sensors

MDPI

Article

An Efficient, Anonymous and Robust Authentication Scheme for Smart Home Environments

Soumya Banerjee [1], Vanga Odelu [2], Ashok Kumar Das [3], Samiran Chattopadhyay [1,4] and Youngho Park [5,*]

1 Department of Information Technology, Jadavpur University, Salt Lake City, Kolkata 700 098, India; soumyaBanerjee@outlook.in (S.B.); samirancju@gmail.com (S.C.)
2 Department of Computer Science and Information Systems, Birla Institute of Technology & Science, Pilani Hyderabad Campus, Hyderabad 500 078, India; odelu.vanga@hyderabad.bits-pilani.ac.in
3 Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India; iitkgp.akdas@gmail.com or ashok.das@iiit.ac.in
4 Northumbria University, Newcastle upon Tyne NE1 8ST, UK
5 School of Electronics Engineering, Kyungpook National University, 80 Daehak-ro, Sangyeok-dong, Buk-gu, Daegu 41566, Korea; parkyh@knu.ac.kr
* Correspondence: parkyh@knu.ac.kr; Tel.: +82-53-950-5114

Received: 27 January 2020; Accepted: 20 February 2020; Published: 22 February 2020

Abstract: In recent years, the Internet of Things (IoT) has exploded in popularity. The smart home, as an important facet of IoT, has gained its focus for smart intelligent systems. As users communicate with smart devices over an insecure communication medium, the sensitive information exchanged among them becomes vulnerable to an adversary. Thus, there is a great thrust in developing an anonymous authentication scheme to provide secure communication for smart home environments. Most recently, an anonymous authentication scheme for smart home environments with provable security has been proposed in the literature. In this paper, we analyze the recent scheme to highlight its several vulnerabilities. We then address the security drawbacks and present a more secure and robust authentication scheme that overcomes the drawbacks found in the analyzed scheme, while incorporating its advantages too. Finally, through a detailed comparative study, we demonstrate that the proposed scheme provides significantly better security and more functionality features with comparable communication and computational overheads with similar schemes.

Keywords: Internet of Things (IoT); smart homes; anonymous authentication; session key agreement; security; Automated Validation of Internet Security Protocols and Applications (AVISPA)

1. Introduction

Interest in the Internet of Things (IoT) has grown exponentially over recent years, and it is likely to continue growing for the foreseeable future [1]. The smart home as an important IoT application has also gained much interest in recent years. Adoption of home automation systems for monitoring and controlling various smart devices is at an all-time high [2,3]. The reduced operating expenses, coupled with the increased quality of life, encourage the users to rely on these more and more. A smart home reduces expenses while providing higher comfort, security and safety to the users [4]. Additionally, smart homes can provide the elderly and disabled with prompt medical care based on the readings of smart gadgets [5]. However, as a direct result of using these services, a large volume of private and sensitive data is being transmitted over insecure networks. Security and privacy are considered the fundamental requirements for consumer technology deployment [6].

Consider a smart gadget for monitoring a patient. In order to get medical services, the external user (for example, a doctor) needs to have direct access to data sensed by the sensors in the gadget

monitoring the patient's body. Such information will invariably include current vital readings like blood sugar level, blood pressure, etc. For obvious reasons, this information needs to private and confidential. Similarly, data generated from the surveillance system, temperature and movement sensors, or control data for lighting or other appliances need to be secure and confidential. Devices in a smart home can be accessed through a gateway node that connects them to the Internet. To ensure data privacy and integrity, various entities, such as the users, the smart devices, and the gateway node need to generate session keys after their mutual authentication. The generated session keys can then be used for further communication without fear of data compromise.

1.1. Network and Threat Models

We follow the widely accepted network model for the proposed scheme, which is defined in the typical smart home architecture [7] shown in Figure 1. The smart devices connect to the public Internet through the gateway nodes (GWN). Users (U) and smart devices (SD) must be registered or enrolled with the registration authority RA before operating in the network. The RA is a fully trusted entity in the network. The registered mobile users can avail of the services provided by the already enrolled smart devices through the gateway node and negotiate the session keys after mutual authentication.

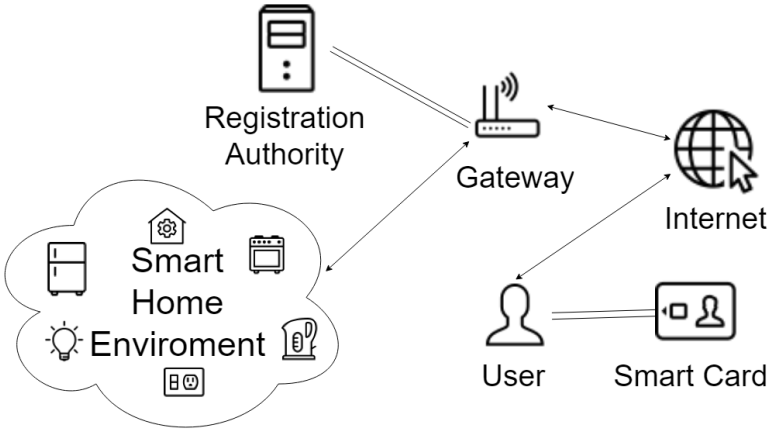

Figure 1. A typical smart home architecture (adapted from [7]).

We evaluate the proposed scheme under the de-facto standard "Dolev-Yao (DY) threat model" [8]. In the DY-threat model, an adversary, say $\mathcal{A}$, has ultimate authority over the communication channel, and consequently he/she is capable of eavesdropping, modifying, dropping, or even inserting forged messages for any communicated messages. Furthermore, it is assumed that $\mathcal{A}$ can physically capture some smart devices, as monitoring the devices 24/7 is not possible, to extract the sensitive information stored in them using power analysis attacks [9]. Moreover, the smart card of a user can be lost or stolen, and the adversary $\mathcal{A}$ can also extract all the sensitive information stored in its memory using power analysis attacks [9]. Both the registration authority (RA) and the gateway node (GWN) are considered trusted in the smart home environment. Furthermore, we use the stronger threat model, known as the "Canetti and Krawczyk's (CK) adversary model" [10], wherein the adversary $\mathcal{A}$, in addition to having all capacities of the DY-therat model, can also compromise ephemeral information like session-specific states and keys. Thus, in the presence of the CK-adversary, a user authentication scheme must be designed such that leakage of ephemeral secrets should have minimal impact on the security of unrelated entities in the authenticated key-exchange scheme [11].

1.2. Research Contributions

The main contributions are given below.

- We first analyze the recently proposed anonymous authentication scheme by Shuai et al. [7] for the smart home environment and then highlight that their scheme fails to resist known attacks,

such as privileged-insider attack, through offline password guessing and lost/stolen smart card attacks, user impersonation attacks, parallel session attacks, and password change attacks.
- We present a more secure user authentication scheme that avoids the security pitfalls demonstrated in Shuai et al.'s scheme.
- Through formal as well as informal security analysis, we show the resistance of the proposed scheme against various potential attacks needed in a smart home environment.
- We then present a comparative study to demonstrate the superior security and functionality features of the proposed scheme relative to the existing relevant authentication schemes.
- Finally, we provide a practical perspective on the applicability of the proposed scheme through a network simulator (NS3) simulation study.

1.3. Related Work

In the last decade, several authors investigated the issues of remote authentication for smart homes. Jeong et al. [12] suggested an authentication protocol for home networks based on "One-Time Passwords (OTPs)" and smart cards. However, their scheme not only transmitted the user identities in plaintext, but also did not provide mutual authentication. Vaidya et al. [13] designed a "remote authentication scheme using lightweight computation modules". Unfortunately, Kim et al. demonstrated that [13] was not only vulnerable to known attacks, but it also failed to provide "user anonymity" and "forward secrecy". To strengthen the security, Kim et al. presented an improved scheme [14] over the Vaidya et al. scheme. [13].

Vaidya et al. [15] presented an "Elliptic Curve Cryptography (ECC)" based device authentication scheme for smart home networks. However, their scheme was found to be susceptible to privileged-insider, password guessing, and user impersonation attacks. Pradeep and Singh [16] proposed a secure three-factor authentication scheme for "ubiquitous computing devices" with a pass-phrase based device integrity check.

Li proposed a lightweight key establishment scheme [17] as a solution to the security issue in smart home energy management systems. Unfortunately, their scheme was not scalable as it requires the management of many keys and certificates. Around the same time, Han et al. [18] designed a key agreement scheme for a secure pairing process for smart home systems. But, their scheme depends on an always-online service by the manufacturer of the devices, which is an infeasible requirement. Additionally, neither the scheme [17] nor the scheme [18] provided "mutual authentication between user and smart devices".

Santoso and Vun [19] suggested an "ECC -based authentication scheme for smart homes", where they presented the idea of using the Wi-Fi gateway as the central node of the system. Unfortunately, their scheme was vulnerable to privileged–insider attack, and consequently, it failed to guarantee user anonymity and untraceability properties.

Kumar et al. [4] designed a "lightweight anonymity preserving authentication scheme for smart home environments". However, their scheme failed to provide "mutual authentication between the user and the smart device". In their scheme, user anonymity and untraceability properties are also compromised.

