication scheme for medical IoT networks. *Comput. Commun.* **2021**, *166*, 154–164. [CrossRef]
31. Shuai, M.; Yu, N.; Wang, H.; Xiong, L. Anonymous authentication scheme for smart home environment with provable security. *Comput. Secur.* **2019**, *86*, 132–146. [CrossRef]
32. Vaidya, B.; Park, J.H.; Yeo, S.S.; Rodrigues, J.J. Robust one-time password authentication scheme using smart card for home network environment. *Comput. Commun.* **2011**, *34*, 326–336. [CrossRef]
33. Kim, H.J.; Kim, H.S. AUTH HOTP-HOTP based authentication scheme over home network environment. In *International Conference on Computational Science and Its Applications*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 622–637.

34. Wazid, M.; Das, A.K.; Odelu, V.; Kumar, N.; Susilo, W. Secure remote user authenticated key establishment protocol for smart home environment. *IEEE Trans. Dependable Secur. Comput.* **2017**, *17*, 391–406. [CrossRef]
35. Lyu, Q.; Zheng, N.; Liu, H.; Gao, C.; Chen, S.; Liu, J. Remotely access "my" smart home in private: An anti-tracking authentication and key agreement scheme. *IEEE Access* **2019**, *7*, 41835–41851. [CrossRef]
36. Poh, G.S.; Gope, P.; Ning, J. Privhome: Privacy-preserving authenticated communication in smart home environment. *IEEE Trans. Dependable Secur. Comput.* **2019**. [CrossRef]
37. Irshad, A.; Usman, M.; Chaudry, S.A.; Bashir, A.K.; Jolfaei, A.; Srivastava, G. Fuzzy-in-the-loop-driven low-cost and secure biometric user access to server. *IEEE Trans. Reliab.* **2020**. [CrossRef]
38. Banerjee, S.; Odelu, V.; Das, A.K.; Chattopadhyay, S.; Park, Y. An efficient, anonymous and robust authentication scheme for smart home environments. *Sensors* **2020**, *20*, 1215. [CrossRef] [PubMed]
39. AL-Turjman, F.; Deebak, D.B. Seamless authentication: For IoT-big data technologies in smart industrial application systems. *IEEE Trans. Ind. Inf.* **2020**. [CrossRef]
40. Lee, J.; Yu, S.; Park, K.; Park, Y.; Park, Y. Secure three-factor authentication protocol for multi-gateway IoT environments. *Sensors* **2019**, *19*, 2358. [CrossRef]
41. Wang, D.; Cheng, H.; Wang, P.; Huang, X.; Jian, G. Zipf's law in passwords. *IEEE Trans. Inf. Forensics Secur.* **2017**, *12*, 2776–2791. [CrossRef]
42. Park, K.; Park, Y.; Das, A.K.; Yu, S.; Lee, J.; Park, Y. A dynamic privacy-preserving key management protocol for V2G in social internet of things. *IEEE Access* **2019**, *7*, 76812–76832. [CrossRef]
43. Yu, S.; Lee, J.; Park, K.; Das, A.K.; Park, Y. IoV-SMAP: Secure and efficient message authentication protocol for IoV in smart city environment. *IEEE Access* **2020**, *8*, 167875–167886. [CrossRef]
44. Boyko, V.; MacKenzie, P.; Patel, S. Provably secure password-authenticated key exchange using Diffie-Hellman. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Bruges, Belgium, 14–18 May 2000; pp. 156–171.
45. Vigano, L. Automated security protocol analysis with the AVISPA tool. *Electron. Notes Theor. Comput. Sci.* **2006**, *155*, 61–86. [CrossRef]
46. Mo, J.; Chen, H. A lightweight secure user authentication and key agreement protocol for wireless sensor networks. *Secur. Commun. Netw.* **2019**, *2019*, 2136506. [CrossRef]