This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2017.2764083, IEEE Transactions on Dependable and Secure Computing

Secure Remote User Authenticated Key Establishment Protocol for Smart Home Environment

Mohammad Wazid, *Student Member, IEEE*, Ashok Kumar Das, *Member, IEEE*, Vanga Odelu, Neeraj Kumar, *Member, IEEE*, and Willy Susilo, *Senior Member, IEEE*

Abstract—The Information and Communication Technology (ICT) has been used in wide range of applications, such as smart living, smart health and smart transportation. Among all these applications, smart home is most popular, in which the users/residents can control the various smart sensor devices of home by using the ICT. However, the smart devices and users communicate over an insecure communication channel, i.e., the Internet. There might be the possibility of various types of attacks, such as smart device capture attack, user, gateway node and smart device impersonation attacks and privileged-insider attack on a smart home network. An illegal user, in this case, can gain access over data sent by the smart devices. Most of the existing schemes reported in the literature for the remote user authentication in smart home environment are not secure with respect to the above specified attacks. Thus, there is need to design a secure remote user authentication scheme for a smart home network so that only authorized users can have access to the smart devices. To mitigate the aforementioned isses, in this paper, we propose a new secure remote user authentication scheme for a smart home environment. The proposed scheme is efficient for resource-constrained smart devices with limited resources as it uses only one-way hash functions, bitwise XOR operations and symmetric encryptions/decryptions. The security of the scheme is proved using the rigorous formal security analysis under the widely-accepted Real-Or-Random (ROR) model. Moreover, the rigorous informal security analysis and formal security verification using the broadly-accepted Automated Validation of Internet Security Protocols and Applications (AVISPA) tool is also done. Finally, the practical demonstration of the proposed scheme is also performed using the widely-accepted NS-2 simulation.

Index Terms—Smart home, user authentication, key agreement, provable security, AVISPA, NS2 simulation.

1 Introduction

The advancement of ICT and the Internet have provided the support for rapid growth in smart home environments. A smart home contains the advanced automation systems for monitoring and controlling of various smart devices. In a smart home, the residents can control various smart sensing devices such as temperature monitoring sensors, lighting equipments sensors, or occupancy sensors, etc. [1], [2], [3], [4]. The smart home environment provides a high level of comfort with reduced operational costs to provide safety and security to its residents [5]. One of the major advantages of this type of environment is for the elderly and disabled people in which these people get assistance in estimating their body parameters using smart gadgets [6]. A smart home is equipped with a number of smart devices $(SD_j\text{s})$, such as low-cost sensors, smart light controllers, smart window shutters, smart AC controllers various and surveillance cameras. Most of the SD_js are resource-constrained having limited computational and communication power, and limited battery backup [5]. A smart home network can be implemented with the help of these SD_js in which all SD_js communicate over wireless channels using the home gateway node (GWN). The GWN acts as a bridge between SD_js and smart home user (U_i). The GWN provides interoperability and control for the SD_js and connects them to the external world using the Internet. This facilitates the U_is to operate the smart home appliances remotely using the Internet-enabled smartphones, tablets, etc. anytime from anywhere in the world [5], [7].

1.1 Network Model

The network model depicted in Fig. 1 consists of the smart home users U_is who want to access smart devices SD_js as per their requirements. Suppose there is a user U_i, who wants to access certain SD_j (e.g. temperature & humidity sensor). To access that SD_j, U_i first needs to register himself/herself at the trusted registration authority RA. Similarly, all SD_js and the gateway node GWN (which acts as the bridge between the SD_j and U_i, and connects SD_j to the external world using the Internet) are also registered at the RA. The GWN is thus a special node

- *M. Wazid is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India (e-mail: mohammad.wazid@research.iiit.ac.in).*
- *A. K. Das is with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India (e-mail: iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in).*
- *V. Odelu is with the Department of Computer Science and Engineering, Indian Institute of Information Technology, Sri City, Chittoor 517 588, Andhra Pradesh, India (e-mail: odelu.vanga@gmail.com).*
- *N. Kumar is with the Department of Computer Science and Engineering, Thapar University, Patiala 147004, India (e-mail: neeraj.kumar@thapar.edu).*
- *W. Susilo is with the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2500, Australia (e-mail: wsusilo@uow.edu.au).*

that takes responsibility of controlling the network data, device and network interoperability and secure management [5]. The registration authority (RA) is a trusted server and it is responsible for registering all the smart devices, users U_i's and the GWN securely. After the successful registration of U_i, SD_j and GWN securely, the RA stores this useful information in the memory of smart phone SP_i of U_i, and also in the memory of SD_j and GWN, which are further used at the time of authentication and key establishment process. U_i, who wants to access a SD_j, sends an authentication request directly to the GWN as both of them have already performed the registration phase at the RA. Three categories of mutual authentications happen: 1) between U_i and GWN, 2) between GWN and SD_j and 3) between U_i and SD_j. Moreover, U_i and SD_j establish a secret session key SK_{ij} between them to protect the exchanged messages.

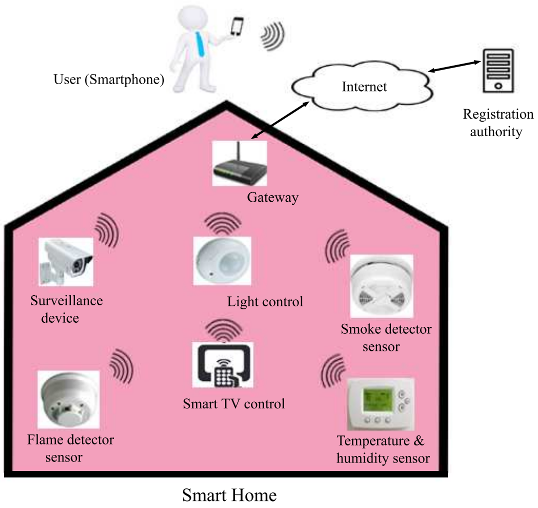


Fig. 1. Smart home environment (Adapted from [5])

1.2 Motivation

Consider the following scenario in smart home environment [8]. Recently, it is noticed that the major trend throughout Europe is the aging society, which is affected by an increasing life expectancy and decreasing birth rates. A large proportion of the European society will be not only from the group of people over 65, but also from a significant increase in the number of people over 80. The proportion of population aged over 65 and over is rising in all countries, however differences can be observed. It is also reported that “the ratio for Iceland, Ireland, Slovak Republic and Turkey lie well below the average for Europe, whereas the ratio for Finland, Germany, Greece, Italy and Sweden lie far above the average for Europe” [8].

The SD_js in smart homes communicate over the insecure communication channels. There might be the possibility of various attacks in a smart home network. An illegal user (attacker), who can monitor the activities in a smart home, can break the security, and also can gain access over the SD_js and other smart home appliances. For example, the attacker can watch the activities in the home by accessing the surveillance camera illegally where disabled people live in the smart home. Most of the existing authentication schemes reported in the literature in a smart home environment are not secure against various known attacks, such as smart device capture attack, user, gateway node and smart device impersonation attacks, and privileged-insider attack. Most of those schemes also fail to preserve traceability and anonymity properties of the users, the GWN as well as of the smart devices SD_js. Moreover, using the smart phone stolen attack, it is possible that an adversary $\mathcal{A}$ can capture a user's secret credentials, such as identity, password and biometrics key with the help of the extracted information stored in the smart phone. In addition, with the help of the user, gateway node and smart device impersonation attacks, $\mathcal{A}$ can create valid messages on the behalf of a user U_i, GWN and smart device SD_j, respectively, and can send the corresponding messages to U_i, GWN and SD_j so that these messages are treated as valid by U_i, GWN and SD_j, respectively. In a privileged-insider attack, an insider user of the RA can act as an adversary. The privileged-insider of the RA being an adversary can use the registration information of the users sent to the RA by a legal U_i during the registration phase and derive user's secret credentials, such as identity, password and biometrics key. However, the GWN registration is usually performed in offline mode securely by the RA, and hence, an adversary can not compromise the sensitive information stored in the tamper-resistant GWN device. Considering various possible attacks in a smart home environment, there is a great need to design a secure remote user authentication scheme suitable for a smart home network so that only authorized users can access the information collected by the deployed SD_js.

1.3 Threat Model

- We have used the Dolev-Yao threat model [9] in our scheme. According to this model, any two communicating parties communicate over an insecure channel and the end-point entities such as U_i and SD_j are not considered as trusted entities. An adversary, say $\mathcal{A}$, can eavesdrop the exchanged messages, and also can modify or delete the message contents during transmission.
- It is assumed that an adversary can physically capture some smart devices equipped at the smart home which are not tamper-resistant, and can extract all the sensitive data stored in those devices.
- As in [5], we also assume that the GWN is fully trusted and can not be compromised by an adversary. Otherwise, the whole network is compromised if the GWN is compromised. For this purpose, as in Bertino *et al.*'s scheme [10], we also assume that the GWN is equipped with the tamper-resistant device so that all the sensitive information including the cryptographic keying materials stored in it is protected from $\mathcal{A}$. Hence, the use of a tamper-resistant GWN makes the security of the proposed scheme is strong enough. Though the attacks on tamper-resistant devices are possible, the attacker $\mathcal{A}$ needs a special equipment to perform attacks to extract the information. Since it is cheaper to install the GWN than the special equipment, so $\mathcal{A}$ does not have economic incentives to mount such an attack [10]. Moreover, the GWN can be physically secured by putting it under a locking system inside the smart home of a user so that the physical capture of the GWN can be much difficult as compared to that for the smart devices.
- The RA is also fully trusted and can not be compromised by an adversary.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2017.2764083, IEEE Transactions on Dependable and Secure Computing

1.4 Contributions

Based upon the above discussion, the following contributions are presented in this paper:

- We propose a new remote user authentication scheme for securing a smart home network. The proposed scheme allows three types of mutual authentications: 1) between a user U_i and the GWN, 2) between the GWN and a smart device SD_j, and 3) a user U_i and a smart device SD_j. At the end, a symmetric session key is established between U_i and SD_j, and they can use the established symmetric key for their future secure communications using a symmetric cipher (for example, the stateless CBC (Cipher Block Chaining) mode of the Advanced Encryption Standard (AES-128), known as AES-CBC [11], [12], [13]).
- The proposed scheme is suitable and efficient for resource-constrained SD_js with limited resources as it uses only hash invocations, simple bitwise XOR operations and symmetric encryption/decryption operations.
- The security of the proposed scheme is proved using the formal security analysis under the widely-accepted ROR model [14], and also using the rigorous informal security analysis. The formal security discussed in Section 5.1 proves the semantic security of the proposed scheme against an adversary to get the session key between a user and a smart device in the smart home environment. On the other hand, using the informal security analysis, we have shown that the proposed scheme is secure against other possible known attacks, which are discussed in detail in Section 5.3.
- The formal security verification of the proposed scheme in Section 5.2 is done using the broadly-used AVISPA tool [15] and the simulation results show that it is also secure against replay and man-in-the-middle attacks.
- Finally, the practical demonstration of the proposed scheme is provided through the widely-accepted NS-2 simulation [16].

1.5 Roadmap of the Paper

The rest of the paper is structured as follows. We briefly discuss the relevant mathematical preliminaries in Section 2. A brief survey of various existing schemes proposed in the literature is given in Section 3. A new user authentication and session key agreement scheme for smart home environment is presented in Section 4. The rigorous formal and informal security analysis are given in Section 5. In addition, the formal security verification using the popular AVISPA tool is also given in this section. The practical demonstration of the proposed scheme using widely-accepted NS-2 simulation is given in Section 6. The performance comparison with the existing relevant schemes is given in Section 7. Finally, Section 8 concludes the article.

2 Mathematical Preliminaries

In this section, we briefly discuss the one-way cryptographic hash function and its properties, and also the indistinguishability of encryption under chosen plaintext attack (IND-CPA), which are necessary to analyze the security of the proposed scheme.

2.1 One-way Cryptographic Hash Function

A one-way cryptographic hash function h: $\{0,1\}^* \rightarrow \{0,1\}^l$ takes an arbitrary-length input, say $x \in \{0,1\}^*$, and outputs a fixed-length (say, l-bits) message digest $h(x) \in \{0,1\}^l$.

Definition 1. *As defined in [17], the formalization of an adversary $\mathcal{A}$'s advantage in finding hash collision is given by $Adv_{\mathcal{A}}^{HASH}(t)$ $= Pr[(a,b) \leftarrow_R \mathcal{A}$: $a \neq b$ and $h(a) = h(b)]$, where $Pr[X]$ denotes the probability of an event X, and $(a,b) \leftarrow_R \mathcal{A}$ denotes the pair (a,b) is randomly selected by $\mathcal{A}$. In this case, $\mathcal{A}$ is allowed to be probabilistic and the probability in the advantage is computed over the random choices made by $\mathcal{A}$ with the execution time t. By an (ϵ, t)-adversary $\mathcal{A}$ attacking the collision resistance of $h(\cdot)$, it is meant that the runtime of $\mathcal{A}$ is at most t and that $Adv_{\mathcal{A}}^{HASH}(t) \leq \epsilon$.*

2.2 Indistinguishability of Encryption Under Chosen Plaintext Attack

The indistinguishability of encryption under chosen plaintext attack (IND-CPA) is formally defined as follows [18], [19]:

Definition 2. *Let SE/ME be the single/multiple eavesdropper respectively, and OR_{ek_1}, OR_{ek_2}, ..., OR_{ek_N} be N different independent encryption oracles associated with encryption keys $ek_1, ek_2, \ldots, ek_N$, respectively. The advantage functions of SE and ME are defined, respectively, as $Adv_{\Omega,SE}^{IND-CPA}(k) = |2Pr[SE \leftarrow OR_{ek_1}; (p_0, p_1 \leftarrow_R SE); \delta \leftarrow_R \{0,1\}; \beta \leftarrow_R OR_{ek_1}(p_\delta) : SE(\beta) = \delta]-1|$, and $Adv_{\Omega,ME}^{IND-CPA}(k) = |2Pr[ME \leftarrow OR_{ek_1}, \ldots, OR_{ek_N}; (p_0, p_1 \leftarrow_R ME); \delta \leftarrow_R \{0,1\}; \beta_1 \leftarrow_R OR_{ek_1}(p_\delta), \ldots, \beta_N \leftarrow_R OR_{ek_N}(p_\delta)$: $ME(\beta_1, \ldots, \beta_N) = \delta] - 1|$, where Ω is the encryption scheme. We call Ω is IND-CPA secure in the single (multiple) eavesdropper setting if $Adv_{\Omega,SE}^{IND-CPA}(k)$ (respectively, $Adv_{\Omega,ME}^{IND-CPA}(k)$) is negligible (in the security parameter k) for any probabilistic, polynomial time adversary SE (ME).*

A deterministic encryption scheme means the same message, when it is encrypted twice, yields the same ciphertext. Thus, any deterministic encryption scheme is not IND-CPA secure [13]. There are five modes of symmetric encryption: Electronic Codebook (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB) and Counter (CTR). Out of these modes, ECB is not IND-CPA secure [13]. Since the adversary knows the Initialization Vector (IV), CBC is essentially reduced to ECB, and hence, the stateful CBC is IND-CPA insecure [13]. On the other hand, in the stateless CBC, the IV value is chosen at random for each message, and due to this property, the stateless CBC is IND-CPA secure [13]. If the stateless CBC of AES-128 symmetric encryption scheme is used for encryption/decryption purpose, it then becomes IND-CPA secure.