Wazid et al. [20] suggested a lightweight remote user authentication scheme for the smart home environment which fulfills the design criteria for the smart home environment. Yu and Li [21] proposed another user authentication scheme for the smart home environment. However, their protocol did not necessitate a secure environment for user and device registration. Moreover, their scheme relied on bilinear pairing operations, and as a result, their scheme incurs exceptionally high overheads. Shuai et al. [7] designed an "ECC-based authentication scheme for the smart home environment". However, in this paper, we discuss the advantages and limitations of their scheme in detail. Naoui et al. [22], Fakroon et al. [23] and Dey and Hossain [24] also presented other user authentication schemes for the smart home environment.

2. Review of Shuai et al.'s Scheme

In this section, we briefly review Shuai et al.'s scheme. Their scheme has the following phases: (a) initialization phase, (b) registration phase, (c) login and authentication phase, and (d) password change phase. In this section, we only review the first three phases, and the details regarding the password change phase can be found in the scheme [7].

2.1. Initialization Phase

During initialization, the registration authority (RA) selects an elliptic curve $E(F_p)$ of the form $y^2 = x^3 + ax + b \pmod p$ of order p over finite field F_p with a generator point P, where p is a large prime number and $a, b \in \mathcal{Z}_p = \{0, 1, \cdots, p-1\}$ such that $4a^3 + 27b^2 \neq 0 \pmod p$. RA then creates a private key x and calculates the corresponding public key $X = x \cdot P$. RA selects a "long term key K" and a "cryptographic one-way collision-resistant hash function $h(\cdot)^* : \{0,1\}^* \rightarrow \mathcal{Z}_p^*$", where $\mathcal{Z}_p^* = \{1, 2, \cdots, p-1\}$. RA commits x and K to the GWN and makes $\{E(F_p), P, X, h(\cdot)\}$ public. RA also picks and saves GID into gateway node's memory as its unique identity. In addition, RA generates SID_d as a random unique identity for each smart device SD. These identities are saved to the respective smart devices SD.

2.2. Registration Phase

This phase comprises of the user registration as well as the smart device enrollment phases.

2.2.1. User Registration

A user U registers with the RA through the following steps:

- **Step 1.** U first picks his/her identity ID_u, password PW_u and generates a random secret a. U then calculates pseudo-password $HPW_u = h(PW_i||a)$ and securely dispatches the credentials $\{ID_u, HPW_u\}$ to RA.
- **Step 2.** If ID_u is already registered, RA rejects the request. Otherwise, RA computes $K_{UG} = h(ID_u||K)$, $A_1 = K_{UG} \oplus HPW_u$. RA generates a random value $TEMP$ in order to record the number of user login failures, and sets $TEMP = 0$. Next, RA writes $\{A_i, TEMP\}$ to a smart card SC_u and securely issues SC_u to the user U.
- **Step 3.** On receiving the smart card SC_u, U calculates $A_2 = a \oplus h(ID_u\ ||PW_u)$ and $A_3 = h(ID_u\ ||HPW_u)$, and appends A_2 and A_3 to the smart card SC_u. The smart card SC_u finally contains the credentials $\{A_1, A_2, A_3, TEMP\}$.

2.2.2. Device Enrollment

The steps for smart device, SD's enrollment with the RA:

- **Step 1.** SD first securely transmits its identity SID_d to RA.
- **Step 2.** If SD is already enrolled, the request is rejected by the RA. Otherwise, RA computes $K_{GS} = h(SID_d||K)$ and securely sends the secret key K_{GS} to SD.
- **Step 3.** On receiving the reply, SD saves the secret key K_{GS} in its memory.

2.3. Login and Authentication Phase

For a registered user U to access a smart device SD, he/she must first establish a session key SK after" mutual authentication between U, SD and GWN". The steps for login, and authentication and session key establishment phase are as follows:

- **Step 1.** User U first enters his/her identity ID_u and password PW_u, and calculates $a^* = A_2 \oplus h(ID_u||PW_u)$, $HPW_u^* = h(PW_u||a^*)$ and $A_3^* = h(ID_u||HPW_u^*)$. Only if the check $A_3^* = A_3$ holds, the login is successful. In case of a failed login attempt, the smart card SC_u of the user U

updates $TEMP = TEMP + 1$. This value records the login attempts and if it exceeds a pre-defied threshold, the user U is considered as compromised and is suspended till he/she re-registers.

After a successful login, the smart card SC_u generates two random numbers R_1 and $w \in \mathbb{Z}_p^*$, and computes $K_{GU} = A_1 \oplus HPW_u$, $A_4 = w \cdot P$, $A_5 = w \cdot X$, $DID_u = ID_u \oplus A_5$, $M_1 = (R_1||SID_d) \oplus K_{UG}$ and $V_1 = h(ID_u||R_1||K_{UG}||M_1)$, and sends the login request message $\langle DID_u, A_4, M_1\ V_1 \rangle$ to GWN via open channel.

- **Step 2.** On receiving the login request $\langle DID_u, A_4, M_1\ V_1 \rangle$, GWN computes $A_5^* = x \cdot A_4$, $ID_u^* = DID_u \oplus A_5^*$, $K_{GU} = h(ID_u^*||K)$, $(R_1^*||SID_d) = M_1 \oplus K_{GU}$, $V_1^* = h(IDi^*||R1^*||K_{GU}||M_1)$. Only if the condition $V_1^* = V_1$ holds, GWN believes the legitimacy of the login request. GWN then generates a random number $R_2 \in \mathbb{Z}_p^*$ and computes $K_{GS} = h(SID_d||K)$, $M_2 = (ID_u||GID||R_1||R_2) \oplus K_{GS}$, $V_2 = h(ID_u||GID||K_{GS}||R_1||R_2)$. Finally, GWN sends the authentication request message $\langle M_2, V_2 \rangle$ to SD via public channel.
- **Step 3.** On receiving the message $\langle M_2, V_2 \rangle$, SD calculates $(ID_u||GID||R_1||R_2) = M_2 \oplus K_{GS}$, $V_2^* = h(ID_u||GID||\ K_{GS}||R_1||R_2)$ and checks if $V_2^* = V_2$. If true, SD generates a random number $R_3 \in \mathbb{Z}_p^*$ and computes $SK = h(ID_u||GID||SID_d||R_1||R_2||R_3)$, $M_3 = R_3 \oplus K_{GS}$, $V_3 = h(R_3||K_{GS}||SK)$ and finally transmits the authentication reply message $\langle M_3, V_3 \rangle$ to GWN.
- **Step 4.** On receiving the message $\langle M_3, V_3 \rangle$ from SD, GWN computes $R_3 = M_3 \oplus K_{GS}$, $SK = h(ID_u||\ GID||SID_d||R_1||R_2||R_3)$, $V_3^* = h(R_3||K_{GS}||SK)$, and if $V_3^* == V_3$, GWN computes $M_4 = (GID||R_2||R_3) \oplus K_{GU}$ and $V_4 = h(K_{GU}||SK||R_2||R_3)$, and sends the acknowledgement message $\langle M_4, V_4 \rangle$ to U via public channel.
- **Step 5.** On receiving the message $\langle M_4, V_4 \rangle$ from GWN, U computes $(GID||R_2||R_3) = M4 \oplus K_{GU}$, $SK = h(ID_u||GID||\ SID_d||R_1||R_2||R_3)$ and $V_4^* = h(K_{GU}||SK||R_2||R_3)$, and if $V_4^* = V4$, SD is authenticated by the GWN, and also the session key SK is established between U and SD.

3. Security Vulnerabilities in Shuai et al.'s Scheme

In this section, we cryptanalyze the scheme proposed by Shuai et al. and observe that in the presence of a passive/active adversary, it is vulnerable to several potential attacks. We detail the possible attacks below.

3.1. Privileged-Insider Attack through Offline Password Guessing and Lost/Stolen Smart Card Attacks

Suppose an adversary $\mathcal{A}$, who is also a privileged insider user, acts as an adversary, say $\mathcal{A}$. In this case, $\mathcal{A}$ knows the credentials ID_u and HPW_u of a legitimate registered user U which are submitted to the RA during the user registration phase (see Section 2.2.1), where $HPW_u = h(PW_i||a)$ and a is a random secret. Moreover, if $\mathcal{A}$ can acquire the lost/stolen smart card SC_u of the user U, using the "power analysis attacks" [9], [25], the adversary $\mathcal{A}$ can extract all the credentials $\{A_1, A_2, A_3, TEMP\}$ stored in the memory of SC_u, where $K_{UG} = h(ID_u||K)$, $A_1 = K_{UG} \oplus HPW_u$, $A_2 = a\ \oplus h(ID_u||PW_u)$ and $A_3 = h(ID_u||HPW_u)$. Now, as $A_2 = a\ \oplus h(ID_u||PW_u)$ and $HPW_u = h(PW_u||a)$, $\mathcal{A}$ can form the following relation:

$$HPW_u = h(PW_u||(A_2 \oplus h(ID_u||PW_u))). \tag{1}$$

$\mathcal{A}$ can then guess a password, say PW_u'. Using the guessed password PW_u', and ID_u and A_2, $\mathcal{A}$ further can calculate $HPW_u' = h(PW_u'||\ (A_2 \oplus h(ID_u\ ||PW_u')))$, and verify if the condition $HPW_u' = HPW_u$ is valid or not. If the condition holds, it means that $\mathcal{A}$ is successful in guessing the user U's correct password. Hence, it is clear that the low-entropy guessed passwords are easily guessed and verified in Shuai et al.'s scheme. As a result, Shuai et al.'s scheme is vulnerable to privileged-insider attack with the help of both offline password guessing and lost/stolen smart card attacks.

