Internet of Things 9 (2020) 100158

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

Secure remote anonymous user authentication scheme for smart home environment

Moneer Fakroon*, Mohammed Alshahrani, Fayez Gebali, Issa Traore

Electrical and Computer Engineering Department, University of Victoria, Victoria, Canada

ARTICLE INFO

Article history:
Received 13 November 2019
Revised 28 December 2019
Accepted 29 December 2019
Available online 9 January 2020

Keywords:
Smart home
Internet of Things (IoT)
User authentication
AVISPA

ABSTRACT

Smart home technology is an emerging application of Internet-of-Things (IoT) where the user can remotely control home devices. Since the user/home communication channel is insecure, an efficient and anonymous authentication scheme is required to provide secure communications in smart home environment. In this paper, we propose a new scheme for user authentication that combines physical context awareness and transaction history. The new scheme offers two advantages: it does not maintain a verification table and avoids clock synchronization problem. Communication overhead and computational cost of the proposed scheme are analyzed and compared with other related schemes. The security of the scheme is evaluated using three different methods: (1) formal analysis using the Burrows-Abadi-Needham logic (BAN); (2) informal analysis; (3) model check using the automated validation of internet security protocols and applications (AVISPA) tool.

1. Introduction

The Internet of Things (IoT) defines an ecosystem where each thing can be any physical or virtual object, identified and reached by other objects and showing smart capabilities. Such smart things are characterized by embedded electronic components that allow them to sense, compute, communicate and integrate seamlessly with the rest of the network. It has been forecast, based on Moore's law, that by 2025 the number of IoT devices will exceed 100 billion, distributed with an average of 1000 devices per person [1].

IoT devices are often resource-constrained, and deployed in unmonitored, physically unsecured environments. As such there is an urgent need to secure IoT infrastructure. One of the fundamental elements in securing an IoT infrastructure is around device identity and mechanisms to authenticate it. However, existing authentication mechanisms involve heavy computations which cannot be afforded by IoT devices, which as mentioned earlier are resource-constrained. Many IoT devices do not have the required compute power, memory or storage to support the current authentication protocols, which rely on computationally intensive cryptography algorithms, e.g., AES and RSA. Additionally, these authentication mechanisms also require a degree of user-intervention in terms of configuration and provisioning. However, many IoT devices will have limited access, thus requiring initial configuration to be protected from tampering, theft and other forms of compromise throughout its usable life, which in many cases could be years.

Recently, a large swathe of the Internet was brought down in a distributed denial of service (DDoS) attack carried out using the Mirai IoT botnet [2]. Mirai propagates by brute-forcing IoT device passwords via Telnet in a way that is much faster

* Corresponding author.
E-mail address: mfakroon@uvic.ca (M. Fakroon).

https://doi.org/10.1016/j.iot.2020.100158
2542-6605/

and less resource-intensive than traditional botnet. Hence, relying on only password-based solutions is not a viable option, as passwords can easily be broken, and many IoT devices do not provide an interface through which password authentication can take place. Consequently, there is a need to develop new mechanisms and standards to authenticate IoT actors in a robust way while taking into account the environmental and engineering constraints underlying the IoT ecosystem.

1.1. Motivations and contributions

Although some user authentication protocols have been proposed for remote smart home access, they are not lightweight or secure enough to be suitable for the smart home's IoT resource-constrained devices, and they did not consider physical context awareness (i.e., location awareness), and transaction history, which will indeed contribute in stopping known authentication attacks such as Mirai attack. Motivated by the importance of anonymous authentication service, and the inclusion of context-awareness (i.e., location awareness), and transaction history in the authentication process for the remote access of IoT smart homes, this paper aims at designing a secure and lightweight user authentication protocol that is suitable for IoT smart homes environment. The contributions of this paper are:

(a) We propose a lightweight and secure user mutual authentication and key agreement protocol for smart home access. The proposed protocol avoids having the verification table for authentication and the clock synchronization problem, which exists in the timestamp-based two-way authentication technique.
(b) We combine context-awareness (i.e., location awareness), and transaction history to improve the overall security of the user remote access of IoT smart homes.
(c) We prove the security of our proposed protocol using three different approaches: informal security analysis, the formal widely accepted BAN logic, and the formal AVISPA simulator tool.
(d) We compare our proposed protocol with other related protocols, comparison results demonstrate that our protocol is superior to the previously related protocol in terms of security and performance.

1.2. Roadmap of the paper

The rest parts of this paper is organized as follows. Section 2, offers a brief survey of existing schemes proposed in the literature. Section 3 presents a secure remote anonymous user authentication scheme and session key agreement for smart home environment. Section 4 provides the security analysis of the proposed scheme using formal and informal security analysis. Furthermore, this section also provides the formal security validation using the AVISPA tool. Section 5 gives performance comparison with the existing relevant schemes. Finally, Section 6 concludes the paper.

2. Related works

Jeong et al. [3] proposed a user authentication scheme based on one-time password (OTP) protocol. This scheme is lightweight because it uses one-way hash functions. However, the mutual authentication between Gateway node (*GWN*) and the smart device is not provided. Moreover, the anonymity and traceability properties are not achieved as the real identity of the user is sent in plain-text. In addition, the scheme is not immune from stolen smart card and privileged–insider attack.