3 Related Work

Jeong *et al.* [20] presented a one-time password based user authentication scheme using smart card for smart home networks. Their scheme is lightweight as it uses one-way hash function operations. Their scheme does not provide mutual authentication between GWN and smart device as well as between user and smart device. Their scheme does not provide traceability, and user anonymity properties as the user identity is sent in plaintext and also the messages can be easily traced by an adversary. Furthermore, their scheme is insecure against stolen smart card attack and privileged-insider attack as the adversary can derive secret credentials of a user from the extracted information stored in the smart card. In addition, their scheme is not resilient against smart device physical capture attack.

Vaidya *et al.* [21] proposed a password based remote user authentication scheme for digital home network. Their scheme is also based upon lightweight computation modules such as hashed one-time password and hash-chaining methods. Similar to Jeong *et al.* [20], their scheme does not provide mutual authentication between GWN and smart device as well as between user and smart device. Kim and Kim [22] analyzed Vaidya *et al.*'s scheme [21] and identified that it is vulnerable to password guessing attack and does not provide forward secrecy with lost smart card. They also proposed a new scheme which withstands the security weaknesses observed in Vaidya *et al.*'s scheme [21]. Vaidya *et al.*'s scheme [21] scheme is insecure against stolen smart card attack and privileged-insider attack as the adversary can derive secret credentials of a user from the extracted information stored in the smart card. In addition, their scheme is not resilient against smart device physical capture attack. Later, Vaidya *et al.* [23] also proposed an elliptic curve cryptography (ECC) based device authentication technique for smart energy home area network which requires more overheads as compared to the scheme in [21]. Kim-Kim's scheme [22] is however not resilient against privileged-insider attack, user impersonation attack and password guessing attack. In addition, Kim-Kim's scheme [22] also fails to preserve traceability and anonymity of user and smart device.

Hanumanthappa *et al.* [24] proposed a secure three-way authentication mechanism for user authentication and privacy preservation. In their mechanism, the users or service providers can check whether the device is compromised or not by the help of their proposed encrypted pass-phrases mechanism.

Santoso and Vun [25] proposed ECC based user authentication scheme for a smart home system. In their scheme, the mobile user can authenticate with the devices deployed in the smart home using a central node, called the home gateway. Similar to the schemes of Jeong *et al.* [20], Vaidya *et al.* [21], and Kim and Kim [22], their scheme does not provide traceability, and user anonymity properties. Furthermore, their scheme is insecure against stolen smart card attack and privileged-insider attack. In addition, their scheme is not resilient against smart device physical capture attack.

Chang and Le [26] recently proposed a two-factor user authentication scheme in wireless sensor networks (WSNs), which uses a user's password and smart card. Their scheme has two protocols: $\mathcal{P}_1$ and $\mathcal{P}_2$. While $\mathcal{P}_1$ is based on bitwise XOR and hash functions, $\mathcal{P}_2$ uses ECC along with bitwise XOR and hash functions. However, Das *et al.* [27] proved that both $\mathcal{P}_1$ and $\mathcal{P}_2$ are insecure against session specific temporary information attack and offline password guessing attack, while $\mathcal{P}_1$ is also insecure against session key breach attack. Moreover, they pointed out that both $\mathcal{P}_1$ and $\mathcal{P}_2$ are inefficient in authentication and password change phases. To erase the security limitations in $\mathcal{P}_1$ and $\mathcal{P}_2$, a new authentication and key agreement scheme using ECC in WSNs is presented [27].

Kumar *et al.* [5] presented a lightweight and secure session key establishment scheme for smart home network. To establish the mutual trust, each smart device control unit establishes a session key with the GWN by using a short authentication token. However, their scheme does not preserve the GWN anonymity and also the traceability properties. In addition, their scheme does not provide mutual authentication between user and smart device as well as between user and the GWN.

Li *et al.* [28] proposed an ECC based key establishment scheme for smart home energy management systems. Through the implementation, it is shown that their scheme is efficient with respect to execution time and memory usage. Han *et al.* [29] presented a secure key agreement scheme for ubiquitous smart home systems, which is particularly applicable to the consumer electronics devices in a smart home. The security and functionality features of the existing schemes summarized in Table 4 are also discussed in detail in Section 7.

TABLE 1
Notations used

Notation	Description
RA	Registration authority
GWN	Gateway node
SD_j	j^{th} smart device in the home
U_i	i^{th} user
SP_i	U_i's smart phone
ID_i	U_i's identity
ID_{SD_j}	SD_j's identity
PW_i, BIO_i	U_i's password & personal biometrics, respectively
T_i	Current timestamp
ΔT	Maximum transmission delay
K_{GWN-U_i}	Secret key of GWN for U_i
K_{GWN-SD_j}	Secret key of GWN for SD_j
$E_K(\cdot)/D_K(\cdot)$	Symmetric encryption/decryption (for example, AES-CBC (128 bits) [12]) using key K
σ_i	Biometric secret key of U_i
τ_i	Public reproduction parameter of U_i
t	Error tolerance threshold used in fuzzy extractor
Gen	Fuzzy extractor probabilistic generation procedure
Rep	Fuzzy extractor deterministic reproduction procedure
$h(\cdot)$	One-way collision-resistant cryptographic hash function
$\|\|$, $\oplus$	Concatenation and bitwise XOR operations, respectively

4 The Proposed Scheme

We propose a new user authenticated key establishment scheme for the smart home environment. In the proposed scheme, we have a registration authority, several smart sensing devices, a gateway node (GWN) and several users, who want to access the smart devices. First of all, the secure offline registration of each smart device and GWN is done at the registration authority (RA). Then a user, who wants to access the smart devices, needs to register at the registration authority providing his/her necessary information. Each user has a smart phone, which is capable to read the credential information such as the user's identity, password and biometric (fingerprint scanning etc.) provided by that user. The GWN acts as an intermediary node. The legal user's authentication request goes to the GWN and then the GWN forwards the request to the requested smart device. The smart device sends response to the GWN accordingly and then the GWN forwards the response to the user. As discussed in the threat model provided in Section 1.3, the GWN is fully trusted and all the sensitive informations stored in the GWN are protected from an adversary [5]. Moreover, we assume that all the heterogeneous devices (i.e., GWN, users (smart phones) and smart devices) are synchronized with their clocks, and agree (mutually) on a maximum transmission delay (ΔT) to protect replay attacks in the proposed scheme [5].

Our scheme has six phases: 1) offline smart device and gateway registration, 2) user registration, 3) login, 4) authentication and agreement, 5) biometric and password update, and 6) dynamic smart device addition. The notations presented in Table 1 are used in the proposed scheme. We assume that there are m users and n smart devices in the smart home environment. In addition, we assume that n' additional smart devices can be added in the network through the dynamic smart device addition phase, where $n' << n$. We also use the fuzzy extractor to verify the biometrics. The fuzzy extractor is a tuple $\langle \mathcal{M}, l, t \rangle$, which is composed of the following two algorithms [30], [31]:

Gen: It is a probabilistic algorithm, which takes a biometric template B_i from a given metric space $\mathcal{M}$ as input, and then outputs a biometric key $\sigma_i \in \{0,1\}^l$ and a public reproduction parameter τ_i, that is, $Gen(B_i) = \{\sigma_i, \tau_i\}$, where l denotes the number of bits present in σ_i.

Rep: This is a deterministic algorithm, which takes a noisy biometric template $B_i' \in \mathcal{M}$ and a public parameter τ_i and t related to B_i, and then it reproduces (recovers) the biometric key σ_i. In other words, $Rep(B_i', \tau_i) = \sigma_i$ provided that the Hamming distance between B_i and B_i' is less than or equal to a predefined error tolerance threshold value t.

4.1 Offline Smart Device and Gateway Registration Phase

The offline smart device (SD_j) and GWN registration is done by the registration authority (RA) in offline securely (for example, in person). For each SD_j $(j = 1, 2, \dots, n)$, the RA selects a unique identity ID_{SD_j} and also generates a unique random 1024-bit secret key K_{GWN-SD_j} of GWN for SD_j, and computes the corresponding temporal credential $h(ID_{SD_j}||K_{GWN-SD_j})$, and stores $\{ID_{SD_j}, h(ID_{SD_j}||K_{GWN-SD_j})\}$ into the memory of SD_j. The RA further randomly generates the unique GWN's identity ID_{GWN} and a unique random 1024-bit secret key K_{GWN-U_i} of GWN for each user U_i $(i = 1, 2, \dots, m)$, and also selects the temporary identity TID_i corresponding to each user U_i's identity ID_i into the memory of the GWN after U_i's successful registration phase described in Section 4.2. Finally, the GWN and SD_j contain the information $\langle \{(TID_i, ID_i, K_{GWN-U_i}) | i = 1, 2, \dots, m\}, \{(ID_{SD_j}, K_{GWN-SD_j}) | j = 1, 2, \dots, n\} \rangle$, and $\langle ID_{SD_j}, h(ID_{SD_j}||K_{GWN-SD_j}) \rangle$ for each user U_i and smart device SD_j, respectively.

4.2 User Registration Phase

To access the services from a particular smart device SD_j, a user U_i first needs to register with the RA securely (for example, in person). The following steps are required for the U_i's registration, which are also summarized in Fig. 2:

Step REG1. U_i chooses a unique identity ID_i and a password PW_i, and generates 160-bit random secrets a and r. U_i also imprints his/her biometrics BIO_i to the sensor of SP_i. The SP_i applies the fuzzy extractor probabilistic generation function $Gen(\cdot)$ to generate secret biometric key σ_i and public parameter τ_i as $Gen(BIO_i) = (\sigma_i, \tau_i)$ [31], [32], [33]. The SP_i of U_i calculates the masked password $RPW_i = h(PW_i||\sigma_i||a) \oplus r$, and sends the registration request $\langle ID_i, RPW_i \rangle$ to the RA using a secure channel. Note that a privileged-insider user of the RA being an adversary knows the registration information $\{ID_i, RPW_i\}$ to mount the privileged-insider attack.

Step REG2. After receiving $\langle ID_i, RPW_i \rangle$ from SP_i, the RA first generates a 1024-bit secret key K_{GWN-U_i} of GWN for U_i, and calculates $A_i = h(ID_i||K_{GWN-U_i}) \oplus RPW_i$. RA also generates a temporary identity TID_i corresponding to ID_i for U_i as discussed in the GWN registration phase (Section 4.1). Finally, RA sends the registration reply with information $\{A_i, TID_i\}$ to U_i securely. Note that the privileged-insider user of the RA being an adversary does not know the information $\{A_i, TID_i\}$ as these information are computed online by the RA.

Step REG3. After receiving $\langle A_i, TID_i \rangle$ from the RA, SP_i of U_i computes parameters $B_i = h(ID_i||\sigma_i) \oplus a$, $RPW_i' = RPW_i \oplus r = h(PW_i ||\sigma_i ||a)$, $C_i = h(ID_i ||RPW_i' ||\sigma_i)$ and $A_i^* = A_i \oplus r = h(ID_i ||K_{GWN-U_i}) \oplus RPW_i' = h(ID_i ||K_{GWN-U_i}) \oplus h(PW_i ||\sigma_i ||a)$. Finally, SP_i stores the information $\langle TID_i, A_i^*, B_i, C_i, \tau_i, h(\cdot), Gen(\cdot), Rep(\cdot), t \rangle$ in its memory, where t is the error tolerance parameter used by the fuzzy extractor $Rep(\cdot)$ function.

At the end of this phase, the user U_i erases A_i from his/her smart phone SP_i in order to avoid the privileged-insider attack as explained in Section 5.3.3. In addition, the RA also deletes A_i and RPW_i from its database.

User (U_i)**/ Smart phone** (SP_i)	**Registration authority** (RA)
Choose ID_i, PW_i, and imprint BIO_i.	
Generate 160-bit random secrets a, r.	Select 1024-bit K_{GWN-U_i}.
Compute $Gen(BIO_i) = (\sigma_i, \tau_i)$,	Compute
$RPW_i = h(PW_i\|\|\sigma_i\|\|a) \oplus r$.	$A_i = h(ID_i\|\|K_{GWN-U_i}) \oplus RPW_i$.
$\xrightarrow{\langle ID_i, RPW_i \rangle}$	Generate temporary identity TID_i
(via a secure channel)	corresponding to ID_i.
	$\xleftarrow{\langle A_i, TID_i \rangle}$
Compute $B_i = h(ID_i\|\|\sigma_i) \oplus a$,	(via a secure channel)
$RPW_i' = RPW_i \oplus r = h(PW_i\|\|\sigma_i\|\|a)$,	
$C_i = h(ID_i\|\|RPW_i'\|\|\sigma_i)$, $A_i^* = A_i \oplus r$	
$= h(ID_i\|\|K_{GWN-U_i}) \oplus RPW_i'$.	
Delete A_i from SP_j's memory.	
Store $\{TID_i, A_i^*, B_i, C_i, \tau_i, h(\cdot),$	Store $\{ID_i, TID_i\}$ in GWN's database.
$Gen(\cdot), Rep(\cdot), t\}$ in SP_j's memory.	Delete A_i and RPW_i from its database.

Fig. 2. User registration phase

4.3 Login Phase

The login process of U_i is performed as per the following steps:

Step UL1. U_i first provides his/her identity ID_i and password PW_i^* into the interface of the smart phone SP_i, and also provides his/her biometrics BIO_i^* to the sensor of SP_i. SP_i extracts the biometric key σ_i^* as $\sigma_i^* = Rep(BIO_i^*, \tau_i)$ with the constraint that the Hamming distance between the original biometrics BIO_i at the time of registration and entered current BIO_i^* is less than or equal to t. SP_i further computes $a^* = B_i \oplus h(ID_i||\sigma_i^*)$, $RPW_i^* = h(PW_i^*|| \sigma_i^* ||a^*)$ and $C_i^* = h(ID_i||RPW_i^*||\sigma_i^*)$. SP_i then checks whether $C_i^* = C_i$. If it is valid, U_i passes both password and biometric verification. Otherwise, the session is terminated immediately.