3.2. User Impersonation and Parallel Session Attacks

A privileged insider adversary $\mathcal{A}$ with the knowledge of registration information ID_u and HPW_u, and extracted A_1 from the stolen smart card SC_u of a valid registered user U (discussed in Section 3.1)

can easily compute secret key $K_{GU} = A_1 \oplus HPW_u$. Consequently, $\mathcal{A}$ can forge the login request message $\langle DID_u, A_4, M_1, V_1 \rangle$ to the GWN in order to impersonate the user U due to the following reason. Since each smart device SD sends its identity SID_d to the RA, the privileged insider adversary $\mathcal{A}$ of the RA also knows it. Now, $\mathcal{A}$ can generate two random numbers R_1' and $w' \in \mathbb{Z}_p^*$, and compute $A_4' = w' \cdot P$, $A_5' = w' \cdot X$, $DID_u' = ID_u \oplus A_5'$, $M_1' = (R_1'||SID_d) \oplus K_{UG}$, $V_1' = h(ID_u||R_1'||K_{UG}||M_1')$. As a result, the adversary $\mathcal{A}$ is able to send a valid login request message $\langle DID_u', A_4', M_1', V_1' \rangle$ to the GWN. Thus, a privileged adversary can impersonate a legal registered user U in Shuai et al.'s scheme.

We consider another attack, where privileged insider adversary $\mathcal{A}$ of the RA, who has calculated K_{GU} from the previous attack, can intercept the message $\langle M4, V4 \rangle$ that is sent from the GWN to a user U. $\mathcal{A}$, having the knowledge of K_{UG} and ID_U, can calculate $(GID||R_2||R_3) = M4 \oplus K_{GU}$ and the session key $SK = h(ID_u \,||GID\, ||SID_d\, ||R_1\, ||R_2\, ||R_3)$. Thus, $\mathcal{A}$ can independently calculate the session key SK making the scheme of Shuai et al. vulnerable to the parallel session attack.

3.3. Password Change Attack

Suppose a privileged insider of the RA being an adversary $\mathcal{A}$ after learning the password PW_u from the previously discussed attack in Section 3.1 can simply execute the password update phase to change a legal registered user U's password if the smart card SC_u of U is being stolen by $\mathcal{A}$. For this purpose, $\mathcal{A}$ has the credentials $\{A_1, A_2, A_3, TEMP\}$ stored in the memory of SC_u, where $K_{UG} = h(ID_u||K)$, $A_1 = K_{UG} \oplus HPW_u$, $A_2 = a \oplus h(ID_u||PW_u)$ and $A_3 = h(ID_u||HPW_u)$. $\mathcal{A}$ first calculates $K_{GU} = A_1 \oplus HPW_u$ using previous registration information HPW_u and $a = A_2 \oplus h(ID_u \,||PW_u)$. Next, $\mathcal{A}$ chooses his/her own password, say PW_u' and calculates $HPW_u' = h(PW_u' \,||a)$, $A_1' = K_{UG} \oplus HPW_u'$, $A_2' = a \oplus h(ID_u \,||PW_u')$ and $A_3' = h(ID_u \,||HPW_u')$. Finally, $\mathcal{A}$ updates the old credentials $\{A_1, A_2, A_3, TEMP\}$ with the newly computed credentials $\{A_1', A_2', A_3', TEMP\}$ in the memory of the smart card SC_u. This clearly shows that the password change attack is easily mounted on Shuai et al.'s scheme.

4. The Proposed Scheme

In this section, we present a more secure "anonymous authentication and session key establishment scheme" for smart home environments, which is free from all the mentioned security vulnerabilities discussed in Section 3. The important phases of our scheme are discussed below.

4.1. Initialization Phase

This phase is similar to that presented in Section 2.1. Note that during initialization, the registration authority (RA) also generates a "long term key K" and a " collision-resistant cryptographic one-way hash function $h(\cdot)^* : \{0,1\}^* \rightarrow \mathcal{Z}_p^*$". RA then commits K to GWN and makes $\{h(\cdot)\}$ public.

4.2. Registration Phase

The registration phase details the procedure for dynamic device enrollment and user registration.

4.2.1. Dynamic Device Enrollment

Any time after initialization, a smart device SD can be enrollment with the RA via secure channel through the following steps:

- **Step 1.** SD first securely transmits its identity SID_d to RA.
- **Step 2.** If SD is already enrolled, the request is rejected by the RA. Otherwise, RA computes the secret key $K_{GS} = h(SID_d||h(K))$, and securely sends K_{GS} to SD and makes SID_j public.
- **Step 3.** On receiving the reply from the RA, SD saves the secret key K_{GS} in its memory.

4.2.2. Mobile User Registration

After system initialization, a mobile user U can be registered with the RA via secure channel.

In our scheme, we use the fuzzy extractor method for user biometric verification [26]. This step is necessary to reduce false negatives during biometric verification. A fuzzy extractor comprises of the following two procedures:

- ***Gen:*** It is a "probabilistic generation function" that computes a pair (σ_u, τ_u) from the user biometrics information. The resultant σ_u is the "biometric secret key" and τ_u is the "public reproduction parameter" necessary for reconstruction of σ_u from Bio'_u, a noisy biometric reading from the same user. Formally, $(\sigma_u, \tau_u) = Gen(Bio_u)$.
- ***Rep:*** It is a "deterministic reproduction method" which constructs the original biometric secret key σ_i using a noisy biometrics reading, Bio'_u and the public reproduction parameter τ_i provided the Hamming distance HD between Bio_u and Bio'_u is less than or equal to a pre-defined error tolerance threshold value, say Δ_t. Formally, $\sigma_u = Rep(Bio'_u, \tau_u)$, with the restriction that $\mid HD(Bio_u, Bio'_u) \leq \Delta_t$.

The following steps are involved in this phase:

- **Step 1.** U selects his/her identity ID_u and securely sends $\{ID_u\}$ to RA.
- **Step 2.** If ID_u is already registered, RA rejects the request. Otherwise, RA generates $R_g, DID_u \in \mathbb{Z}_p$ and computes $K_{UG} = h(ID_u||h(R_g||K))$, and also sets $TEMP = 0$. After that RA commits the tuple $\langle DID_u, ID_u, R_g \rangle$ to the *user_data* table in the gateway node GWN. RA also writes the credentials $\{K_{UG}, DID_u, TEMP\}$ to a smart card SC_u, and securely issues SC_u to the user U.
- **Step 3.** After getting SC_u, U provides a password PW_u and imprints biometric template Bio_u at the sensor of a specific terminal. U uses the probabilistic fuzzy generator function $Gen(Bio_u)$ to calculate the biometric secret ket σ_u and a public reproduction parameter τ_u as $(\sigma_u, \tau_u) = Gen(Bio_u)$. After that, U computes $A_1 = DID_u \oplus h(ID_u||PW_u||\sigma_u)$, $A_2 = h(DID_u||ID_u||\sigma_u||PW_u)$ and $A_3 = K_{UG} \oplus h(ID_u||DID_u||PW_u||\sigma_u)$, and replaces K_{UG} and DID_u in the smart card with A_1, A_2, A_3, τ_u. The smart card SC_u finally contains the credentials $\{A_1, A_2, A_3, \tau_u, TEMP\}$.

The user registration phase is also briefed in Figure 2.

User U	Registration Authority $\{K\}$
Enter ID_u.	
$\xrightarrow[\text{secure channel}]{\langle ID_u \rangle}$	
	Generate $R_g, DID_u \in \mathbb{Z}_p$.
	Calculate $K_{UG} = h(ID_u\|\|h(R_g\|\|K))$.
	Set $TEMP = 0$.
	Save $\langle DID_u, ID_u, R_g \rangle$.
	$\xleftarrow[\text{secure channel}]{SC_u = \langle K_{UG}, DID_u, TEMP \rangle}$
Input PW_u and imprint Bio_u.	
Calculate $(\sigma_u, \tau_u) = Gen(Bio_u)$,	
$A_1 = DID_u \oplus h(ID_u\|\|PW_u\|\|\sigma_u)$,	
$A_2 = h(DID_u\|\|ID_u\|\|\sigma_u\|\|PW_u)$,	
$A_3 = K_{UG} \oplus h(ID_u\|\|DID_u\|\|PW_u\|\|\sigma_u)$.	
Replace K_{UG} and DID_u	
with A_1, A_2, A_3, τ_u in SC_u.	

Figure 2. Summary of user registration.

4.3. Login and Authentication Phase

A registered user U through the following steps can anonymously establish a session key with a smart device SD once mutual authentication in presence of the gateway node GWN is successful.

- **Step 1.** U first inputs his/her identity ID_u and password PW_u, and imprints his/her biometric Bio_u at the sensor of a particular terminal. The smart card SC_u of U then uses public τ_u to compute σ_u from Bio_u as $\sigma_u = Rep(Bio_u, \tau_u)$, and proceeds to calculate $DID_u = A_1 \oplus h(ID_u||PW_u||\sigma_u)$ and $A_2^* = h(DID_u||ID_u||\sigma_u||PW_u)$. If the condition $A_2^* = A_2$ holds, the login is treated as successful one. In case of a failed login attempt, the smart card SC_u increments $TEMP$ and aborts the phase. On the other side, if it exceeds a pre-defined threshold, the user U is considered as compromised, and is suspended till he/she re-registers.

  After a successful login, SC_u generates two random numbers R_1 and $w \in \mathbb{Z}_p^*$, and calculates $K_{UG} = A_3 \oplus h(ID_u||DID_u||PW_u||\sigma_u)$, $M_1 = (R_u||SID_d) \oplus K_{UG}$ and $V_1 = h(ID_u||R_u||K_{UG}||M_1)$, and dispatched the login request message $\langle DID_u, M_1, V_1 \rangle$ to the GWN via public channel.
- **Step 2.** After receiving the login request $\langle DID_u, M_1, V_1 \rangle$, the GWN looks up ID_u, R_g using DID_u from its *user_data* table, and computes $K_{UG} = h(ID_u||h(R_g||K))$, $(R_u||SID_d) = M_1 \oplus K_{UG}$. If R_u is fresh, the GWN calculates $V_1^* = h(ID_u||R_u||K_{UG}||M_1)$. Now, if $V_1^* \neq V_1$, the request is considered as invalid, and the process is aborted instantly. Otherwise, the GWN generates a new random number $R_g' \in \mathbb{Z}_p^*$ and calculates $K_{GS} = h(SID_d||h(K))$, $C_1 = h(R_g'||K)$, $C_2 = h(ID_u||R_u||C_1)$, $M_2 = C_2 \oplus K_{GS}$ and $V_2 = h(C_2||K_{GS})$. Finally, GWN dispatches the authentication request message $\langle M_2, V_2 \rangle$ to the accessed smart device SD via open channel.
- **Step 3.** On receiving the message $\langle M_2, V_2 \rangle$, SD calculates $C_2 = M_2 \oplus K_{GS}$. If C_2 is fresh, SD calculates $V_2^* = h(C_2||K_{GS})$. If $V_2^* \neq V_2$, the request is considered as failed, and it is then aborted. On the other side, SD picks a random number $R_d \in \mathbb{Z}_p^*$, computes the session key $SK = h(C_2||R_d||SID_d)$ shared with U, $M_3 = (R_d||h(SK)) \oplus K_{GS}$ and $V_3 = h(R_d||K_{GS}||h(SK))$. Next, SD transmits the authentication reply message $\langle M_3, V_3 \rangle$ to GWN via public channel.
- **Step 4.** On receiving the message $\langle M_3, V_3 \rangle$ from SD, GWN computes $(R_d||h(SK)) = M3 \oplus K_{GS}$. If R_d is also fresh, the GWN continues to calculate $V_3^* = h(R_d||K_{GS}||h(SK))$. If $V_3^* \neq V_3$, the request is considered as invalid and the process is aborted immediately. Otherwise, the GWN generates another random number $DID_u' \in \mathbb{Z}_p^*$ and computes $M_4 = (DID_u'||C_1||R_d) \oplus K_{UG}$, $K_{UG}' = h(ID_u||C_1)$ and $V_4 = h(DID_u'||C_1||R_d||K_{UG}')$. GWN then updates the tuple $\langle DID_u', ID_u, R_g' \rangle$ in its *user_data* table, and sends the ackowledgement message $\langle M_4, V_4 \rangle$ to the U via open channel.
- **Step 5.** On receiving the message $\langle M_4, V_4 \rangle$ from GWN, the user U recovers $(DID_u'||C_1||R_d) = M4 \oplus K_{UG}$, and then computes $K_{UG}' = h(ID_u||C_1)$ and $V_4^* = h(DID_u'||C_1||R_d||K_{UG}')$. If $V_4^* \neq V_4$, the login is considered as failed one and it is aborted immediately. Otherwise, the user U computes the session key $SK = h(h(ID_u||R_u||C_1)||R_d||SID_d)$ and the updated values for $A_1' = (R_u||DID_u') \oplus h(ID_u||PW_u||\sigma_u)$, $A_2' = h(DID_u||\sigma_u||PW_u)$, $A_3' = K_{UG}' \oplus h(DID_u||PW_u||\sigma_u)$. Finally, U resets $TEMP$ to 0 as $TEMP' = 0$, and updates the smart card SC_u with the values $\{A_1', A_2', A_3', TEMP'\}$ by replacing the old values $\{A_1, A_2, A_3, TEMP\}$.

  The login and authentication phase is finally briefed in Figure 3.

Remark 1. *An adversary might block the message $\langle M_4, V_4 \rangle$ during the communication happen in the login and authentication phase. As DID_u and R_g have already been updated on the gateway node GWN, the subsequent login attempts by the user U will fail. This attack can be prevented, if the gateway node GWN also maintains the old values of DID_u and R_g until the next successful authentication happens.*

User U	Gateway node GWN	Smart device SD
$\{SC_u = \langle A_1, A_2, A_3, \tau_u, TEMP\rangle\}$	$\{K\}$	$\{K_{GS}\}$
Enter ID_u, PW_u and Bio_u. Compute $\sigma_u = rep(Bio_u, \tau_u)$, $DID_u = A_1 \oplus h(ID_u \| \| PW_u \| \| \sigma_u)$, $A_2^* = h(DID_u \| \| ID_u \| \| \sigma_u \| \| PW_u)$. If $A_2^* \neq A_2$, set $TEMP = TEMP + 1$ and abort. Generate $R_u \in \mathbb{Z}_p$ and compute $K_{UG} = A_3 \oplus h(ID_u \| \| DID_u \| \| PW_u \| \| \sigma_u)$, $M_1 = (R_u \| \| SID_d) \oplus K_{UG}$, $V_1 = h(ID_u \| \| R_u \| \| K_{UG} \| \| M_1)$. $\langle DID_u, M_1, V_1\rangle \rightarrow$	Look up ID_u, R_g using DID_u. Compute $K_{UG} = h(ID_u \| \| h(R_g \| \| K))$, $(R_u \| \| SID_d) = M_1 \oplus K_{UG}$, $V_1^* = h(ID_u \| \| R_u \| \| K_{UG} \| \| M_1)$. If $V_1^* \neq V_1$, abort. Generate $R_g' \in \mathbb{Z}_p^*$ and compute $K_{GS} = h(SID_d \| \| h(K))$, $C_1 = h(R_g' \| \| K)$,	
	Compute $C_2 = h(ID_u \| \| R_u \| \| C_1)$, $M_2 = C_2 \oplus K_{GS}$, $V_2 = h(C_2 \| \| K_{GS})$. $\langle M_2, V_2\rangle \rightarrow$	Compute $C_2 = M_2 \oplus K_{GS}$, $V_2^* = h(C_2 \| \| K_{GS})$. If $V_2^* \neq V_2$, abort. Generate $R_d \in \mathbb{Z}_p^*$ and compute $SK = h(C_2 \| \| R_d \| \| SID_d)$, $M_3 = (R_d \| \| h(SK)) \oplus K_{GS}$, $V_3 = h(R_d \| \| K_{GS} \| \| h(SK))$. $\langle M_3, V_3\rangle \leftarrow$
Compute $(DID_u' \| \| C_1 \| \| R_d) = M4 \oplus K_{UG}$, $K_{UG}' = h(ID_u \| \| C_1)$, $V_4^* = h(DID_u' \| \| C_1 \| \| R_d \| \| K_{UG}')$. If $V_4^* \neq V_4$, abort. Compute $SK = h(h(ID_u \| \| R_u \| \| C_1) \| \| R_d \| \| SID_d)$, $A_1' = (R_u \| \| DID_u') \oplus h(ID_u \| \| PW_u \| \| \sigma_u)$, $A_2' = h(DID_u \| \| \sigma_u \| \| PW_u)$, $A_3' = K_{UG}' \oplus h(DID_u \| \| PW_u \| \| \sigma_u)$. Set $TEMP' = 0$. Update $A_1', A_2', A_3', TEMP'$ in SC_u.	If $V_3^* \neq V_3$, abort. Generate $DID_u' \in \mathbb{Z}_p^*$. Compute $M_4 = (DID_u' \| \| C_1 \| \| R_d) \oplus K_{UG}$, $K_{UG}' = h(ID_u \| \| C_1)$, $V_4 = h(DID_u' \| \| C_1 \| \| R_d \| \| K_{UG}')$. Update tuple $\langle DID_u', ID_u, R_g'\rangle$ $\langle M_4, V_4\rangle \leftarrow$.	

Figure 3. Summary of login and authentication phase.