Step UL2. SP_i calculates $M_1 = A_i^* \oplus RPW_i^* = h(ID_i ||K_{GWN-U_i})$. Then SP_i generates a random nonce r_{U_i} and the current timestamp T_1, and calculates parameters $M_2 = M_1 \oplus r_{U_i}$ and $M_3 = h(M_2||T_1|| ID_i||TID_i|| r_{U_i})$. Finally, SP_i sends the login request message $\langle TID_i, M_2, M_3, T_1 \rangle$ to GWN via an open channel.

4.4 Authentication and Key agreement Phase

On receiving the login request $\langle TID_i, M_2, M_3, T_1 \rangle$ from SP_i, following steps are performed by U_i/SP_i, GWN and an accessed

User (U_i)/**Smart phone** (SP_i) $\langle TID_i, A_i^*, B_i, C_i, \tau_i,$ $h(\cdot), Gen(\cdot), Rep(\cdot), t\rangle$	**Gateway node** (GWN) $\langle\{(TID_i, ID_i, K_{GWN-U_i})\|i = 1, 2, \ldots, m\},$ $\{(ID_{SD_j}, K_{GWN-SD_j})\|j = 1, 2, \ldots, n\}\rangle$	**Smart device** (SD_j) $\langle ID_{SD_j}, h(ID_{SD_j}\|\|K_{GWN-SD_j})\rangle$
Input ID_i, PW_i^* & BIO_i^*. Compute $\sigma_i^* = Rep(BIO_i^*, \tau_i)$, $a^* = B_i \oplus h(ID_i\|\|\sigma_i^*)$, $RPW_i^* = h(PW_i^*\|\|\sigma_i^*\|\|a^*)$, $C_i^* = h(ID_i\|\|RPW_i^*\|\|\sigma_i^*)$. Check if $C_i^* = C_i$? If so, compute $M_1 = A_i^* \oplus RPW_i^* = h(ID_i\|\|K_{GWN-U_i})$. Generate r_{U_i} & T_1, and calculate $M_2 = M_1 \oplus r_{U_i}$, $M_3 = h(M_2\|\|T_1\|\|ID_i\|\|TID_i\|\|r_{U_i})$. $\xrightarrow{\langle TID_i, M_2, M_3, T_1\rangle}$ (via open channel)	Check if $\|T_1 - T_1^*\| \leq \Delta T$? If so, extract ID_i and K_{GWN-U_i} corresponding to TID_i. Compute $M_4 = h(ID_i\|\|K_{GWN-U_i})(= M_1)$ using extracted ID_i & K_{GWN-U_i}, $r_{U_i}^* = M_2 \oplus M_4$, $M_5 = h(M_2\|\|T_1\|\|ID_i\|\|TID_i\|\|r_{U_i}^*)$. Check if $M_5 = M_3$? If matches, generate r_{GWN} & T_2. Compute $M_6 = h(ID_{SD_j}\|\|K_{GWN-SD_j})$, $M_7 = E_{M_6}[ID_i, ID_{GWN}, r_{U_i}^*, r_{GWN}, h(M_4)]$, $M_8 = h(M_6\|\|T_2\|\|ID_i\|\|ID_{SD_j}\|\|ID_{GWN}\|\|r_{GWN})$. $\xrightarrow{\langle M_7, M_8, T_2\rangle}$ (via open channel)	Check if $\|T_2 - T_2^*\| \leq \Delta T$? If so, decrypt M_7 to retrieve $(ID_i, ID_{GWN}, r_{U_i}^*, r_{GWN}, h(M_4))$ $= D_{h(ID_{SD_j}\|\|K_{GWN-SD_j})}[M_7]$. Compute $M_9 = h[h(ID_{SD_j}\|\|K_{GWN-SD_j})$ $\|\|T_2\|\|ID_i\|\|ID_{SD_j}\|\|ID_{GWN}\|\|r_{GWN}]$. Check if $M_9 = M_8$? If so, generate r_{SD_j} & T_3, and compute $SK_{ij} = h[ID_i\|\|ID_{SD_j}\|\|$ $ID_{GWN}\|\|r_{U_i}^*\|\|r_{GWN}\|\|r_{SD_j}\|\|h(M_4)$ $\|\|h(h(ID_{SD_j}\|\|K_{GWN-SD_j}))]$, $M_{10} = h(h(ID_{SD_j}\|\|K_{GWN-SD_j})\|\|T_3) \oplus r_{SD_j}$, $M_{11} = h(SK_{ij}\|\|T_3)$, $M_{12} = h(r_{SD_j}\|\|r_{GWN}\|\|ID_{SD_j}\|\|ID_{GWN}\|\|T_3)$. $\xleftarrow{\langle M_{10}, M_{11}, M_{12}, T_3\rangle}$ (via open channel)
Check if $\|T_4 - T_4^*\| \leq \Delta T$? If so, decrypt $D_{M_1}[M_{14}] = (r_{U_i}^*, r_{GWN}^*, r_{SD_j}^*,$ $ID_{SD_j}, ID_{GWN}, h(M_6))$. Check if $r_{U_i}^* = r_{U_i}$? If so, compute $SK_{ij}' = h[ID_i\|\|ID_{SD_j}$ $\|\|ID_{GWN}\|\|r_{U_i}\|\|r_{GWN}^*\|\|r_{SD_j}^*$ $\|\|h(M_1)\|\|h(M_6)]$, $M_{17} = h(h(SK_{ij}'\|\|T_3)\|\|T_4\|\|r_{U_i})$. Check if $M_{17} = M_{16}$? If so, U_i and SD_j establish session key $SK_{ij}'(= SK_{ij})$. Compute $TID_i^{new} = M_{15} \oplus h(TID_i$ $\|\|M_1\|\|T_3\|\|T_4)$. Replace TID_i with TID_i^{new}.	Check if $\|T_3 - T_3^*\| \leq \Delta T$? If so, compute $r_{SD_j}^* = M_{10}\oplus$ $h[h(ID_{SD_j}\|\|K_{GWN-SD_j})\|\|T_3]$, $M_{13} = h(r_{SD_j}^*\|\|r_{GWN}\|\|ID_{SD_j}\|\|ID_{GWN}\|\|T_3)$. Check if $M_{13} = M_{12}$? If so, compute $M_{14} = E_{M_4}[r_{U_i}^*, r_{GWN}, r_{SD_j}^*,$ $ID_{SD_j}, ID_{GWN}, h(M_6)]$. Generate T_4, select TID_i^{new} and compute $M_{15} = TID_i^{new} \oplus h(TID_i\|\|M_4\|\|T_3\|\|T_4)$, $M_{16} = h(M_{11}\|\|T_4\|\|r_{U_i}^*)$. $\xleftarrow{\langle M_{14}, M_{15}, M_{16}, T_3, T_4\rangle}$ (via open channel)	

Fig. 3. Summary of login, and authentication and key agreement phases

smart device SD_j to establish a session key between U_i and SD_j for later secure communication:

Step AUKA1. GWN first checks the timeliness of T_1 by condition $|T_1 - T_1^*| \leq \Delta T$, where the maximum transmission delay is denoted by ΔT and T_1^* is the reception time of the message $\langle TID_i, M_2, M_3, T_1\rangle$. If the condition matches, the GWN searches the received TID_i in its database and if it is found in the database, the GWN extracts ID_i and K_{GWN-U_i} corresponding to TID_i from its database, and calculates $M_4 = h(ID_i||K_{GWN-U_i})$ $(= M_1)$ using the extracted ID_i and K_{GWN-U_i}, $r_{U_i}^* = M_2 \oplus M_4 = M_2 \oplus M_1$, $M_5 = h(M_2||T_1|| ID_i|| TID_i|| r_{U_i}^*)$.

Step AUKA2. GWN checks if $M_5 = M_3$ holds. If it does not match, it terminates the authentication process. Otherwise GWN generates a random nonce r_{GWN} and timestamp T_2, and calculates parameters $M_6 = h(ID_{SD_j} ||K_{GWN-SD_j})$, $M_7 = E_{M_6}[ID_i,\, ID_{GWN},\, r_{U_i}^*,\, r_{GWN},\, h(M_4)]$, $M_8 = h(M_6|| T_2|| ID_i|| ID_{SD_j} ||ID_{GWN} ||r_{GWN})$. For computing M_7, if we use the stateless CBC of AES-128 (AES-CBC) symmetric encryption scheme, then the GWN needs to set the IV of CBC as $IV = h(M_6||T_1)$ so that it is random for each message in a particular session. Then GWN sends the authentication request message $\langle M_7, M_8, T_2\rangle$ to SD_j via an open channel.

Step AUKA3. After receiving the message $\langle M_7, M_8, T_2\rangle$ from GWN, SD_j checks the timeliness of T_2 by the criteria $|T_2 - T_2^*| \leq \Delta T$, where T_2^* is the reception time of the message $\langle M_7, M_8, T_2\rangle$. If condition holds, SD_j decrypts M_7 using the stored key $h(ID_{SD_j}||K_{GWN-SD_j})$ as $(ID_i,\, ID_{GWN},\, r_{U_i}^*,\, r_{GWN},\, h(M_4)) = D_{h(ID_{SD_j}||K_{GWN-SD_j})}$ $[M_7]$. For decrypting M_7, SD_j also needs to set the IV of CBC as $IV = h(h(ID_{SD_j}||K_{GWN-SD_j})||T_1)$ $(= h(M_6||T_1))$.

Step AUKA4. SD_j calculates $M_9 = h[h(ID_{SD_j}|| K_{GWN-SD_j})\ ||T_2\ ||ID_i\ ||ID_{SD_j}\ ||ID_{GWN}\ ||r_{GWN}]$ and checks the condition $M_9 = M_8$. If it does not match, it terminates the authentication process. Otherwise, SD_j generates a random nonce r_{SD_j} and the current timestamp T_3, and computes the session key as $SK_{ij} = h[ID_i\ ||ID_{SD_j}\ ||ID_{GWN}\ ||r_{U_i}^*\ ||r_{GWN}||\ r_{SD_j}\ ||h(M_4)||\ h(h(ID_{SD_j}||\ K_{GWN-SD_j}))]$. After that, SD_j computes parameters $M_{10} = h(h(ID_{SD_j}\ ||K_{GWN-SD_j})\ ||T_3)\oplus r_{SD_j}$, $M_{11} = h(SK_{ij}\ ||T_3)$ and $M_{12} = h(r_{SD_j}\ ||r_{GWN}\ ||ID_{SD_j}\ ||ID_{GWN}\ ||T_3)$. Then SD_j sends the authentication reply message $\langle M_{10}, M_{11}, M_{12}, T_3\rangle$ to the GWN via an insecure channel.

Step AUKA5. Upon receiving authentication request message, GWN checks the timeliness of T_3 by applying the criteria $|T_3 - T_3^*| \leq \Delta T$, where T_3^* is the reception time of the message $\langle M_{10},\, M_{11},\, M_{12},\, T_3\rangle$. If condition matches, GWN computes $r_{SD_j}^* = M_{10}\oplus\ h[h(ID_{SD_j}\ ||K_{GWN-SD_j})\ ||T_3]$ and $M_{13} = h(r_{SD_j}^*\ ||r_{GWN}\ ||ID_{SD_j}\ ||ID_{GWN}\ ||T_3)$. The GWN checks the condition $M_{13} = M_{12}$. If it does not match, the GWN aborts the message. Otherwise, GWN computes M_{14} using previously computed $M_4 = h(ID_i||K_{GWN-U_i})$ as $M_{14} = E_{M_4}$ $[r_{U_i}^*,\, r_{GWN},\, r_{SD_j}^*,\, ID_{SD_j},\, ID_{GWN},\, h(M_6)]$. For encrypting the information in M_{14} using the key M_4, we also use the stateless CBC of AES-128 (AES-CBC) symmetric encryption scheme and thus, the GWN needs to set the IV of CBC as $IV = h(M_4||T_4)$ so that it is random for each message in a particular session. The GWN chooses current timestamp T_4 and generates a new temporary identity TID_i^{new} corresponding to ID_i. The GWN further computes $M_{15} = TID_i^{new}\ \oplus h(TID_i\ ||M_4\ ||T_3\ ||T_4)$

and $M_{16} = h(M_{11} \,||T_4 \,||r^*_{U_i})$. The GWN sends the message $\langle M_{14}, M_{15}, M_{16}, T_3, T_4\rangle$ to U_i via insecure channel.

Step AUKA6. After receiving the message $\langle M_{14}, M_{15}, M_{16}, T_3, T_4\rangle$, SP_i of U_i first checks the timeliness of T_4 with the condition $|T_4 - T^*_4| \leq \Delta T$, where T^*_4 is the reception time of the message. If condition matches, U_i decrypts M_{14} using pre-computed M_1 as $D_{M_1}\,[M_{14}] = (r^*_{U_i}, r^*_{GWN}, r^*_{SD_j}, ID_{SD_j}, ID_{GWN}, h(M_6))$. For decrypting M_{14}, SD_j also needs to set the IV of CBC as $IV = h(M_1||T_4)\ (= h(M_4||T_4))$.

Then SP_i checks if $r^*_{U_i} = r_{U_i}$. If they do not match, SP_i terminates the authentication process. Otherwise, it computes the session key $SK'_{ij} = h[ID_i \,||ID_{SD_j} \,||ID_{GWN} \,||r_{U_i} \,||r^*_{GWN} \,||r^*_{SD_j} \,||h(M_1) \,||h(M_6)]$ and $M_{17} = h(h(SK'_{ij} \,||T_3) \,||T_4 \,||r_{U_i})$, and then matches if $M_{17} = M_{16}$. If it does not match, SP_i terminates the session and discards the computed session key. Otherwise, message comes from the valid source and the computed session key SK'_{ij} is authentic. Finally, SP_i computes the new temporary identity as $TID^{new}_i = M_{15} \oplus h(TID_i \,||M_1 \,||T_3 \,||T_4)$ and replaces TID_i with TID^{new}_i in its memory.

The login, and authentication and agreement phases are summarized in Fig. 3.

4.5 Password and Biometric Update Phase

The proposed scheme provides password and biometric update facility through which a legitimate user U_i can update his/her password and biometrics for security reasons at any time after user registration phase without further involving the RA. Note that the biometric information of a given user U_i is unique and unchanged as compared to the chosen password by that user U_i. However, we suggest the user U_i to update his/her biometric information in the proposed scheme, if he/she desires to do so. This is required to protect strongly the offline password guessing attack to be considered in this phase as described by Huang *et al.* [34], which is discussed in detail in Section 5.3.11. This phase needs the following steps:

Step PBU1. U_i provides his/her identity ID_i, old password PW^{old}_i to interface of the SP_i and current his/her biometrics BIO^{old}_i to the sensor of the SP_i. SP_i then computes $\sigma^{old}_i = Rep(BIO^{old}_i, \tau_i)$, $a' = B_i \oplus h(ID_i||\sigma^{old}_i)$, $RPW^{old}_i = h(PW^{old}_i \,||\sigma^{old}_i \,||a')$ and $C^{old}_i = h(ID_i||\, RPW^{old}_i \,||\sigma^{old}_i)$. SP_i checks the condition $C^{old}_i = C_i$. If it matches, U_i is the actual user; otherwise, the phase is terminated immediately.