4.4. Password and Biometric Update Phase

To update "password and/or biometric", a registered user U inputs identity ID_u along with the existing password PW_iu and imprints biometric Bio_u, and then logins with the steps similar to that described in the "login and authentication phase" discussed in Section 4.3.

If the login is successful, U provides new password PW_u', imprints new biometric Bio_u' and recalculates $(\sigma_u', \tau_u') = Gen(Bio_u')$. Next, U computes $A_1' = DID_u \oplus h(ID_u \| \| PW_u' \| \| \sigma_u')$, $A_2' = h(DID_u \| \| ID_u \| \| \sigma_u' \| \| PW_u')$ and $A_3' = K_{UG} \oplus h(ID_u \| \| DID_u \| \| PW_u' \| \| \sigma_u')$, and replace $\{A_1, A_2, A_3, \tau_u\}$ in the smart card SC_u with $\{A_1', A_2', A_3', \tau_u'\}$. SC_u now contains the updated credentials $\{A_1', A_2', A_3', \tau_u', TEMP\}$.

4.5. Smart Card Revocation Phase

A "lost or stolen smart card" can be revoked by requesting for a new smart card by a registered authorized user U to the registration authority RA via secure channel. Hence, the steps are identical to those for the mobile user registration phase as discussed in Section 4.2.2.

5. Security Analysis

In this section, through the widely accepted "Real-Or-Random (ROR) model" [27], the formal security analysis of the proposed scheme is presented. Furthermore, through the formal security verification tool, called AVISPA [28], the proposed scheme's resistance to "man-in-the-middle and replay attacks" is verified. In addition, a through informal (non-mathematical) analysis presented in Section 5.3 demonstrates the proposed scheme's resistance to various other known attacks.

5.1. Formal Security Analysis through Real-Or-Random Model

The ROR model proposed in [27] is widely accepted for security analysis of authentication and key agreement schemes. We describe the ROR model and then utilize the same to analyze the proposed scheme formally.

- **Participants:** Let the oracles π_U^u, π_{SD}^d and π_{GWN}^g denote the uth, dth and gth instances of a user U, a smart device SD and the gateway node GWN, respectively.
- **Partnering:** Two oracles π_U^u and π_{SD}^d are said to be partnered provided they share the same communication session-id sid, and the partial transcript of the exchanged messages is unique.
- **Freshness:** π_U^u and π_{SD}^d are considered fresh as long as the session key SK between U and SD remains unexposed to an adversary $\mathcal{A}$.
- **Adversary:** The ROR model defines the DY adversary $\mathcal{A}$. Formally, the adversary $\mathcal{A}$ can execute the queries described below.
  - $Execute(\pi^u, \pi^d)$: This query is modeled as an eavesdropping attack. Therefore, this query allows $\mathcal{A}$ to intercept the messages exchanged among U, SD, and GWN.
  - $Send(\pi^d, m)$: This query is modeled an active attack. It allows $\mathcal{A}$ to transmit a message, say msg to an oracle π^d, and receive the response message in reply.
  - $CorruptSC(\pi^u)$: Through this query, $\mathcal{A}$ can learn the confidential values $\{A_1, A_2t, A_3, \tau_u, TEMP\}$ from a user U's smart card SC_u.
  - $CorruptSD(\pi^d)$: Through this query, $\mathcal{A}$ can learn the secret key $\{K_{GS}\}$ stored in the captured smart device SD. The queries $CorruptSC$ and $CorruptSD$ are assumed to be under a weak corruption model [29] and they can not corrupt the ephemeral keys and states of the participating oracle.
  - $Test(\pi^u, \pi^d)$: As per the "indistinguishability in the ROR model" [27], the semantic security of the session key SK between U and SD can be determined by this query. To initiate, $\mathcal{A}$ tosses an "unbiased coin" whose outcome, say c, determines the output of the $Test$ query. If SK is fresh, the oracle π^u or π^d produces SK, if $c = 1$. Otherwise, if $c = 0$, the oracle produces a random number. In all other cases, the returned value will be null.
- **Semantic security of the session key:** As per the ROR model, to compromise the semantic security of the session key, $\mathcal{A}$ must be able to differentiate an instance's actual session key from a random key. $\mathcal{A}$ can perform a limited number of $CorruptSC(\pi^u)$ and $CorruptSD(\pi^d)$ queries, but can execute as many $Test(\cdot)$ queries as desired.

  If $Adv_{PS,\mathcal{A}}(t)$ represents the advantage of $\mathcal{A}$ in compromising the semantic security of the proposed scheme PS, we have, $Adv_{PS,\mathcal{A}}(t) = |2.Pr[SCS] - 1|$, where SCS is an event of $\mathcal{A}$'s success.
- **Random oracle:** All participating entities including $\mathcal{A}$ can invoke the "cryptographic one-way hash function", $h(\cdot)$, which is further modeled as a random oracle, say $\mathcal{HO}$.

Accounting to Wang et al.'s important findings [30] regarding the Zipf's law on passwords, Theorem 1 defines the "semantic security of the proposed scheme".

Theorem 1. *Let a polynomial time adversary $\mathcal{A}$ attempts to break the semantic security of the proposed scheme $\mathcal{P}$ under the ROR model in time t. If the chosen passwords follow the Zipf's law [30], and the bit-lengths of the biometric secret key σ_u and the user identity ID_u are l_1 and l_2, respectively, $\mathcal{A}$'s advantage in compromising the semantic security of the proposed scheme PS is*

$$Adv_{PS,\mathcal{A}}(t) \leq \frac{q_h^2}{|Hash|} + 2\max\{C'.q_s^{s'}, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\},$$

where q_h, q_s and $|Hash|$ represent the number of hash queries, the number of Send queries and the range of $h(\cdot)$, respectively, and C' and s' are the Zipf's parameters [30].

Proof. We design our proof on the lines of the proofs that presented in [11,31,32]. Four sequential games, say G_i, $i \in [0-3]$, are played. The event SCS_i represents that an adversary $\mathcal{A}$ can successfully guess the bit c in the game G_i. The details regarding all the games are given below.

- **Game** G_0: This game models a real attack on the semantic security of the proposed scheme PS by $\mathcal{A}$. As initially the bit c is guessed,

$$Adv_{PS,\mathcal{A}}(t) = |2.Pr[SCS_0] - 1|. \tag{2}$$

- **Game** G_1: This game models as an eavesdropping attack by $\mathcal{A}$ on PS. Through the $Execute(\pi^u, \pi^d)$ query, $\mathcal{A}$ can intercept the messages $\langle DID_u, M_1, V_1\rangle$, $\langle M_2, V_2\rangle$, $\langle M_3, V_3\rangle$ and $\langle M_4, V_4\rangle$. $\mathcal{A}$ can query the *Test* oracle and attempt to determine if the received result is the actual session key. As the session key is $SK = h(h(ID_u||R_u||h(R'_g||K))||R_d||SID_d)$, and to compute the same $\mathcal{A}$ must learn short term secret keys (R_u, R'_g and R_d) as well as long term secrets (ID_u, SID_d and K). Therefore, $\mathcal{A}$ gains no additional advantage for wining this game. Consequently, it follows that

$$Pr[SCS_1] = Pr[SCS_0]. \tag{3}$$

- **Game** G_2: This game models as an active attack through use of the *Send* and hash queries. $\mathcal{A}$ attempts to beguile a legitimate entity into accepting a modified message. As discussed previously, $\mathcal{A}$ can repeat the queries to the oracles in order to induce hash collisions. However, since all the messages contain random nonces, hash coalitions cannot be induced on $h(\cdot)$ by $\mathcal{A}$. It is worth noticing that both the games G_1 and G_2 are identical except for the *Send* and hash queries in the game G_2. Thus, through the use of birthday paradox, we have,

$$|Pr[SCS_2] - Pr[SCS_1]| \leq \frac{q_h^2}{2|Hash|}. \tag{4}$$

- **Game** G_3: An extension to G_2, the game G_3 is the final game and it simulates the $CorruptSC$ and $CorruptSD$ queries. Querying these oracles, $\mathcal{A}$ can learn $\{A_1, A_2t, A_3, \tau_u, TEMP\}$ and $\{K_{GS}\}$, respectively. The probability of $\mathcal{A}$ to correctly guess the biometric secret key σ_i of bit-length l_1 and the user identity ID_u of bit-length l_2 are $\frac{1}{2^{l_1}}$ and $\frac{1}{2^{l_2}}$, respectively [33].