Step PBU2. SP_i asks U_i to enter a new password PW^{new}_i and also imprint new biometrics BIO^{new}_i. The SP_i then calculates $Gen(BIO^{new}_i) = (\sigma^{new}_i, \tau^{new}_i)$, $RPW^{new}_i = h(PW^{new}_i \,||\sigma^{new}_i \,||a')$, $B^{new}_i = h(ID_i||\sigma^{new}_i) \oplus a'$, $C^{new}_i = h(ID_i \,||RPW^{new}_i \,||\sigma^{new}_i)$ and $A^{new}_i = A^*_i \oplus RPW^{old}_i \oplus RPW^{new}_i$, $= h(ID_i||\, K_{GWN-U_i}) \oplus RPW^{new}_i = h(ID_i||\, K_{GWN-U_i}) \oplus h(PW^{new}_i \,||\sigma^{new}_i \,||a')$.

Step PBU3. Finally, SP_i replaces τ_i, A^*_i, B_i, and C_i with τ^{new}_i, A^{new}_i, B^{new}_i, and C^{new}_i in its memory, respectively.

The password and biometric update phase is also summarized in Fig. 4.

User (U_i)	Smart phone (SP_i) $\langle TID_i, A^*_i, B_i, C_i, \tau_i, h(\cdot), Gen(\cdot), Rep(\cdot), t\rangle$
Provide ID_i, PW^{old}_i & BIO^{old}_i.	Compute $\sigma^{old}_i = Rep(BIO^{old}_i, \tau_i)$, $a' = B_i \oplus h(ID_i\|\|\sigma^{old}_i)$, $RPW^{old}_i = h(PW^{old}_i\|\|\sigma^{old}_i\|\|a')$, $C^{old}_i = h(ID_i\|\|RPW^{old}_i\|\|\sigma^{old}_i)$. Check if $C^{old}_i = C_i$? If so, ask U_i to provide new password & biometrics.
Provide PW^{new}_i & BIO^{new}_i.	Compute $Gen(BIO^{new}_i) = (\sigma^{new}_i, \tau^{new}_i)$, $RPW^{new}_i = h(PW^{new}_i\|\|\sigma^{new}_i\|\|a')$, $B^{new}_i = h(ID_i\|\|\sigma^{new}_i) \oplus a'$, $C^{new}_i = h(ID_i\|\|RPW^{new}_i\|\|\sigma^{new}_i)$, $A^{new}_i = A^*_i \oplus RPW^{old}_i \oplus RPW^{new}_i$, $= h(ID_i\|\|K_{GWN-U_i}) \oplus RPW^{new}_i$. Finally, SP_i replaces τ_i, A^*_i, B_i and C_i with τ^{new}_i, A^{new}_i, B^{new}_i and C^{new}_i, respectively.

Fig. 4. Summary of password and biometric update phase

4.6 Dynamic Smart Device Addition Phase

To deploy a new smart device SD^{new}_j in the existing smart home network, the RA performs the following steps in offline:

Step DA1. RA first assigns a unique new identity $ID^{new}_{SD_j}$ and also generates a new secret key $K_{GWN-SD^{new}_j}$ of GWN for SD^{new}_j. RA further computes the temporal credential of SD^{new}_j as $h(ID_{SD^{new}_j}||K_{GWN-SD^{new}_j})$.

Step DA2. RA stores the information $\{ID_{SD^{new}_j}, h(ID_{SD^{new}_j} \,||K_{GWN-SD^{new}_j})\}$ into the memory of SD_j before its deployment in the smart home. RA also sends the information $\{ID_{SD^{new}_j}, K_{GWN-SD^{new}_j}\}$ to the GWN securely, which are then stored in the database of the GWN.

Finally, RA also needs to inform the existing users in the network about the deployment of new smart device SD^{new}_j so that they can access the services from SD^{new}_j, if needed.

5 Security Analysis

In this section, we analyze the security of the proposed scheme using both formal and informal analysis.

5.1 Formal Security Analysis using Real-Or-Random Model

The widely-accepted Real-Or-Random (ROR) model [14] is used for formal security analysis of the proposed scheme.

5.1.1 ROR Model

We follow the Abdalla *et al.*'s ROR model [14] for formal security analysis as done in [26]. According to our scheme, we have three participants in the smart home: smart device SD_j, user U_i and GWN.

Participants. Let $\Pi^t_{SD_j}$, $\Pi^u_{U_i}$ and Π^v_{GWN} be the instances t, u and v of SD_j, U_i and GWN, respectively. These are called oracles [26].

Accepted state. An instance Π^t is known to be accepted, if upon receiving the last expected protocol message, it goes into an accept state. The ordered concatenation of all communicated (sent and received) messages by Π^t forms the session identification (sid) of Π^t for the current session.

Partnering. Two instances Π^{t_1} and Π^{t_2} are said to be partnered if the following three conditions are fulfilled simultaneously: 1) both Π^{t_1} and Π^{t_2} are in accept state; 2) both Π^{t_1} and Π^{t_2} mutually authenticate each other and share the same sid; and 3) Π^{t_1} and Π^{t_2} are mutual partners of each other.

Freshness. The instance $\Pi^u_{U_i}$ or $\Pi^t_{SD_j}$ is fresh, if the session key SK_{ij} between U_i and SD_j has not revealed to an adversary $\mathcal{A}$ using the Reveal(Π^t) query given below [26].

Adversary. It is assumed that $\mathcal{A}$ has fully control over all the communications in a smart home. $\mathcal{A}$ has the ability to read, modify the exchanged messages, or can fabricate new messages and inject them into the network. Furthermore, $\mathcal{A}$ has access to the following queries [26]:

$Execute(\Pi^u, \Pi^v, \Pi^t)$: $\mathcal{A}$ can execute this query to obtain the messages exchanged between three legitimate participants U_i, GWN and SD_j, which is further modeled as an eavesdropping attack.

$Reveal(\Pi^t)$: This query reveals the current session key SK_{ij} generated by Π^t (and its partner) to an adversary $\mathcal{A}$.

$Send(\Pi^t, msg)$: $\mathcal{A}$ runs this query to send a message, say msg, to a participant instance Π^t, and also receives a response message. It is modeled as an active attack.

$CorruptSmartPhone(\Pi^u_{U_i})$: It represents the smart phone SP_i lost/stolen attack, which outputs the information stored in SP_i.

$CorruptSmartDevice(\Pi^t_{SD_j})$: This represents an attack in which secret $h(ID_{SD_j}||K_{GWN-SD_j})$ is disclosed to $\mathcal{A}$, which is applied to verify the security of the proposed scheme. As mentioned in [26], both $CorruptSmartPhone$ and $CorruptSmartDevice$ queries ensure the weak-corruption model in which temporary keys and the internal data of the participant instances are not corrupted.

$Test(\Pi^t)$: It represents the semantic security of a session key SK_{ij} between U_i and SD_j following the indistinguishability in the ROR model [14]. An unbiased coin b is flipped before start of the experiment, and its result is only known to $\mathcal{A}$ which is used to decide the output of the $Test$ query. If $\mathcal{A}$ runs this query, and the established session key SK_{ij} is also new, then Π^t returns SK_{ij} in case $b = 1$ or a random number for $b = 0$; otherwise, it outputs $\perp$ (null).

Note that we impose a restriction that the adversary $\mathcal{A}$ has access to only limited number of $CorruptSmartPhone(\Pi^u_{U_i})$ and $CorruptSmartDevice(\Pi^t_{SD_j})$ queries, whereas he/she can access the $Test(\Pi^t)$ query many times. According to the threat model described in Section 1.3, the GWN is trusted. Thus, $\mathcal{A}$ does not have any access to a corrupt query related to the GWN.

Semantic security of session key. According to the requirements of the ROR model [14], $\mathcal{A}$ needs to distinguish between an instance's real session key and a random key. $\mathcal{A}$ can make several $Test$ queries to either $\Pi^t_{SD_j}$ or $\Pi^u_{U_i}$. The output of $Test$ query should be consistent with respect to the random bit b. After the experiment is finished, $\mathcal{A}$ returns a guessed bit b' and he/she can win the game if the condition $b' = b$ is met. Let $SUCC$ be an event that $\mathcal{A}$ win the game. The advantage $Adv^{AKE}_{\mathcal{P}}$ of $\mathcal{A}$ in breaking the semantic security of our authenticated key agreement (AKE) scheme, say $\mathcal{P}$ against deriving the session key SK_{ij} between U_i and SD_j is given by $Adv^{AKE}_{\mathcal{P}} = |2.Pr[SUCC] - 1|$. In the ROR sense, $\mathcal{P}$ is secure if $Adv^{AKE}_{\mathcal{P}} \leq \psi$, where $\psi > 0$ is a sufficiently small real number.

Random oracle. As mentioned in [26], all communicating participants as well as $\mathcal{A}$ have access to a collision-resistant one-way cryptographic hash function $h(\cdot)$. $h(\cdot)$ is modeled by a random oracle, say HO.

5.1.2 Security Proof

Theorem 1 provides the semantic security of our proposed scheme under the widely-accepted ROR model [26], [35].

Theorem 1. *Let $\mathcal{A}$ be an adversary running in polynomial time t against our scheme $\mathcal{P}$ in the random oracle, D a uniformly distributed password dictionary and l the number of bits present in the biometrics key σ_i. The advantage of $\mathcal{A}$ in breaking semantic security of our scheme is estimated as $Adv^{AKE}_{\mathcal{P}} \leq \frac{q_h^2}{|Hash|} + \frac{q_{send}}{2^{l-1}.|D|} + 2Adv^{IND-CPA}_{\Omega}(k)$, where q_h, q_{send}, $|Hash|$, $|D|$ and $Adv^{IND-CPA}_{\Omega,SE}(k)$ or $Adv^{IND-CPA}_{\Omega,ME}(k)$ are the number of HO queries, the $Send$ queries, the range space of $h(\cdot)$, the size of D, and the advantage of $\mathcal{A}$ in breaking the IND-CPA secure symmetric cipher Ω (provided in Definition 2), respectively, and $Adv^{IND-CPA}_{\Omega}(k) = Adv^{IND-CPA}_{\Omega,SE}(k)$ or $Adv^{IND-CPA}_{\Omega,ME}(k)$.*

Proof. The proof is similar to that presented in the schemes [26], [35]. The sequence of five games, say GM_i, are defined in the security analysis, where $i = 0, 1, 2, 3, 4$. Assume that $SUCC_i$ be an event wherein an adversary $\mathcal{A}$ can guess the random bit b in GM_i correctly.

GM_0**:** This game corresponds to a real attack performed by $\mathcal{A}$ against our scheme $\mathcal{P}$ in the ROR sense. The bit b is chosen at the beginning of GM_0. Hence, it follows that

$$Adv^{AKE}_{\mathcal{P}} = |2.Pr[SUCC_0] - 1|. \qquad (1)$$

GM_1**:** This game represents an eavesdropping attack performed by the single/multiple eavesdropper SE/ME, say $\mathcal{A}$, where $\mathcal{A}$ can query $Execute(\Pi^u,\ \Pi^v,\ \Pi^t)$ oracle. At the end of the game, $\mathcal{A}$ makes queries to the $Test$ oracle. The output of $Test$ oracle determines whether it is the actual session key SK_{ij} or a random number. Note that the session key SK_{ij} is calculated by both U_i and SD_j as $SK_{ij} = h[ID_i||\ ID_{SD_j}\ ||ID_{GWN}\ ||r^*_{U_i}\ ||r_{GWN}\ ||r_{SD_j}\ ||h(M_4)\ ||h(h(ID_{SD_j}\ ||K_{GWN-SD_j}))]$, where $M_4 = h(ID_i\ ||K_{GWN-U_i})$. To calculate SK_{ij}, $\mathcal{A}$ must have M_4 and $h(ID_{SD_j}\ ||K_{GWN-SD_j})$, which further involve secret keys K_{GWN-U_i} and K_{GWN-SD_j}. $\mathcal{A}$ also requires ID_i, ID_{SD_j}, ID_{GWN}, r_{U_i}, r_{GWN} and r_{SD_j} for calculating SK_{ij}, which are unknown to him/her. As a consequence, the chance of winning the game GM_1 for $\mathcal{A}$ is not increased by eavesdropping attack. It is then obvious that

$$Pr[SUCC_0] = Pr[SUCC_1]. \qquad (2)$$

GM_2**:** By adding the simulations of the $Send$ and HO oracles are added into GM_1, GM_1 is transformed into GM_2, which represents an active attack. In this game, the objective of $\mathcal{A}$ is to fool a participant to accept a modified message. $\mathcal{A}$ is permitted to make different HO queries to examine the existence of the hash collisions. All the exchanged messages $\langle TID_i,\ M_2,\ M_3,\ T_1\rangle$, $\langle M_7,\ M_8,\ T_2\rangle$, $\langle M_{10},\ M_{11},\ M_{12},\ T_3\rangle$ and $\langle M_{14},\ M_{15},\ M_{16},\ T_3,\ T_4\rangle$ during the login and authentication phase contain the participant's identity, random nonce and timestamps. Hence, there is no collision when the $Send$ oracle is queried by $\mathcal{A}$. The results of the birthday paradox give

$$|Pr[SUCC_1] - Pr[SUCC_2]| \leq q_h^2/(2|Hash|). \qquad (3)$$

GM_3**:** GM_2 is transformed into GM_3 by adding the simulation of $CorruptSmartPhone$ oracle. $\mathcal{A}$ can choose low-entropy passwords, and using the information stored into SP_i he/she may try to acquire the user's password using the dictionary attack. Again, $\mathcal{A}$ may try to acquire the biometrics key σ_i from the information stored in SP_i. We have used a strong fuzzy extractor in our scheme $\mathcal{P}$, which is capable to extract at most l random bits and the guessing probability of $\sigma_i \in \{0, 1\}^l$ by $\mathcal{A}$ is approximately $\frac{1}{2^l}$ [31]. It is also assumed that the system allows the limited number of wrong password inputs. Thus, we have the following result,