  As the user chosen passwords tend to follow the Zipf's law, by utilizing trawling guessing attacks, $\mathcal{A}$'s advantage will be over 0.5 when $q_s = 10^7$ or 10^8 [30]. If $\mathcal{A}$ can utilize a user's personal information for the targeted guessing attacks, he/she will have an advantage over 0.5 when $q_s \leq 10^6$ [30]. In practical implementation, only a finite number of erroneous password attempts are permitted to the adversary $\mathcal{A}$. Therefore, the games G_3 and G_2 are identical except for the guessing attacks. Thus, we can formulate the following relation as in [32]:

$$|Pr[SCS_3] - Pr[SCS_2]| \leq \max\{C'.q_s^{s'}, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\}. \tag{5}$$

  However, $\mathcal{A}$ must guess a bit c' after executing the *Test* query to win the game G_3. Therefore, it follows that

$$|Pr[SCS_3] = \frac{1}{2}. \tag{6}$$

  From Equations (2), (3) and (6), we have,

$$\begin{aligned} \frac{1}{2}Adv_{PS,\mathcal{A}}(t) &= |Pr[SCS_0] - \frac{1}{2}| \\ &= |Pr[SCS_1] - \frac{1}{2}| \\ &= |Pr[SCS_1] - |Pr[SCS_3]|. \end{aligned} \tag{7}$$

Summing the inequalities from Equations (4) and (5), we obtain the following relation:

$$|Pr[SCS_2] - |Pr[SCS_1]|| + |Pr[SCS_3] - |Pr[SCS_2]|| \leq \frac{q_h^2}{2|Hash|} + \max\{C'.q_s^{s'}, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\}. \quad (8)$$

Simultaneously solving Equations (7) and (8), we arrive at the desired result:

$$Adv_{PS,\mathcal{A}}(t) \leq \frac{q_h^2}{|Hash|} + 2\max\{C'.q_s^{s'}, \frac{q_s}{2^{l_1}}, \frac{q_s}{2^{l_2}}\}.$$

□

5.2. Formal Security Verification through AVISPA Simulation

AVISPA is an automated software tool for the formal verification of security-sensitive protocols and applications [28]. AVISPA implements the Dolev-Yao (DY) threat model and verifies whether a scheme is resistant to replay and man-in-the-middle attacks. A security protocol to be verified needs to be modeled in the associated "High Level Protocol Specification Language (HLPSL)" [34]. AVISPA provides a translator, known as HLPSL2IF, for translating HLSPL into the Intermediate Format (IF). The IF can be interpreted by one of the available four backends to generate a report in the Output Format (OF). The structure of the OF contains following:

- SUMMARY: It states if the tested protocol is "safe", "unsafe", or if the analysis was "inconclusive".
- DETAILS: It reports the explanation relevant to the SUMMARY section.
- PROTOCOL: It provides the protocol to be verified.
- GOAL: It states the goal as specified in the HLPSL.
- BACKEND: It mentions the backend that has been utilized.
- STATISTICS: It provides the trace for the vulnerabilities to the target protocol, if they are present, with additional useful statistics.

A more detailed report on AVISPA and HLPSL is available at in [28]. The four backends available with AVISPA are [28]: (a) "On-the-fly Model-Checker (OFMC)", (b) "Constraint-Logic-based Attack Searcher (CL-AtSe)", (c) "SAT-based Model-Checker (SATMC)", and (d) "Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)". Among these, OFMC and CL-AtSe are most widely accepted, and we evaluate the proposed scheme under these backends to formally verify its resistance to the "man-in-the-middle and replay attacks".

We have implemented the proposed scheme in HLSPL and defined the necessary roles for a user U, a smart device SD, and the GWN for the different phases of the proposed scheme. We have also specified the roles for the session, goal, and environment as per the HLPSL specification. Finally, we have simulated the proposed scheme using the "SPAN, the Security Protocol ANimator for AVISPA tool'' [35]. Figure 4 presents the simulation results under the widely-used OFMC and CL-AtSe backends. The simulation results clearly demonstrate that the proposed scheme is safe against the "man-in-the-middle and replay attacks".

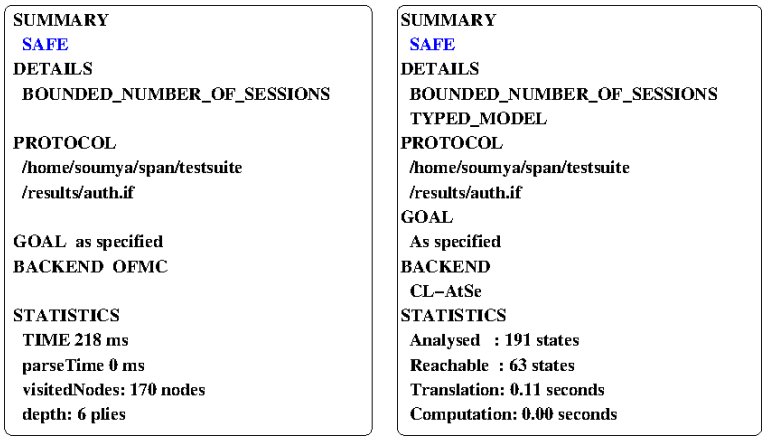

Figure 4. The simulation results under OFMC & CL-AtSe back-ends.

5.3. Informal Security Analysis

In the following, we demonstrate that the proposed scheme is secure against various known attacks.

5.3.1. Replay Attack

Assuming an adversary $\mathcal{A}$ replays the old message M_1 to GWN, GWN will reject the replayed message after it detects that R_u is not fresh. Similarly, all messages are composed of random nonces, which can be further verified for their freshness. Thus, the proposed scheme is resilient against replay attack.

5.3.2. Forgery Attack

An adversary $\mathcal{A}$ can attempt to forge the message $\langle DID_u, M_1, V_1 \rangle$ to the GWN. However, M_1 is encrypted with the secret key K_{UG}, and V_1 is also encapsulated with DID_U and M_1 against forgery. $\mathcal{A}$ cannot forge this message. Similarly, other messages cannot be forged either, and the proposed scheme is resilient against forgery attack.

5.3.3. Impersonation Attack

Assuming an adversary $\mathcal{A}$, after capturing the messages from a successful login an authentication attempts, to impersonate the user U. But, as DID_u is of single-use and V_1 encapsulates ID_U and M_1 against forgery, $\mathcal{A}$ cannot simply modify the captured messages with his/her own R_u to impersonate U. Similarly, $\mathcal{A}$'s attempt to impersonate the GWN will fail because he/she will be unable to generate $\langle M_2, V_2 \rangle$ and $\langle M_4, V_4 \rangle$ without the knowledge of K_{GS} and K_{UG}, respectively. As a result, the proposed scheme is resilient against impersonation attacks.

5.3.4. Man-in-the-Middle Attack

Assuming an adversary $\mathcal{A}$ attempts to execute a man-in-the-middle attack by capturing and modifying the login message from U to GWN. Nevertheless, the message cannot be forged or modified without knowledge of the secret credentials. Thus, the "man-in-the-middle attack" is also protected in the proposed scheme.

5.3.5. Loss of Smart Card and Offline Guessing Attack

Assuming an adversary $\mathcal{A}$ recovers a lost smart card, he/she can learn the values A_1, A_2, A_3, τ_u and $TEMP$ through the "power analysis attacks". Of these, except for $TEMP$ and τ_u, none is in plaintext and it is combination of the secret identity, password, and biometrics. It is worth noticing that τ_u and $TEMP$ are the public reconstruction parameter for biometrics and failed login attempts counter, respectively, which are not sensitive. For $\mathcal{A}$ to subvert the proposed scheme through the offline

guessing attack, he/she will have to simultaneously guess ID_u, PW_u, and σ_u, which is "computationally infeasible" task. Thus, the proposed scheme is resilient against the "loss of smart card and offline guessing attacks".

5.3.6. Privileged-Insider Attack

Assuming an adversary $\mathcal{A}$ is a privileged-insider, he/she can eavesdrop during the registration phase and learn user identity ID_u. Now, assume that he/she has subverted the user's smart card SC_u to recover the stored values $A_1 = DID_u \oplus h(ID_u||PW_u||\sigma_u)$, $A_2 = h(DID_u||ID_u||\sigma_u||PW_u)$ and $A_3 = K_{UG} \oplus h(ID_u||DID_u||PW_u||\sigma_u)$. It is clear that even if ID_u is known, in order to subvert the scheme with the available information, $\mathcal{A}$ must simultaneously guess password PW_u and biometric secret key σ_u, which is computationally infeasible. As a result, the privileged-insider attack is protected in the proposed scheme.

5.3.7. Ephemeral Secret Leakage (ESL) Attack

Assume adversary $\mathcal{A}$ learns one or both of the session specific secrets (R_u, R_g, R_d) through the session hijacking attack under the CK-adversary model. Since the session key $SK = h(h(ID_u||R_u||C_1)||R_d||SID_d)$ is derived from the user secret identity ID_u and the GWN's long term secret of K in addition to (R_u, R_g, R_d), $\mathcal{A}$ cannot subvert the session key SK without any long term secrets. Thus, the proposed scheme is secure against ESL attack.

5.3.8. Parallel Session Attack

For an adversary $\mathcal{A}$ to successfully execute a parallel session attack, he/she needs to compose the session key $SK = h(h(ID_u||R_u||C_1)||R_d||SID_d)$ by eavesdropping on the authentication related messages. But, no secrets are compromised regardless of lost smart card attack or privileged insider attack. As a result, the proposed scheme is secure against a parallel session attack.

5.3.9. Stolen Verifier Attack

As the gateway node GWN maintains the tuple $\langle DID_u, ID_u, R_g \rangle$ for each user U. Of these, DID_u and R_g are the distict random nonces. Exposure of ID_u is equivalent to a privileged-insider attack. However, the proposed scheme is resistant against privileged-insider attack. Thus, a stolen verifier attack is not a threat to the proposed scheme.