$$|Pr[SUCC_2] - Pr[SUCC_3]| \leq q_{send}/(2^l.|D|). \qquad (4)$$

GM_4**:** GM_3 is transformed into GM_4, where GM_4 is the last game. It models an attack in which $\mathcal{A}$ can physically capture (compromise) a smart device SD_j by adding the simulation of $CorruptSmartDevice$ oracle. $\mathcal{A}$ then knows the information $\{ID_{SD_j},\ h(ID_{SD_j}||K_{GWN-SD_j})\}$ which is stored in SD_j. Let $\mathcal{A}$ also has all the eavesdropped messages $\langle TID_i,\ M_2,\ M_3,\ T_1\rangle$, $\langle M_7,\ M_8,\ T_2\rangle$, $\langle M_{10},\ M_{11},\ M_{12},\ T_3\rangle$ and $\langle M_{14},\ M_{15},\ M_{16},\ T_3,\ T_4\rangle$. Then, $\mathcal{A}$ tries to retrieve the information $\{ID_i,\ ID_{GWN}, r_{U_i},\ r_{GWN}, h(M_4)\}$ by decrypting M_7 using $h(ID_{SD_j}||K_{GWN-SD_j})\}$ as $(ID_i, ID_{GWN},\ r^*_{U_i}, r_{GWN},\ h(M_4)) = D_{h(ID_{SD_j}||K_{GWN-SD_j})}\ [M_7]$. However, $\mathcal{A}$ can not decrypt M_{14} as M_4 is unknown to him/her since as $M_{14} = E_{M_4}$ $[r^*_{U_i}, r_{GWN}, r^*_{SD_j}, ID_{SD_j}, ID_{GWN}, h(M_6)]$. This implies that without having $M_4 = h(ID_i||K_{GWN-U_i})\ (= M_1)$, it is quite difficult task for $\mathcal{A}$ to extract the information $\{r^*_{U_i}, r_{GWN}, r^*_{SD_j},\ ID_{SD_j},\ ID_{GWN},\ h(M_6)\}$. Thus, computation of the session key $SK_{ij} = h[ID_i\ ||ID_{SD_j}\ ||ID_{GWN}\ ||r_{U_i}\ ||r_{GWN}\ ||r_{SD_j}\ ||h(M_1)\ ||h(M_6)]\ (= SK'_{ij})$ is difficult as $\mathcal{A}$ needs the necessary information including r_{SD_j} and $M_1\ (= M_4)$ due to the IND-CPA secure symmetric cipher used in the proposed scheme for encryption/decryption. This concludes that

$$|Pr[SUCC_3] - Pr[SUCC_4]| \leq Adv_{\Omega}^{IND-CPA}(k). \qquad (5)$$

In GM_4, all the random oracles are simulated. $\mathcal{A}$ is only left to guess the bit b for winning the game after querying the $Test$ oracle. It is clear that $Pr[SUCC_4] = 1/2$.

From Equation (1), we get, $\frac{1}{2}.Adv_{\mathcal{P}}^{AKE} = |Pr[SUCC_0] - \frac{1}{2}|$. Using the triangular inequality, we have, $|Pr[SUCC_1] - Pr[SUCC_4]| \leq |Pr[SUCC_1] - Pr[SUCC_2]| + |Pr[SUCC_2] - Pr[SUCC_4]| \leq |Pr[SUCC_1] - Pr[SUCC_2]| + |Pr[SUCC_2] - Pr[SUCC_3]| + |Pr[SUCC_3] - Pr[SUCC_4]| \leq \frac{q_h^2}{2.|Hash|} + \frac{q_{send}}{2^l.|D|} + Adv_{G_q}^{ECDDHP}(t)$.

Using Equations (2) – (5), we have,

$$\begin{aligned} |Pr[SUCC_0] - 1/2| \quad \leq \quad & q_h^2/(2.|Hash|) + q_{send}/(2^l.|D|) \\ & + Adv_{\Omega}^{IND-CPA}(k). \qquad (6) \end{aligned}$$

Finally, Equation (6) yields the required result:

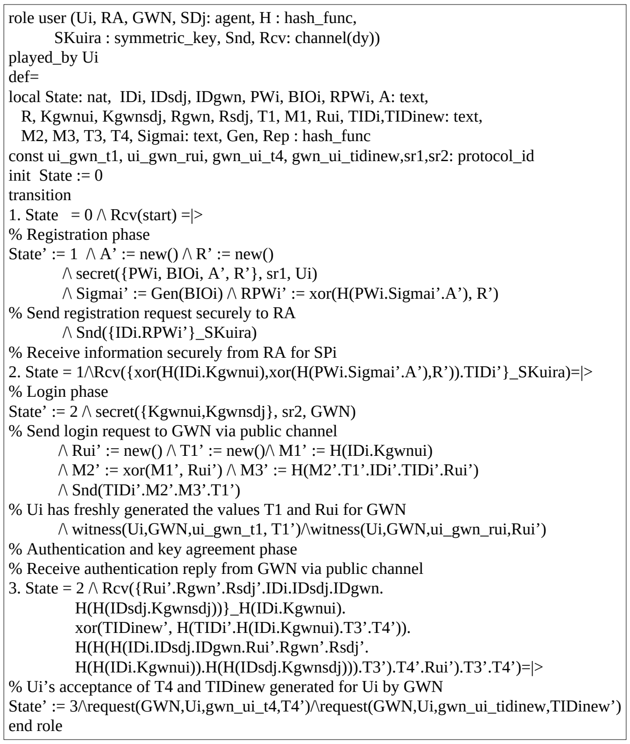
$$Adv_{\mathcal{P}}^{AKE} \leq \frac{q_h^2}{|Hash|} + \frac{q_{send}}{2^{l-1}.|D|} + 2Adv_{\Omega}^{IND-CPA}(k).$$

□

5.2 Formal Security Verification using AVISPA

The proposed scheme is simulated for the formal security verification using the broadly-accepted Automated Validation of Internet Security Protocols and Applications (AVISPA) tool to exhibit that the proposed scheme withstands replay and man-in-the-middle attacks.

AVISPA integrates four back ends that implement different state-of-the-art automatic analysis mechanisms: (i) OFMC; (ii) CL-AtSe; (iii) SATMC; and (iv) TA4SP. The detailed description and functionality of these back ends are available in [15], [35], [36], [37], [38]. A security protocol requires to be implemented in the High Level Protocols Specification Language (HLPSL) [39], which is converted into intermediate format (IF) using the HLPSL2IF translator. The IF is then given as input to one of the four backends to produce output, which has various sections highlighting whether the designed scheme is safe or unsafe against an adversary.



```
role user (Ui, RA, GWN, SDj: agent, H : hash_func,
        SKuira : symmetric_key, Snd, Rcv: channel(dy))
played_by Ui
def=
local State: nat,  IDi, IDsdj, IDgwn, PWi, BIOi, RPWi, A: text,
  R, Kgwnui, Kgwnsdj, Rgwn, Rsdj, T1, M1, Rui, TIDi,TIDinew: text,
  M2, M3, T3, T4, Sigmai: text, Gen, Rep : hash_func
const ui_gwn_t1, ui_gwn_rui, gwn_ui_t4, gwn_ui_tidinew,sr1,sr2: protocol_id
init  State := 0
transition
1. State   = 0 /\ Rcv(start) =|>
% Registration phase
State' := 1  /\ A' := new() /\ R' := new()
         /\ secret({PWi, BIOi, A', R'}, sr1, Ui)
         /\ Sigmai' := Gen(BIOi) /\ RPWi' := xor(H(PWi.Sigmai'.A'), R')
% Send registration request securely to RA
         /\ Snd({IDi.RPWi'}_SKuira)
% Receive information securely from RA for SPi
2. State = 1/\Rcv({xor(H(IDi.Kgwnui),xor(H(PWi.Sigmai'.A'),R')).TIDi'}_SKuira)=|>
% Login phase
State' := 2 /\ secret({Kgwnui,Kgwnsdj}, sr2, GWN)
% Send login request to GWN via public channel
        /\ Rui' := new() /\ T1' := new()/\ M1' := H(IDi.Kgwnui)
        /\ M2' := xor(M1', Rui') /\ M3' := H(M2'.T1'.IDi'.TIDi'.Rui')
        /\ Snd(TIDi'.M2'.M3'.T1')
% Ui has freshly generated the values T1 and Rui for GWN
        /\ witness(Ui,GWN,ui_gwn_t1, T1')/\witness(Ui,GWN,ui_gwn_rui,Rui')
% Authentication and key agreement phase
% Receive authentication reply from GWN via public channel
3. State = 2 /\ Rcv({Rui'.Rgwn'.Rsdj'.IDi.IDsdj.IDgwn.
           H(H(IDsdj.Kgwnsdj))}_H(IDi.Kgwnui).
           xor(TIDinew', H(TIDi'.H(IDi.Kgwnui).T3'.T4')).
           H(H(H(IDi.IDsdj.IDgwn.Rui'.Rgwn'.Rsdj'.
           H(H(IDi.Kgwnui)).H(H(IDsdj.Kgwnsdj))).T3').T4'.Rui').T3'.T4')=|>
% Ui's acceptance of T4 and TIDinew generated for Ui by GWN
State' := 3/\request(GWN,Ui,gwn_ui_t4,T4')/\request(GWN,Ui,gwn_ui_tidinew,TIDinew')
end role
```

Fig. 5. The user U_i's role in HLPSL

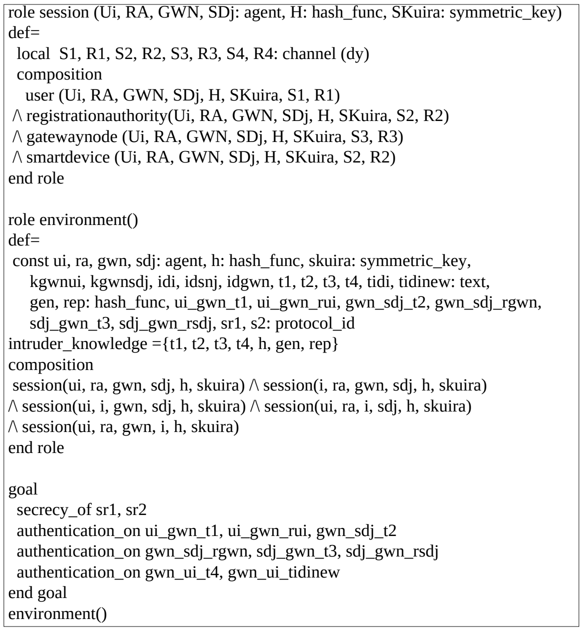
The registration, login, authentication and session key agreement phases of our scheme are implemented in HLPSL. In our implementation, four basic roles are defined: *registration authority*, *user*, *gateway node* and *smart device* for representing the RA, a user U_i, the GWN and a smart device SD_j, respectively. The HLPSL role specification $user$ for U_i is given in Fig. 5. U_i as an initiator receives the start signal, updates its state from 0 to 1, and sends the registration request $\langle ID_i, RPW_i\rangle$ to the RA using $Snd(\,)$ channel securely. The RA accepts the registration request of U_i, and sends information $\langle A_i,\ TID_i\rangle$ to U_i using $Snd(\,)$ channel securely. U_i then receives information $\langle A_i, TID_i\rangle$ using $Rcv(\,)$ channel securely. U_i sends the login request $\langle TID_i,\ M_2,\ M_3,\ T_1\rangle$ to the GWN using public channel. The GWN further sends the authentication request $\langle M_7,\ M_8,\ T_2\rangle$ to SD_j using public channel. The SD_j also sends reply message $\langle M_{10},\ M_{11},\ M_{12}, T_3\rangle$ to the GWN using public channel. Finally, the GWN sends authentication reply $\langle M_{14}, M_{15}, M_{16}, T_3, T_4\rangle$ to U_i using public channel. Both $Snd(\,)$ and $Rcv(\,)$ public channels use Dolev-Yao threat model type [9]. So, an intruder (always denoted by (i)) can read, modify or delete the contents of exchanged messages. Similarly, we also have specified the roles for RA, GWN and SD_j in our HLPSL implementation.

In the session role specified in Fig. 6, all the basic roles are started with concrete arguments. Fig. 6 also consists of top level $environment$ role, which is the starting point for the execution. At the end, in the goal section, four authentication goals and two secrecy goals are specified.

The declaration $witness(Ui, GWN, ui\_gwn\_t1, T1')$ says that U_i has freshly generated the current timestamp T_1 for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2017.2764083, IEEE Transactions on Dependable and Secure Computing

GWN. The declaration $request(GWN, Ui, gwn\_ui\_t4, T4')$ expresses U_i's acceptance of timestamp T_4 generated for U_i by GWN. The declaration $secret(\{PWi, A', R'\}, sr1, Ui)$ also says that the information PW_i, a and r are only known to U_i. This is specified with protocol id $sr1$ in the goal section (given in Fig. 6).



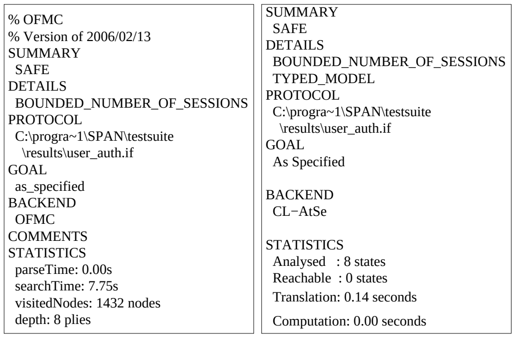
```
role session (Ui, RA, GWN, SDj: agent, H: hash_func, SKuira: symmetric_key)
def=
  local  S1, R1, S2, R2, S3, R3, S4, R4: channel (dy)
  composition
    user (Ui, RA, GWN, SDj, H, SKuira, S1, R1)
 /\ registrationauthority(Ui, RA, GWN, SDj, H, SKuira, S2, R2)
 /\ gatewaynode (Ui, RA, GWN, SDj, H, SKuira, S3, R3)
 /\ smartdevice (Ui, RA, GWN, SDj, H, SKuira, S2, R2)
end role

role environment()
def=
 const ui, ra, gwn, sdj: agent, h: hash_func, skuira: symmetric_key,
      kgwnui, kgwnsdj, idi, idsnj, idgwn, t1, t2, t3, t4, tidi, tidinew: text,
      gen, rep: hash_func, ui_gwn_t1, ui_gwn_rui, gwn_sdj_t2, gwn_sdj_rgwn,
      sdj_gwn_t3, sdj_gwn_rsdj, sr1, s2: protocol_id
intruder_knowledge ={t1, t2, t3, t4, h, gen, rep}
composition
 session(ui, ra, gwn, sdj, h, skuira) /\ session(i, ra, gwn, sdj, h, skuira)
/\ session(ui, i, gwn, sdj, h, skuira) /\ session(ui, ra, i, sdj, h, skuira)
/\ session(ui, ra, gwn, i, h, skuira)
end role

goal
  secrecy_of sr1, sr2
  authentication_on ui_gwn_t1, ui_gwn_rui, gwn_sdj_t2
  authentication_on gwn_sdj_rgwn, sdj_gwn_t3, sdj_gwn_rsdj
  authentication_on gwn_ui_t4, gwn_ui_tidinew
end goal
environment()
```

Fig. 6. The session, goal and environment roles in HLPSL

We have simulated our scheme using the widely-used OFMC and CL-AtSe backends. The executability check on non-trivial HLPSL specifications, replay attack check, and Dolev-Yao model check are verified in the proposed scheme. For more details on these verifications, one can refer to [31], [40]. The simulation results shown in Fig. 7 depicts that the proposed scheme is secure against replay as well as man-in-the-middle attacks.