5.3.10. Smart Card Impersonation Attack

Smart card impersonation attack can only be executed by an adversary $\mathcal{A}$, if he/she can learn the secret values ID_u, PW_u and σ_u in a user's smart card. Nevertheless, the secret values are not compromised through a lost smart card even in the presence of a privileged insider attacker. The proposed scheme is then secure against smart card impersonation attack.

5.3.11. Anonymity and Untracability

Assume that an adversary $\mathcal{A}$ eavesdrops and monitors the messages from a successful login and authentication. None of the eavesdropped values $\{DID_u, M_1, M_2, M_3, M_4, V_1, V_2, V_3, V_4\}$, contains any plaintext information useful for identifying the user U or the smart device SD. Thus, the proposed scheme provides anonymity. Furthermore, all of the eavesdropped values are composed of some random nonces, and consequently these are always unique across different authentication sessions. Thus, the proposed scheme also provides anonymity and untracability.

6. Comparative Study

In this section, we benchmark the proposed scheme against the schemes proposed by Shuai et al. [7], Yu and Li [21], Naoui et al. [22], Fakroon et al. [23], and Dey and Hossain [24].

6.1. Communication Costs Comparison

For communication cost comparison, it is assumed that an ECC point is 320 bits, hash digest (assuming SHA-1 hashing algorithm is applied) is 160 bits, nonces as well as identities are 128 bits long. In the presented scheme, the four messages exchanged during the login and authentication phase are $\langle DID_u, M_1, V_1 \rangle$ which needs $(128 + (128 + 126) + 160) = 544$ bits; $\langle M_2, V_2 \rangle$ which requires $(160 + 160) = 320$ bits; $\langle M_3, V_3 \rangle$ which demands $((128 + 160) + 160) = 448$ bits and $\langle M_4, V_4 \rangle$ which needs $((128 + 160 + 128) + 160) = 576$ bits. Thus, the total communication overhead of the proposed scheme turns out to be $(544 + 320 + 448 + 576) = 1888$ bits $= 236$ bytes. Table 1 summarizes the proposed scheme and other existing schemes in terms of communications overheads. From this table, we observe that the proposed scheme requires less communication overhead as compared to that for the schemes of Shuai et al. [7] and second lowest among all other schemes.

Table 1. Communication costs comparison.

Scheme	No. of Bytes	No. of Messages
Shuai et al. [7]	(108 + 84 + 36 + 68) = 296	4
Yu and Li [21]	(84 + 124 + 164 + 164)×2 = 1072	8
Naoui et al. [22]	(104 + 52 + 56) = 212	3
Fakroon et al. [23]	(100 + 52 + 52 +84) = 288	4
Dey and Hossain [24]	(132 + 132 + 52 + 52 + 52) = 420	5
Proposed scheme	(68 + 40 +56 +72) = 236	4

6.2. Computation Costs Comparison

For computation cost analysis, we denote T_{bp}, T_m, T_b and T_h as the time needed for computing "bilinear pairing", "ECC multiplication", "fuzzy extractor function $Gen(\cdot)/Rep(\cdot)$ for biometric verification" and "hashing" operations, respectively. Based on experimental results reported in [36], we have $T_{bp} \approx 32.713$ ms (milliseconds), $T_m \approx 13.405$ ms, $T_b \approx T_m = 13.405$ ms and $T_h \approx 0.056$ ms, respectively. Table 2 briefs the computational costs for the proposed scheme and other existing schemes. It is clear that the presented scheme has a significantly less computation cost as compared to that for the schemes of Shuai et al. [7]. With the exception of Fakroon et al. [23], which might incur a greater computation cost, the proposed scheme has the lowest computation cost.

Table 2. Computation costs comparison.

Scheme	*U*	*GWN*	*SD*	Total Cost
Shuai et al. [7]	$6T_h + 1T_m$ ≈ 13.741 ms	$7T_h + 1T_m$ ≈ 13.797 ms	$3T_h$ ≈ 0.168 ms	$16T_h + 3T_m$ ≈ 27.604 ms
Yu and Li [21]	$7T_h + 14T_m$ ≈ 188.062ms	$12T_h + 19T_m + 4T_{bp}$ ≈ 386.219ms	$7T_h + 14T_m$ ≈ 188.062ms	$26T_h + 47T_m + 4T_{bp}$ $\approx$762.343ms
Naoui et al. [22]	$12T_h + 3T_{sym} + 2T_m$ ≈ 32.453ms	$13T_h + 4T_{sym} + 2T_m$ ≈ 34.166ms	$1T_h + 1T_{sym}$ ≈ 1.713ms	$26T_h + 7T_{sym} + 4T_m$ $\approx$68.332ms
Fakroon et al. [23]	$4T_h$ ≈ 0.224ms	$5T_h$ ≈ 0.28ms	$24T_h$ ≈ 1.344ms	$33T_h$ $\approx$1.848ms
Dey and Hossain [24]	$4Th + 2Tm + 3Tsym$ ≈ 32.005ms	- ≈ 0.0ms	$3Th + 2Tm + 3Tsym$ ≈ 31.949ms	$7Th + 4Te + 6Tsym$ $\approx$63.954ms
Proposed	$10T_h + 1T_b$ ≈ 13.965ms	$10T_h$ ≈ 0.56ms	$4T_h$ ≈ 0.224ms	$24T_h + 1T_b$ $\approx$14.749ms

6.3. Security and Functionality Features Comparison

Finally, in Table 3, the functionality of the proposed scheme and other existing schemes are compared. From this table, it is apparent that the proposed scheme provides better security and functionality features features as compared to those for other existing schemes. Moreover, from the

Tables 1 and 2, we can see that the proposed scheme requires less computation and communication overheads as compared to other schemes.

Table 3. Security & functionality features comparison.

Feature	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}
Shuai et al. [7]	☑	☑	☑	☑	☑	☒	☒	☒	☒	☒	☒
Yu and Li [21]	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Naoui et al. [22]	☑	☑	☑	☑	☑	☒	☒	☒	☒	☒	☒
Fakroon et al. [23]	☑	☑	☑	☑	☑	☒	☒	☒	☑	☑	☒
Dey and Hossain [24]	☒	☒	☒	☑	☒	☒	NA	NA	☒	NA	☒
Proposed	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑

Note: ☑: The scheme is resilient against an attack or it supports a feature; ☒: The scheme is not secure against an attack or it does not support a feature; ⊠: Discussed in text. V_1: "user anonymity", V_2: "sensor node anonymity", V_3: "untraceability", V_4: "resilience against replay attack", V_5: "resilience against man-in-the-middle attack", V_6: "resilience against ESL attack under the CK-adversary model", V_7: "resist off-line password guessing attack", V_8: "resist smart card impersonation attack", V_9: "resist parallel session attack", V_{10}: "resist password change attack", V_{11}: "support three-factor authentication".

7. Practical Impact Study through NS3 Simulation

To estimate the practicability of the proposed scheme, we have performed a simulation study. We have utilized the most recent iteration of the widely accepted network simulator tool, NS3 (3.28). We run our simulation on a Linux workstation. For our simulation, we specify the location of the gateway node (GWN) at the origin of the coordinate system. The smart devices are considered at random positions 20 to 100 m from the GWN. The users are permitted to move across a square of 150 m side centered around the gateway GWN with a maximum speed of 3 m per second. Users attempt to establish session keys with all available devices. Communication is measured across the IEEE 802.11 2.4 GHz channel. We have then simulated several scenarios with differing number of users and smart devices. The details regarding the simulation parameters are presented in Table 4. Any parameters that are not explicitly mentioned here are assumed to have their default values as defined by the NS3.

Table 4. Simulation parameters.

Parameter	Description	
Platform	NS3(3.28) / Ubuntu 16.04 LTS	
Network scenarios	No. of users	No. of smart devices
1	3	5
2	3	10
3	3	15
4	5	15
5	5	20
6	8	20
Mobility	Random (0–3 m/s)	
Simulation time	1200 s	

Figure 5a,b presents the network throughput and end-to-end delay for the proposed scheme, respectively, under different scenarios. The network throughput is calculated according to the formula:

$$\frac{N_p * |byte|}{T_{sum}},$$

whereas the end-to-end delay is computed with the formula:

$$\frac{\sum_{i=0}^{N_p}(T_{r_i} - T_{s_i})}{N_p}.$$

Here, N_p is the total number of packets received, $|byte|$ is the number of bytes in each packet, T_{sum} represents the total time taken, and T_{s_i} and T_{r_i} are the transmission and receiving time of the ith packet, respectively. The simulation results demonstrate the expected correlation between the number of participants, the network throughput and also the end-to-end delay.