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite
    \results\user_auth.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 7.75s
  visitedNodes: 1432 nodes
  depth: 8 plies
```

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  C:\progra~1\SPAN\testsuite
    \results\user_auth.if
GOAL
  As Specified

BACKEND
  CL−AtSe

STATISTICS
  Analysed   : 8 states
  Reachable  : 0 states
  Translation: 0.14 seconds
  Computation: 0.00 seconds
```

Fig. 7. The results of the analysis using OFMC and CL-AtSe backends

5.3 Informal Security Analysis

The informal security analysis shows that the following other possible known attacks are prevented.

5.3.1 *Traceability*

In many applications, it is desirable that a user authentication should not allow an adversary to trace a user during login and authentication phases. Therefore, it also becomes important that the identity of the user should no be revealed to an adversary to preserve the privacy of that user in a network, especially in a smart home environment. The login request $\langle TID_i, M_2, M_3, T_1\rangle$ sent by U_i to the GWN is different each time due to the following reason. The smart phone SP_i of U_i computes $M_1 = A^{*} \oplus RPW_i^{*} = h(ID_i||\ K_{GWN-U_i})$, $M_2 = M_1 \oplus\ r_{U_i}$ and $M_3 = h(M_2||\ T_1||\ ID_i||\ TID_i||\ r_{U_i})$, where T_1 is current timestamp and r_{U_i} random nonce of U_i. The involvement of T_1 and r_{U_i} ensures that M_2 and M_3 are distinct for each session. Moreover, other exchanged messages $\langle M_7, M_8, T_2\rangle$, $\langle M_{10}, M_{11}, M_{12}, T_3\rangle$ and $\langle M_{14}, M_{15}, M_{16}, T_3, T_4\rangle$ are also different for each session due to the use of timestamps and random nonces. In addition, our scheme allows to update old TID_i with a new TID_i^{new} for each session while the message $\langle M_{14}, M_{15}, M_{16}, T_3, T_4\rangle$ is sent to U_i by the GWN. After receiving the message, SP_i of the user U_i calculates $TID_i^{new} = M_{15} \oplus\ h(TID_i\ ||M_1\ ||T_3\ ||T_4)$ and replaces TID_i with TID_i^{new} in its memory. Due to this, TID_i in the login request messages are distinct for different sessions. Thus, our scheme avoids traceability of U_i and SD_j by an attacker.

5.3.2 *Anonymity*

Prior to sending the login request $\langle TID_i, M_2, M_3, T_1\rangle$ to the GWN, U_i hides its identity ID_i in $M_1 = A^{*} \oplus RPW_i^{*} = h(ID_i\ ||K_{GWN-U_i})$, M_2 and M_3. The GWN also hides the identities of U_i and SD_j as it computes $M_6 = h(ID_{SD_j}||\ K_{GWN-SD_j})$, $M_7 = E_{M_6}[ID_i, ID_{GWN}, r_{U_i}, r_{GWN}, h(M_4)]$ and $M_8 = h(M_6\ ||T_2\ ||ID_i\ ||ID_{SD_j}\ ||ID_{GWN}\ ||r_{GWN})$ and $M_{14} = E_{M_4}\ [r_{U_i}, r_{GWN}, r_{SD_j}, ID_{SD_j}, ID_{GWN}, h(M_6)]$. SD_j also hides its own identity by computing $M_{10} = h(h(ID_{SD_j}\ ||K_{GWN-SD_j})\ ||T_3) \oplus\ r_{SD_j}$. If an attacker intercepts all the messages during login and authentication phases, he/she is unable to identify ID_i and ID_{SD_j} as these are protected by symmetric encryption and one-way cryptographic hash function $h(\cdot)$. Therefore, the user and smart device anonymity are preserved in our scheme.

5.3.3 *Privileged-Insider Attack*

Suppose $\mathcal{A}$ is a malicious insider user of the RA, who knows ID_i and RPW_i, which were sent to RA by U_i during his/her registration phase. Note that $RPW_i = h(PW_i\ ||\sigma_i\ ||a) \oplus r$. We assume that $\mathcal{A}$ obtains the smart phone SP_i of U_i only after the user registration phase is finished. $\mathcal{A}$ can then extract all the information $\{TID_i, A_i^{*}, B_i, C_i, \tau_i, h(\cdot), Gen(\cdot), Rep(\cdot), t\}$ stored in SP_i using the power analysis attacks [41]. Note that the user U_i already deleted the information A_i from its smart phone SP_i at the end of the user registration phase described in Section 4.2. Hence, without having A_i, it is computationally hard for $\mathcal{A}$ to derive the secret r as $r = A_i^{*} \oplus A_i$. As a result, without r, $\mathcal{A}$ can not derive $h(PW_i\ ||\sigma_i\ ||a) = RPW_i \oplus r$. Furthermore, without knowing a, it is computationally infeasible to derive the biometric key σ_i as $h(ID_i||\sigma_i) = B_i \oplus a$. As a consequence, without having a, σ_i and K_{GWN-U_i}, it is also computationally hard for $\mathcal{A}$ to guess correctly the password PW_i of U_i from $C_i = h(ID_i\ ||RPW_i'\ ||\sigma_i) = h(ID_i\ ||(h(ID_i\ ||K_{GWN-U_i}) \oplus h(PW_i\ ||\sigma_i\ ||a))\ ||\sigma_i)$. In summary, it is computationally hard for $\mathcal{A}$ to guess and verify correctly PW_i and σ_i from RPW_i, A_i^{*}, B_i and C_i due to the collision resistant property of $h(\cdot)$. Therefore, our scheme is secure against the privileged-insider attack.

5.3.4 Stolen Smart Phone Attack

Suppose the smart phone SP_i of U_i is lost or stolen by an attacker $\mathcal{A}$. $\mathcal{A}$ can then extract all information $\langle TID_i,\ A_i^*,\ B_i,\ C_i,\ \tau_i,\ h(\cdot),\ Gen(\cdot),\ Rep(\cdot),\ t\rangle$ stored in SP_i using the power analysis attacks [41]. Note that $B_i = h(ID_i||\sigma_i) \oplus a$, $RPW_i' = RPW_i \oplus r = h(PW_i\ ||\sigma_i\ ||a)$, $C_i = h(ID_i\ ||RPW_i'\ ||\sigma_i)$ and $A_i^* = A_i \oplus\ r = h(ID_i\ ||K_{GWN-U_i})\ \oplus RPW_i'$. To correctly guess ID_i and PW_i from B_i and C_i respectively, $\mathcal{A}$ needs to know both a and r. Again, to know a from B_i, $\mathcal{A}$ needs both ID_i and PW_i. Thus, it is computationally infeasible for $\mathcal{A}$ to correctly guess both ID_i and PW_i as ID_i and PW_i are protected by the one-way hash function $h(\cdot)$. Therefore, our scheme is secure against such an attack.

5.3.5 Session Key Security

The session key $SK_{ij} = h[ID_i||\ ID_{SD_j}\ ||ID_{GWN}\ ||r_{U_i}\ ||r_{GWN}\ ||r_{SD_j}\ ||h(M_4)\ ||h(h(ID_{SD_j}\ ||K_{GWN-SD_j}))]$ is calculated by both U_i and SD_j. The message $\{M_{10},\ M_{11},\ M_{12},\ T_3\}$ sent by SD_j to GWN contains session key SK_{ij} as $M_{11} = h(SK_{ij}\ ||T_3)$. Suppose an attacker $\mathcal{A}$ intercepts this message and tries to compute the session key $SK_{ij}' = h[ID_i\ ||ID_{SD_j}\ ||ID_{GWN}\ ||r_{U_i}\ ||r_{GWN}'\ ||r_{SD_j}'\ ||h(M_4)\ ||h(h(ID_{SD_j}\ ||K_{GWN-SD_j}))]$ by generating the random nonces r_{U_i}', r_{GWN}', r_{SD_j}' and timestamp T_3'. However, the computation of SK_{ij}' is not possible for $\mathcal{A}$ because he/she does not know the various identities ID_i, ID_{SD_j}, ID_{GWN}, secret key K_{GWN-SD_j}, $M_4' = h(ID_i\ ||K_{GWN-U_i})$. Without the knowledge of these parameters, and due to the collision resistance property of $h(\cdot)$, it is very difficult for $\mathcal{A}$ to obtain SK_{ij}'. Therefore, our scheme preserves the session key security.

5.3.6 User Impersonation Attack

Suppose there is an adversary $\mathcal{A}$, who has the lost/stolen smart phone SP_i of a legal user U_i, and knows all the information stored in SP_i by the help of power analysis attacks [41]. Assume that $\mathcal{A}$ intercepts U_i's login request $\langle TID_i,\ M_2,\ M_3,\ T_1\rangle$ and tries to create another valid login request, say $\langle TID_i,\ M_2',\ M_3',\ T_1'\rangle$ on behalf of U_i, using the current timestamp T_1' of his/her system. To compute M_2', M_1' is required to compute as $M_1' = A^* \oplus\ RPW_i^* = h(ID_i\ ||K_{GWN-U_i})$. Suppose $\mathcal{A}$ generates random nonce r_{U_i}'. To calculate $M_2' = M_1' \oplus\ r_{U_i}'$ and $M_3' = h(M_2'\ ||T_1'||\ ID_i||\ TID_i\ ||r_{U_i}')$, $\mathcal{A}$ needs ID_i and K_{GWN-U_i}, which are infeasible for him/her to obtain them. Due to the one-way hash function $h(\cdot)$, it is computationally infeasible for $\mathcal{A}$ to create valid login request $\langle TID_i,\ M_2',\ M_3',\ T_1'\rangle$ on behalf of U_i, even he/she knows the all information from the lost/stolen SP_i. So, it is clear that our scheme is secure against the user impersonation attack.

5.3.7 GWN Impersonation Attack

Suppose an adversary $\mathcal{A}$ intercepts the messages $\langle M_7,\ M_8,\ T_2\rangle$ and $\langle M_{14}, M_{15}, M_{16}, T_3, T_4\rangle$, and attempts to create other valid messages, say $\langle M_7',\ M_8',\ T_2'\rangle$ and $\langle M_{14}',\ M_{15}',\ M_{16}',\ T_3',\ T_4'\rangle$ on behalf of the GWN, where $M_7 = E_{M_6}\ [ID_i,\ ID_{GWN},\ r_{U_i},\ r_{GWN},\ h(M_4)]$, $M_6 = h(ID_{SD_j}\ ||K_{GWN-SD_j})$, $M_4 = h(ID_i\ ||K_{GWN-U_i})$ and $M_8 = h(M_6\ ||T_2\ ||ID_i\ ||ID_{SD_j}\ ||ID_{GWN}\ ||r_{GWN})$, $M_{14} = E_{M_4}\ [r_{U_i}, r_{GWN}, r_{SD_j}, ID_{SD_j}, ID_{GWN}, h(M_6)]$, $M_{15} = TID_i^{new} \oplus h(TID_i\ ||M_4\ ||T_3\ ||T_4)$, $M_{16} = h(M_{11}\ ||T_4\ ||r_U)$. Suppose T_2', T_3', T_4' and r_{U_i}', r_{GWN}', r_{SD_j}' are the current timestamps and different random nonces generated by $\mathcal{A}$. To compute M_7', M_6', M_4' and M_8', the secret key K_{GWN-SD_j}, and various identities ID_i, ID_{SD_j} and ID_{GWN} are required. To calculate M_{14}', M_{15}' and M_{16}', the secret key K_{GWN-U_i}, and various identities TID_i, ID_i, ID_{SD_j} and ID_{GWN}' are required. Moreover, the messages are protected by the one-way hash function $h(\cdot)$. Thus, $\mathcal{A}$ is not able to create other valid messages $\langle M_7',\ M_8',\ T_2'\rangle$, $\langle M_{14}',\ M_{15}',\ M_{16}',\ T_3',\ T_4'\rangle$ on behalf of the GWN. Therefore, the proposed scheme is secure against the GWN impersonation attack.

5.3.8 Smart Device Impersonation Attack

Suppose an adversary $\mathcal{A}$ intercepts the message $\langle M_{10}, M_{11}, M_{12}, T_3\rangle$ and attempts to create another valid message, say $\langle M_{10}',\ M_{11}',\ M_{12}',\ T_3'\rangle$ on behalf of the smart device SD_j, where T_3' is the current timestamp of $\mathcal{A}$'s system when this message is created. Note that $M_{10}' = h(h(ID_{SD_j}\ ||K_{GWN-SD_j})\ ||T_3')\ \oplus r_{SD_j}'$, $M_{11}' = h(SK_{ij}'\ ||T_3')$, $SK_{ij}' = h[ID_i\ ||ID_{SD_j}\ ||ID_{GWN}\ ||r_{U_i}'\ ||r_{GWN}'\ ||r_{SD_j}'\ ||h(M_4)\ ||h(h(ID_{SD_j}\ ||K_{GWN-SD_j}))]$, $M_{12}' = h(r_{SD_j}'\ ||r_{GWN}'\ ||ID_{SD_j}\ ||ID_{GWN}\ ||T_3')$ and $M_4' = h(ID_i\ ||K_{GWN-U_i})$, where r_{U_i}', r_{GWN}' and r_{SD_j}' are the random nonces created by $\mathcal{A}$. To calculate M_{10}', M_{11}' and M_{12}', the secret keys K_{GWN-SD_j} and $h(ID_i\ ||K_{GWN-U_i})$, and various identities ID_i, ID_{SD_j} and ID_{GWN} are necessary. Therefore, $\mathcal{A}$ is not able to create another valid message $\langle M_{10}',\ M_{11}',\ M_{12}',\ T_3'\rangle$ on behalf of SD_j. This confirms that the proposed scheme is secure against this attack.

5.3.9 Resilience against Smart Device Capture Attack

Suppose a smart device SD_j is physically captured by an attacker $\mathcal{A}$. Each SD_j contains the information $\{ID_{SD_j},\ h(ID_{SD_j}\ ||K_{GWN-SD_j})\}$. Since each K_{GWN-SD_j} is distinct, $h(ID_{SD_j}\ ||K_{GWN-SD_j})$ is also distinct for each SD_j. If $\mathcal{A}$ tries to extract K_{GWN-SD_j} from $h(ID_{SD_j}||K_{GWN-SD_j})$ using ID_{SD_j}, it is difficult task for $\mathcal{A}$ to compute K_{GWN-SD_j} as K_{GWN-SD_j} is a long 1024-bit secret key. However, $\mathcal{A}$ can know the session key SK_{ij} shared with the legal user U_i, which is stored in SD_j's memory. Thus, compromise of this particular smart device SD_j in the smart home network does not lead to compromise of the session keys between that U_i and other non-compromised smart devices SD_l's as the stored $h(ID_{SD_l}\ ||K_{GWN-SD_l})$ is distinct for SD_l. The proposed scheme is then unconditionally secure against this attack.