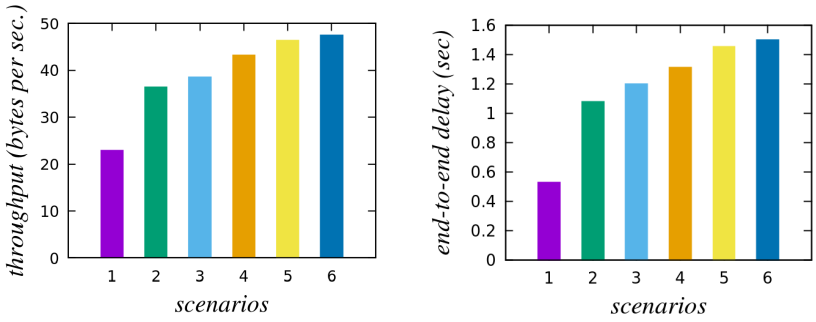

Figure 5. (**a**)Throughput (bytes per second) (**b**) End-to-end delay (seconds).

8. Conclusions

We first discussed the issue of anonymous user authentication in smart home environments. We then cryptanalyzed the recently proposed user authentication scheme and discovered its several security vulnerabilities. Furthermore, we proposed a more secure and robust authentication scheme for anonymous user authentication and key agreement in smart homes to erase the security pitfalls found in the existing Shuai et al.'s scheme, while retaining its advantages at the same time. The security analysis and performance comparison show that the proposed scheme can provide better security and more functionality features at low communication and computation overheads, when compared these with other recent existing schemes. In our future work, we plan to investigate the possibility of extending the proposed scheme to support remote registration as it is designed in the scheme proposed by Yu and Li [21] at a more acceptable communication and computation overheads.

Author Contributions: Conceptualization, S.B., A.K.D., S.C. and Y.P.; Methodology, S.B. and A.K.D.; Security analysis, S.B. and A.K.D.; Investigation, S.B., A.K.D., V.O., S.C. and Y.P.; Formal security verification, S.B. and A.K.D.; Resources, A.K.D., V.O., S.C. and Y.P.; Writing-original draft preparation, S.B.; Writing-review and editing, A.K.D. and Y.P.; Supervision, A.K.D., V.O. and Y.P.; Project administration, A.K.D., S.C. and Y.P.; Funding acquisition, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2017R1A2B1002147). This work is also supported by the Mathematical Research Impact Centric Support (MATRICS) project funded by the Science and Engineering Research Board (SERB), India (Reference No. MTR/2019/000699).

Acknowledgments: We thank the anonymous reviewers and the Editor for their valuable comments, which helped us to improve the quality and presentation of the paper.

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

References

1. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A Survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
2. Gomez, C.; Paradells, J. Wireless home automation networks: A survey of architectures and technologies. *IEEE Commun. Mag.* **2010**, *48*, 92–101. [CrossRef]

3. Kim, J.E.; Boulos, G.; Yackovich, J.; Barth, T.; Beckel, C.; Mosse, D. Seamless integration of heterogeneous devices and access control in smart homes. In Proceedings of the Eighth International Conference on Intelligent Environments (IE'12), Guanajato, Mexico, 26–29 June 2012; pp. 206–213.
4. Kumar, P.; Gurtov, A.; Iinatti, J.; Ylianttila, M.; Sain, M. Lightweight and secure session-key establishment scheme in smart home environments. *IEEE Sen. J.* **2015**, *16*, 254–264. [CrossRef]
5. Suryadevara, N.K.; Mukhopadhyay, S.C.; Wang, R.; Rayudu, R. Forecasting the behavior of an elderly using wireless sensors data in a smart home. *Eng. Appl. Artif. Intell.* **2013**, *26*, 2641–2652. [CrossRef]
6. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
7. 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]
8. Dolev, D.; Yao, A.C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
9. Messerges, T.S.; Dabbish, E.A.; Sloan, R.H. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* **2002**, *51*, 541–552. [CrossRef]
10. Canetti, R.; Krawczyk, H. Universally Composable Notions of Key Exchange and Secure Channels. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques–Advances in Cryptology (EUROCRYPT'02), Amsterdam, The Netherlands, 28 April–2 May 2002; pp. 337–351.
11. Banerjee, S.; Odelu, V.; Das, A.K.; Jangirala, S.; Kumar, N.; Chattopadhyay, S.; Choo, K.K.R. A Provably-Secure and Lightweight Anonymous User Authenticated Session Key Exchange Scheme for Internet of Things Deployment. *IEEE Internet Things J.* **2019**, *6*, 8739–8752. [CrossRef]
12. Jeong, J.; Chung, M.Y.; Choo, H. Integrated OTP-based user authentication scheme using smart cards in home networks. In Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS'08), Waikoloa, HI, USA, 7–10 January 2008; p. 294.
13. 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]
14. Kim, H.J.; Kim, H.S. AUTH HOTP-HOTP based authentication scheme over home network environment. In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA'11), Santander, Spain, 20–23 June 2011; pp. 622–637.
15. Vaidya, B.; Makrakis, D.; Mouftah, H.T. Device authentication mechanism for smart energy home area networks. In Proceedings of the IEEE International Conference on Consumer Electronics (ICCE'11), Berlin, Germany, 9–12 January 2011; pp. 787–788.
16. Hanumanthappa, P.; Singh, S. Privacy preserving and ownership authentication in ubiquitous computing devices using secure three way authentication. In Proceedings of the International Conference on Innovations in Information Technology (IIT'12), Abu Dhabi, UAE, 18–20 March 2012; pp. 107–112.
17. Li, Y. Design of a key establishment protocol for smart home energy management system. In Proceedings of the Fifth International Conference on Computational Intelligence, Communication Systems and Networks (CICSYN'13), Madrid, Spain, 5–7 June 2013; pp. 88–93.
18. Han, K.; Kim, J.; Shon, T.; Ko, D. A novel secure key paring protocol for RF4CE ubiquitous smart home systems. *Pers. Ubiquitous Comput.* **2013**, *17*, 945–949. [CrossRef]
19. Santoso, F.K.; Vun, N.C. Securing IoT for smart home system. In Proceedings of the International Symposium on Consumer Electronics (ISCE'15), Madrid, Spain, 9–11 April 2015; pp. 1–2.
20. Wazid, M.; Das, A.K.; Odelu, V.; Kumar, N.; Susilo, W. Secure remote user authenticated key establishment protocol for smart home environment. *IEEE Trans. Depend. Secur. Comput.* **2017**, doi:10.1109/TDSC.2017.2764083. [CrossRef]
21. Yu, B.; Li, H. Anonymous authentication key agreement scheme with pairing-based cryptography for home-based multi-sensor Internet of Things. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1–11. [CrossRef]
22. Naoui, S.; Elhdhili, M.H.; Saidane, L.A. Novel Smart Home Authentication Protocol LRP-SHAP. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'19), Marrakech, Morocco, 15–18 April 2019; pp. 1–6.
23. Fakroon, M.; Alshahrani, M.; Gebali, F.; Traore, I. Secure remote anonymous user authentication scheme for smart home environment. *Internet Things* **2020**, *9*, 1–20. [CrossRef]

24. Dey, S.; Hossain, A. Session-key establishment and authentication in a smart home network using public key cryptography. *IEEE Sen. Lett.* **2019**, *3*, 1–4. [CrossRef]
25. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Annual International Cryptology Conference (CRYPTO'99), Santa Barbara, CA, USA, 15–19 August 1999; pp. 388–397.
26. Dodis, Y.; Reyzin, L.; Smith, A. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques–Advances in Cryptology (EUROCRYPT'04), Lecture Notes in Computer Science (LNCS), Interlaken, Switzerland, 2–6 May 2004; Volume 3027, pp. 523–540.
27. Abdalla, M.; Fouque, P.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In Proceedings of the 8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Lecture Notes in Computer Science (LNCS), Les Diablerets, Switzerland, 23–26 January 2005; Volume 3386, pp. 65–84.
28. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 23 March 2019).
29. Chang, C.C.; Le, H.D. A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 357–366. [CrossRef]
30. 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]
31. Gope, P.; Das, A.K.; Kumar, N.; Cheng, Y. Lightweight and Physically Secure Anonymous Mutual Authentication Protocol for Real-Time Data Access in Industrial Wireless Sensor Networks. *IEEE Trans. Ind. Inf.* **2019**, *15*, 4957–4968. [CrossRef]
32. Banerjee, S.; Odelu, V.; Das, A.K.; Chattopadhyay, S.; Rodrigues, J.J.; Park, Y. Physically Secure Lightweight Anonymous User Authentication Protocol for Internet of Things Using Physically Unclonable Functions. *IEEE Access* **2019**, *7*, 85627–85644. [CrossRef]
33. Odelu, V.; Das, A.K.; Goswami, A. A Secure Biometrics-Based Multi-Server Authentication Protocol Using Smart Cards. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1953–1966. [CrossRef]
34. von Oheimb, D. The high-level protocol specification language hlpsl developed in the eu project avispa. In Proceedings of the 3rd APPSEM II (Applied Semantics II) Workshop (APPSEM'05), Frauenchiemsee, Germany, 12–15 September 2005; pp. 1–17.
35. AVISPA. SPAN, the Security Protocol ANimator for AVISPA. 2019. Available online: http://www.avispa-project.org/ (accessed on 23 March 2019).
36. Wu, L.; Wang, J.; Choo, K.R.; He, D. Secure Key Agreement and Key Protection for Mobile Device User Authentication. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 319–330. [CrossRef]