5.3.10 Gateway Bypass Attack

In our scheme, both U_i and SD_j can not bypass the GWN due to the following argument. U_i can only send the login request through the GWN, and SD_j can send the authentication response only through the GWN. Both U_i and SD_j also establish the session key SK_{ij} through the GWN. When the GWN receives login request from U_i, it computes $M_7 = E_{M_6}\ [ID_i, ID_{GWN},\ r_{U_i}^*,\ r_{GWN},\ h(M_4)]$ and $M_8 = h(M_6\ ||T_2\ ||ID_i\ ||ID_{SD_j}\ ||ID_{GWN}\ ||r_{GWN})$ and sends $\langle M_7, M_8, T_2\rangle$ to SD_j, where $M_6 = h(ID_{SD_j}\ ||K_{GWN-SD_j})$, and T_2 is the current timestamp generated by U_i. U_i can not compute M_6 as he/she does not know K_{GWN-SD_j} and it is only known to the GWN. Therefore, U_i is not able to compute M_7 and M_8. When the GWN receives authentication reply from SD_j, it computes $M_{14} = E_{M_4}\ [r_{U_i}^*,\ r_{GWN},\ r_{SD_j}^*,\ ID_{SD_j},\ ID_{GWN},\ h(M_6)]$, $M_{15} = TID_i^{new} \oplus h(TID_i\ ||M_4\ ||T_3\ ||T_4)$, $M_{16} = h(M_{11}\ ||T_4\ ||r_{U_i}^*)$ and sends the message $\langle M_{14},\ M_{15},\ M_{16},\ T_3,\ T_4\rangle$ to U_i. SD_j can not compute M_4 as he/she does not know K_{GWN-U_i}. Therefore,

SD_j can not compute M_{14} and M_{15}. To compute M_{16}, even if SD_j chooses current timestamp T'_4 to compute $M_{16} = h(M_{11} ||T'_4 ||r^*_{U_i})$, but he/she does not know the random nonce r_{U_i*} of the user U_i. So, SD_j can not compute M_{14}, M_{15} and M_{16}. As a result, neither U_i nor GWN bypass the GWN in our proposed scheme.

5.3.11 *Offline-Dictionary Attack*

We consider an interesting attack scenario in our proposed scheme as illustrated by Huang *et al.* [34] to verify whether an adversary $\mathcal{A}$ can derive the password of a legal user U_i or not. As in [34], we also consider the following attacking scenario as follows.

- At time T_1, suppose U_i invokes the password and biometric update phase to change the password to PW_{i1}. At the end of this phase, the smart phone SP_i of U_i contains the information $\langle TID_i,\ A^*_i,\ B_i,\ C_i,\ \tau_i,\ h(\cdot),\ Gen(\cdot),\ Rep(\cdot), t\rangle$, where $A^*_i = h(ID_i ||K_{GWN-U_i}) \oplus h(PW_{i1} ||\sigma_{i1} ||a)$ and σ_{i1} is the biometric key derived from the new biometrics BIO_{i1} entered by U_i at this time.
- At some time later (say, T_2), U_i again changes his/her password PW_1 to a new password PW_2. At the end of this phase, the SP_i of U_i contains the information $\langle TID_i,\ A^{**}_i,\ B_i,\ C_i,\ \tau_i,\ h(\cdot),\ Gen(\cdot),\ Rep(\cdot),\ t\rangle$, where $A^{**}_i = h(ID_i ||K_{GWN-U_i}) \oplus h(PW_{i2} ||\sigma_{i2} ||a)$ and σ_{i2} is the biometric key derived from the new biometrics BIO_{i2} entered by U_i at this time T_2.
- A passive adversary $\mathcal{A}$ with smart phone can obtain the data stored in the smart phone at time T_1 and T_2.

Now, given (A^*_i, A^{**}_i), $\mathcal{A}$ can calculate $A^*_i \oplus\ A^{**}_i = h(PW_{i1} ||\sigma_1 ||a)\ \oplus h(PW_{i2} ||\sigma_{i2} ||a)$. By testing all password pairs in the password dictionary, $\mathcal{A}$ can try to find at least one pair (pw_1, pw_2) such that $A^*_i \oplus\ A^{**}_i = h(pw_1 ||\sigma_{i1} ||a)\ \oplus h(pw_2 ||\sigma_{i2} ||a)$. However, to satisfy this condition, $\mathcal{A}$ further needs to guess correctly the biometric keys pair $(\sigma_{i1}, \sigma_{i2})$. In addition, $\mathcal{A}$ also needs the random secret a which is only known to U_i. To derive a, $\mathcal{A}$ requires to guess the biometric key too. Thus, without having the biometric keys pair $(\sigma_{i1}, \sigma_{i2})$ and random secret a, it is computationally infeasible problem for $\mathcal{A}$ to verify whether the guessed passwords pair (pw_1, pw_2) is correct or not. As a result, the proposed scheme has the ability to protect the offline-dictionary attack described in [34].

6 Practical Perspective: NS2 Simulation

The proposed scheme is simulated using the widely-accepted networking simulation tool, NS2 2.35 simulator [16] on Ubuntu 14.04 LTS platform.

6.1 Simulation Parameters

The various simulation parameters are given in Table 2. The network coverage area is taken as $400 \times 200\ m^2$. The communication ranges of the gateway node (GWN) and smart devices (SD_j) are taken as $200m$ and $50m$, respectively. The network simulation time is taken as 1800 seconds (30 minutes). The traditional Ad hoc On-Demand Distance Vector (AODV) routing protocol is used as the routing protocol. Two types of users are taken in the simulation: first type consists of the static users, who do not move (for example, some smart home users seat on the chair and access SD_j), while the second type has moving users (for example, somebody is walking in the garden and accessing SD_j, or somebody is driving the card and accessing SD_j). The speeds for these smart home users are considered as 2, 10 and 15 mps, respectively.

6.2 Simulation Environment

We have considered the following three network scenarios in the simulation. For all the scenarios, we have taken one GWN and 50 SD_js.

Scenario 1. In this case, we have taken two users $(U_i$s): one is static and other one is moving with 2 mps.

Scenario 2. In this case, we have taken three users $(U_i$s): one is static and other two are moving with the speeds of 2 mps and 15 mps, respectively.

Scenario 3. In this case, we have taken eight users $(U_i$s): four are static and other four are moving with the speeds of 2 mps, 2 mps, 10 mps and 15 mps, respectively.

Moreover, we assume that the bit lengths of the identity, hash output (if we use SHA-1 hash algorithm) and random number/nonce are 128, 160 and 128 bits, respectively. In each scenario, we have considered the following messages between different network entities: $\langle TID_i, M_2, M_3, T_1\rangle$, $\langle M_7, M_8, T_2\rangle$, $\langle M_{10}, M_{11}, M_{12}, T_3\rangle$ and $\langle M_{14}, M_{15}, M_{16}, T_3, T_4\rangle$ of sizes 480 bits, 960 bits, 512 bits and 1280 bits, respectively.

TABLE 2
Various simulation parameters

Parameter	Description
Platform	Ubuntu 14.04 LTS
Network coverage area	$400 \times 200\ m^2$
Network scenarios	1, 2 and 3
Number of users (U_i)	$2, 3, 8$ for scenarios $1, 2, 3$
Number of gateway nodes (GWN)	1 for all scenarios
Number of smart devices (SD_j)	50 for all scenarios
Mobility	2 mps, 10 mps, 15 mps
Simulation time	1800 seconds
Routing protocol	AODV
Communication range of GWN	200 m
Communication range of SD_j	50 m

6.3 Simulation Results and Discussions

The network performance parameters, such as end-to-end delay (in seconds) and throughput (in bps) are calculated during the simulation.

6.3.1 *Impact on End-to-end Delay*

The end-to-end delay (EED) is calculated as the average time taken by the data packets to arrive at the destination from the source. The EEDs of our scheme for different scenarios are given Fig. 8(a). The EEDs are 0.29832, 0.28687 and 0.28637 seconds for the network scenarios 1, 2 and 3, respectively. Note that the EED decreases in the scenarios 2 and 3, because in these scenarios we have considered more number of mobile users who are traveling towards the gateway node as compared to the scenario 1. For this reason, the EED reduces as the distance between the gateway node and mobile users decreases which affects the reducibility of the EEDs accordingly.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2017.2764083, IEEE Transactions on Dependable and Secure Computing

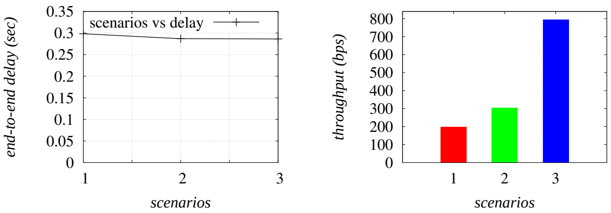


Fig. 8. (a) End-to-end delay (b) Throughput

6.3.2 Impact on Throughput

The throughput is measured as the number of bits transmitted per unit time. Fig. 8(b) depicts the network throughput (in bps) of our scheme under different network scenarios. The throughput values are 197.56, 303.87 and 793.78 bps for the scenarios 1, 2 and 3, receptively. Note that the throughput increases with an increase in the number of users. Due to the large number of users, more number of messages are exchanged in the network, and as a result, the throughput also increases.

7 Performance Comparison

In this section, the proposed scheme is compared with related existing schemes of Kumar *et al.* [5], Vaidya *et al.* [21], Kim and Kim [22], Jeong *et al.* [20], and Santoso and Vun [25] during the login, and authentication and key agreement phases. Since the registration, and password and biometric update phases are not frequent, the costs involved in these phases are not discussed.

The communication costs of different existing schemes and our scheme are compared in Table 3. We have made a reasonable assumption that the identities are 128 bits in length; random nonces are 128 bits; timestamps are 32 bits; plaintext/ciphertext block in symmetric encryption/decryption (using AES-CBC algorithm) is 128 bits, and the hash digest is of 160 bits (if we use SHA-1 as $h(\cdot)$ [42]). By considering these values, the communication costs for the schemes of Kumar *et al.*, Vaidya *et al.*, Kim-Kim, Jeong *et al.*, Santoso-Vun and our scheme are 1696, 2272, 4352, 1568, 4416, and 3232 bits, respectively. Note that in our scheme, the messages $MSG_1 = \langle TID_i, M_2, M_3, T_1\rangle$, $MSG_2 = \langle M_7, M_8, T_2\rangle$, $MSG_3 = \langle M_{10}, M_{11}, M_{12}, T_3\rangle$, $MSG_4 = \langle M_{14}, M_{15}, M_{16}, T_3, T_4\rangle$ are used. The cost of M_7 is $\lceil(128+ 128+ 128+ 128+ 160)\ /128\rceil \times 128 = 768$ bits. Similarly, M_{14} needs $\lceil(128 +128 +128 +128 +128 +160)\ /128\rceil \times 128 = 896$ bits. So, the communication costs of different messages MSG_1, MSG_2, MSG_3 and MSG_4 are 480 bits, 960 bits, 512 bits, and 1280 bits, respectively. As a result, the total communication cost of the proposed scheme turns out to be $(480+ 960+ 512+ 1280) = 3232$ bits. Though our scheme requires more communication cost as compared to that for the schemes of Kumar *et al.*, Vaidya *et al.* and Jeong *et al.*, it is justified as our scheme supports additional functionality and security features (see Table 5).

In Table 4, we have used the notations T_{exp}, T_E/T_D, T_h, T_{fe}, T_{mac} and T_{hmac} to denote the computational time for modular exponentiation operation, symmetric encryption/decryption, hash function $h(\cdot)$ (using SHA-1 hashing algorithm), $Gen(\cdot)/Rep(\cdot)$, message authentication code (MAC) and hashed MAC, respectively. The bitwise XOR operation execution time is negligible, and we do not consider it as a performance evaluation parameter. The existing experimental values of these operations are given as follows in [43], [44]: T_{exp}, T_h, T_E/T_D, and T_{fe} are $0.0192s$, $0.00032s$, $0.0056s$ and $0.0171s$, respectively. It is further assumed that $T_{mac} \approx T_{hmac} \approx T_h$. The computational costs of various schemes are given in Table 4. The total computational cost for our scheme is $22T_h+\ 4T_E/T_D\ +T_{fe}$, whereas the computational cost for a smart device is $7T_h+\ T_D \approx 7.84ms$ only. This indicates that our scheme is suitable for resource-constrained smart devices. The computation cost of our scheme is more than that for the schemes of Kumar *et al.*, Vaidya *et al.*, Kim-Kim and Jeong *et al.*, because we have used the fuzzy extractor for providing additional security level of the system as compared to other schemes. However, our scheme provides extra functionality features and security features, and the cost for a resource constrained smart device is low.

TABLE 3
Communication cost comparisons

Scheme	Total messages	Total cost $(bits)$
Kumar *et al.* [5]	3	1696
Vaidya *et al.* [21]	2	2272
Kim-Kim [22]	2	4352
Jeong *et al.* [20]	2	1568
Santoso-Vun [25]	3	4416
Our	4	3232

TABLE 4
Computation costs comparison

Scheme/phase	Total cost	Rough estimation
Kumar *et al.* [5]	$2T_h + T_{mac}$ $+1T_{hmac} + 2T_E/T_D$	12.48 ms
Vaidya *et al.* [21]	$20T_h + 3T_E/T_D$	23.20 ms
Kim-Kim [22]	$30T_h + 3T_E/T_D$	26.40 ms
Jeong *et al.* [20]	$10T_h + 3T_E/T_D$	20.00 ms
Santoso-Vun [25]	$2T_h + 3T_{exp}$	58.24 ms
Our	$22T_h + 4T_E/T_D + T_{fe}$	46.54 ms

Finally, the functionality and security features comparison among our scheme and other schemes is shown in Table 5. The scheme of Vaidya *et al.* is insecure against privileged-insider, password guessing, and smart device capture attacks, and it does not have the traceability, user anonymity and smart device anonymity properties. Moreover, the dynamic smart device addition phase, offline smart device registration phase, formal security proof under standard model and formal security verification using AVISPA are not supported in their scheme. Kim-Kim's scheme is vulnerable to password guessing attack, password change attcak, privileged-insider attack, user impersonation attack through privileged-insider attack and smart device capture attack, and it does not have traceability, user anonymity and smart device anonymity properties. Additionally, the dynamic smart device addition phase, offline smart device registration phase, formal security proof under the ROR model and formal security verification using AVISPA are not available in Kim-Kim's scheme. Kumar *et al.* does not support traceability and gateway anonymity properties and it does not provide formal security proof under the ROR model. The schemes of Kumar *et al.*, Jeong *et al.* and Santoso-Vun also lack the functionality features, which are shown in Table 5. In summary, our scheme provides significantly better security and functionality features as compared to those for other existing schemes.

8 Conclusion

This paper presents a new scheme to address the user authentication issue in a smart home environment. The proposed scheme

TABLE 5
Security and functionality features comparison

Functionality features	[5]	[21]	[22]	[20]	[25]	Our
SFF_1	✓	×	×	×	×	✓
SFF_2	×	×	×	×	✓	✓
SFF_3	×	✓	✓	✓	×	✓
SFF_4	✓	✓	✓	✓	✓	✓
SFF_5	×	×	×	×	×	✓
SFF_6	N/A	×	×	×	×	✓
SFF_7	N/A	✓	×	×	×	✓
SFF_8	✓	×	×	×	×	✓
SFF_9	N/A	×	×	×	×	✓
SFF_{10}	×	×	✓	✓	×	✓
SFF_{11}	✓	×	×	×	×	✓
SFF_{12}	✓	✓	✓	✓	✓	✓
SFF_{13}	✓	×	×	×	✓	✓
SFF_{14}	✓	✓	✓	✓	✓	✓
SFF_{15}	N/A	×	✓	×	×	✓
SFF_{16}	N/A	✓	×	×	×	✓
SFF_{17}	✓	✓	✓	✓	✓	✓
SFF_{18}	✓	✓	✓	✓	✓	✓
SFF_{19}	✓	✓	✓	✓	×	✓
SFF_{20}	✓	×	×	×	×	✓
SFF_{21}	✓	×	×	×	×	✓
SFF_{22}	N/A	✓	✓	✓	×	✓
SFF_{23}	N/A	×	×	×	×	✓
SFF_{24}	×	×	×	×	×	✓
SFF_{25}	✓	×	×	×	×	✓

Note: SFF_1: mutual authentication between GWN and smart device; SFF_2: mutual authentication between user and smart device; SFF_3: mutual authentication between user and GWN; SFF_4: key agreement; SFF_5: traceability property; SFF_6: password guessing attack; SFF_7: password change attack; SFF_8: dynamic smart device addition phase; SFF_9: user anonymity property; SFF_{10}: GWN anonymity property; SFF_{11}: smart device anonymity property; SFF_{12}: replay attack; SFF_{13}: privileged-insider attack; SFF_{14}: man-in-the-middle attack; SFF_{15}: stolen smart phone/smart card attack; SFF_{16}: user impersonation attack; SFF_{17}: smart device impersonation attack; SFF_{18}: GWN bypassing attack; SFF_{19}: DoS attack; SFF_{20}: resilient against smart device capture attack; SFF_{21}: offline smart device registration phase; SFF_{22}: password change phase; SFF_{23}: biometric update phase; SFF_{24}: formal security proof under ROR model; SFF_{25}: formal security verification using AVISPA.
✓: the scheme is secure or supports a particular functionality/security feature; ×: the scheme is not secure or does not support a particular functionality/security feature. N/A: not applicable in the scheme.

provides additional functionality features. The proposed scheme is secure against several known attacks, which are shown through random oracle model, informal security and AVISPA tool. The practical implementation of the proposed scheme is also demonstrated though the widely-accepted NS-2 simulator. Overall, the proposed scheme provides a better trade-off between security and functionality features provided in Table 5, and overheads as compared to other existing related schemes.

ACKNOWLEDGMENTS

We thank the anonymous reviewers and the Associate Editor for providing constructive and generous feedback. This work was supported by the Information Security Education & Awareness (ISEA) Phase II Project, Department of Electronics and Information Technology (DeitY), India.

REFERENCES

[1] "What is a Smart Home?" http://smarthomeenergy.co.uk /what-smart-home. Accessed on April 2016.

[2] C. Gomez and J. Paradells, "Wireless home automation networks: A survey of architectures and technologies," *IEEE Communications Magazine*, vol. 48, no. 6, pp. 92–101, 2010.

[3] J. E. Kim, G. Boulos, J. Yackovich, T. Barth, C. Beckel, and D. Mosse, "Seamless integration of heterogeneous devices and access control in smart homes," in *8th International Conference on Intelligent Environments (IE)*. Guanajuato, Mexico: IEEE, 2012, pp. 206–213.

[4] A. Lazakidou, *Wireless technologies for ambient assisted living and healthcare: systems and applications: Systems and applications*. IGI Global, 2010.

[5] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, and M. Sain, "Lightweight and Secure Session-Key Establishment Scheme in Smart Home Environments," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 254–264, Jan 2016.

[6] R. Volner, P. Bore, and V. Smrz, "A product based security model for smart home appliances," in *11th International Biennial Baltic Electronics Conference*, Tallinn, Estonia, 2008, pp. 221–222.

[7] H. Tschofenig, J. Arkko, and D. McPherson, "Architectural considerations in smart object networking, Internet Engineering Task Force, RFC-7452," *Internet Engineering Task Force, Fremont, CA, USA*, 2014.

[8] I. Bierhoff *et al.*, *Smart home environment*, 2007, Towards an inclusive future–Impact and wider potential of information and communication technologies, East Sussex Press, Brussels, Belgium.

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

[10] E. Bertino, N. Shang, and S. S. W. Jr., "An Efficient Time-Bound Hierarchical Key Management Scheme for Secure Broadcasting," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 2, pp. 65–70, April 2008.

[11] S. Frankel, R. Glenn, and S. Kelly, "The AES-CBC Cipher Algorithm and Its Use with IPsec," 2013, http://tools.ietf.org/html/rfc3602. Accessed on July 2017.

[12] "Advanced Encryption Standard (AES)," FIPS PUB 197, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. Accessed on April 2016.

[13] R. Canetti, "Introduction to Cryptography. Lecture 9 - Symmetric Encryption," 2008, http://www.cs.tau.ac.il/ canetti/f08-materials/scribe9.pdf. Accessed on July 2017.

[14] M. Abdalla, P. Fouque, and D. Pointcheval, "Password-based authenticated key exchange in the three-party setting," in *8th International Workshop on Theory and Practice in Public Key Cryptography (PKC'05), Lecture Notes in Computer Science (LNCS)*, vol. 3386, Les Diablerets, Switzerland, 2005, pp. 65–84.

[15] AVISPA, "Automated Validation of Internet Security Protocols and Applications," http://www.avispa-project.org/. Accessed on April 2016.

[16] "The Network Simulator-ns-2," http://www.isi.edu/nsnam/ns/. Accessed on April 2016.

[17] P. Sarkar, "A Simple and Generic Construction of Authenticated Encryption with Associated Data," *ACM Transactions on Information and System Security*, vol. 13, no. 4, pp. 1–16, 2010, article No. 33.

[18] K.-K. R. Choo, "Key Establishment: Proofs and Refutations," Ph.D. dissertation, Queensland University of Technology, Brisbane, Australia, 2006, available at http://eprints.qut.edu.au/16262/1/Kim-Kwang_Choo_Thesis.pdf.

[19] S. Wu and K. Chen, "An Efficient Key-Management Scheme for Hierarchical Access Control in E-Medicine System," *Journal of Medical Systems*, vol. 36, no. 4, pp. 2325–2337, 2012.

[20] J. Jeong, M. Y. Chung, and H. Choo, "Integrated OTP-Based User Authentication Scheme Using Smart Cards in Home Networks," in *41st Annual Hawaii International Conference on System Sciences*, Waikoloa, HI, USA, 2008, pp. 294–294.

[21] B. Vaidya, J. H. Park, S. S. Yeo, and J. J. Rodrigues, "Robust one-time password authentication scheme using smart card for home network environment," *Computer Communications*, vol. 34, no. 3, pp. 326–336, 2011.

[22] H. J. Kim and H. S. Kim, "AUTH_HOTP - HOTP Based Authentication Scheme over Home Network Environment," in *International Conference on Computational Science and Its Applications*. Santander, Spain: Springer, 2011, pp. 622–637.

[23] B. Vaidya, D. Makrakis, and H. T. Mouftah, "Device authentication mechanism for Smart Energy Home Area Networks," in *IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2011, pp. 787–788.

[24] P. Hanumanthappa and S. Singh, "Privacy preserving and ownership authentication in ubiquitous computing devices using secure three way authentication," in *International Conference on Innovations in Information Technology (IIT)*. Al Ain, UAE: IEEE, 2012, pp. 107–112.

[25] F. K. Santoso and N. C. H. Vun, "Securing IoT for smart home system," in *International Symposium on Consumer Electronics (ISCE)*, Madrid, Spain, 2015, pp. 1–2.

[26] C.-C. Chang and H.-D. Le, "A Provably Secure, Efficient and Flexible Authentication Scheme for Ad hoc Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 357–366, 2016.

[27] A. K. Das, S. Kumari, V. Odelu, X. Li, F. Wu, and X. Huang, "Provably secure user authentication and key agreement scheme for wireless sensor networks," *Security and Communication Networks*, vol. 9, no. 16, pp. 3670–3687, 2016.

[28] Y. Li, "Design of a key establishment protocol for smart home energy management system," in *Fifth International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*. Madrid, Spain: IEEE, 2013, pp. 88–93.

[29] K. Han, J. Kim, T. Shon, and D. Ko, "A novel secure key paring protocol for RF4CE ubiquitous smart home systems," *Personal and Ubiquitous Computing*, vol. 17, no. 5, pp. 945–949, 2012.

[30] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques - Advances in Cryptology (Eurocrypt'04), Lecture Notes in Computer Science*, vol. 3027, Interlaken, Switzerland, 2004, pp. 523–540.

[31] V. Odelu, A. K. Das, and A. Goswami, "A Secure Biometrics-Based Multi-Server Authentication Protocol using Smart Cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[32] A. K. Das, "A Secure User Anonymity-Preserving Three-Factor Remote User Authentication Scheme for the Telecare Medicine Information Systems," *Journal of Medical Systems*, vol. 39, no. 3, 2015.

[33] A. K. Das, "A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 9, no. 1, pp. 223–244, 2016.

[34] X. Huang, X. Chen, J. Li, Y. Xiang, and L. Xu, "Further Observations on Smart-Card-Based Password-Authenticated Key Agreement in Distributed Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1767–1775, 2014.

[35] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "Secure Biometric-Based Authentication Scheme using Chebyshev Chaotic Map for Multi-Server Environment," *IEEE Transactions on Dependable and Secure Computing*, 2016, Accepted for publication. DOI: 10.1109/TDSC.2016.2616876.

[36] A. K. Das, "A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor," *International Journal of Communication Systems*, vol. 30, no. 1, pp. 1–25, 2017.

[37] A. Armando et al., "The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications," in *17th International Conference on Computer Aided Verification (CAV'05), Lecture Notes in Computer Science (LNCS), Springer-Verlag*, vol. 3576, Scotland, UK, 2005, pp. 281–285.

[38] AVISPA, "SPAN, the Security Protocol ANimator for AVISPA," http://www.avispa-project.org/. Accessed on October 2016.

[39] D. von Oheimb, "The high-level protocol specification language hlpsl developed in the eu project avispa," in *Proceedings of 3rd APPSEM II (Applied Semantics II) Workshop (APPSEM'05)*, Frauenchiemsee, Germany, 2005, pp. 1–17.

[40] V. Odelu, A. K. Das, and A. Goswami, "SEAP: Secure and Efficient Authentication Protocol for NFC Applications Using Pseudonyms," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 1, pp. 30–38, 2016.

[41] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541–552, 2002.

[42] "Secure Hash Standard," FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, April 1995. Available at http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf. Accessed on September 2015.

[43] D. He, N. Kumar, J. H. Lee, and R. S. Sherratt, "Enhanced three-factor security protocol for consumer USB mass storage devices," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 30–37, 2014.

[44] C.-C. Lee, C. T. Chen, P. H. Wu, and T. Y. Chen, "Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices," *IET Computers & Digital Techniques*, vol. 7, pp. 48–55, 2013.

Mohammad Wazid (S'17) received the M.Tech. degree in computer network engineering from Graphic Era University, Dehradun, India and the Ph.D. degree in computer science and engineering from the International Institute of Information Technology (IIIT), Hyderabad, India. His current research interests include security in wireless sensor network, vehicular adhoc network, Internet of Things (IoT) and cloud computing. He has published over 50 papers in international journals and conferences in the above areas.

Ashok Kumar Das (M'17) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering from IIT Kharagpur, India. He is currently an Assistant Professor with the Center for Security, Theory and Algorithmic Research, IIIT, Hyderabad, India. His current research interests include security in wireless sensor network, vehicular ad hoc networks, smart grid, Internet of Things (IoT) and cloud computing. He has authored over 145 papers in international journals and conferences in the above areas. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He is in the editorial board of KSII Transactions on Internet and Information Systems, and the International Journal of Internet Technology and Secured Transactions (Inderscience).

Vanga Odelu received the M.Tech. degree in computer science and data processing and Ph.D. degree from IIT Kharagpur, India. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Indian Institute of Information Technology, Sri City, India. His research interests include user authentication, security in cloud computing and smart grid. He has authored over 40 papers in international journals and conferences.

Neeraj Kumar (M'16) received the Ph.D. degree in computer science and engineering from Shri Mata Vaishno Devi University, Katra (J&K), India, in 2009. He was a Post-Doctoral Research Fellow at Coventry University, Coventry, U.K. He is currently an Associate Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. He has authored more than 160 technical research papers published in leading journals and conferences from the IEEE, Elsevier, Springer, John Wiley, including IEEE TIE, IEEE TDSC, IEEE TIFS, IEEE TCE, IEEE Network, IEEE Com, IEEE WC, IEEE IoTJ, IEEE SJ, FGCS, JNCA, and ComCom. He is in the editorial board of JNCA (Elsevier) and IJCS (Wiley).

Willy Susilo (SM'02) received the Ph.D. degree in computer science from the University of Wollongong, Australia. He is currently a Professor and the Head of the School of Computing and Information Technology with the University of Wollongong, Australia. He is also the Director of the Centre for Computer and Information Security Research with the University of Wollongong. He has been awarded the Prestigious ARC Future Fellow by the Australian Research Council. His main research interests include cloud security, cryptography, and information security. He has served as a Program Committee Member in major international conferences, including Asiacrypt and CT-RSA. He is the Editor-in-Chief of the Information journal. He is also an Associate Editor of the IEEE Transactions on Information Forensics and Security, Computer Standards & Interfaces and International Journal of Information Security.