Vaidya et al. [4] proposed one-time password authentication scheme for home network environment. This scheme is also lightweight as it uses only hash-chaining methods and hashed one-time password. Kim et al. [5] studied Vaidya et al.'s scheme and indicated that it does not provide user anonymity and forward secrecy. Furthermore, it is vulnerable to password guessing attack. an enhanced authentication scheme is proposed subsequently where they improved the weakness observed in Vaidya et al.'s scheme [4]. However, Kim et al.'s scheme also suffered from guessing attack, user impersonation attack and privileged-insider attack. Moreover, the anonymity and traceability properties were not achieved.

Santoso et al. [6] proposed a user authentication scheme for a smart home system based on elliptic curve cryptography (ECC). Similar to the schemes in [3–5], the anonymity and traceability properties were not provided. In addition, the scheme is not immune against privileged-insider attack and stolen smart card attack.

Kumar et al. [7] proposed a lightweight and secure session key establishment scheme for smart home environments. Using a short authentication token, a session key was established between *GWN* and smart device.

Wazid et al. [8] proposed a new secure remote user authentication scheme for a smart home environment. The scheme only utilizes the one way hash function and XOR operation as result it is efficient for resource-constrained smart devices. However, this scheme uses a verification table saved in the *GWN*'s database which if it was stolen by the attacker, then the result is disastrous. In addition, the proposed scheme suffered from synchronization attack as it uses time stamp to resist replay attack.

Shuai et al. [9] proposed a remote authentication scheme for smart home environment using ECC. The scheme dose not require to store the verification table for authentication purposes. However, the scheme suffers from unsatisfactory performance in terms of computational and communication costs.

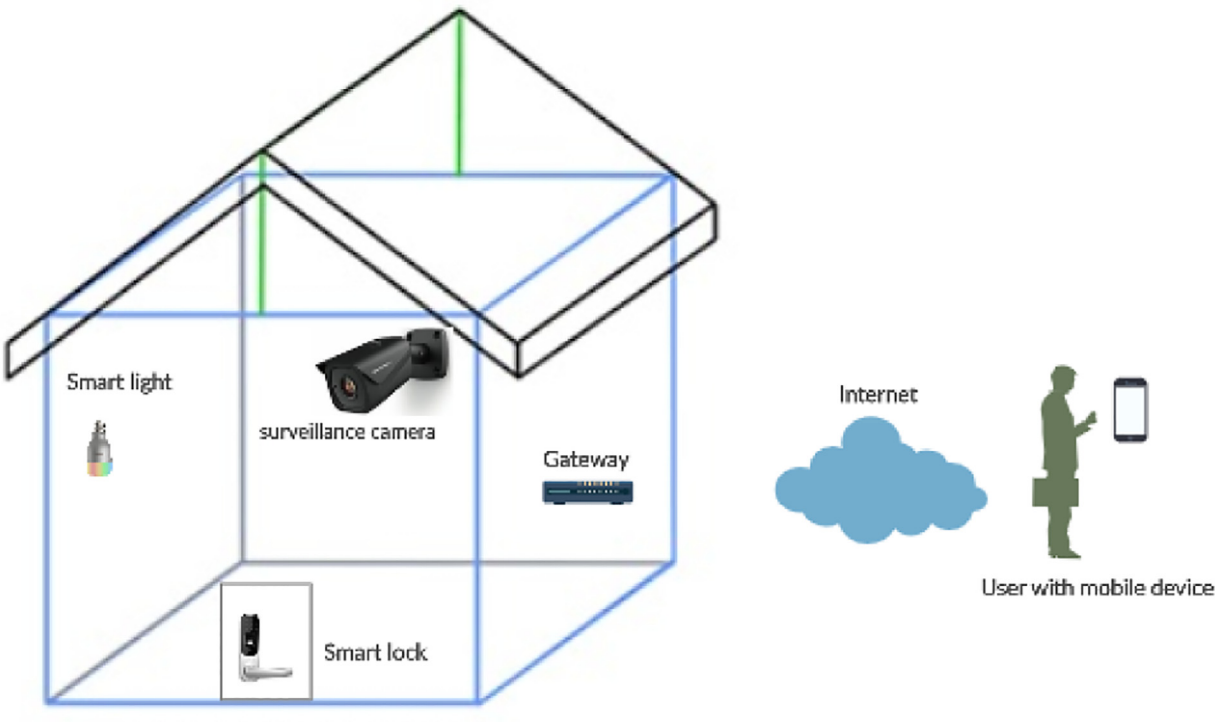

Fig. 1. high-level architecture of smart home environment

3. The proposed scheme

In this section, we discuss our proposed secure remote anonymous user authentication scheme. The proposed scheme uses context-awareness and transaction history to achieve the desirable security features. A typical high-level architecture of smart home environment is illustrated in Fig. 1 adapted from [7–9].

There are four types of participants: User devices, Home devices (smart devices), gateway and registration authority. Every smart device, and *GWN* is securely registered offline at the registration authority (*RA*). At that point the user who needs to access the smart device requires to register at the registration authority by offering his/her necessary information.

Each user has a mobile device (MD_i) capable of reading the credentials supplied by that user, such as identity, password and biometrics (fingerprint scanning, etc.).

The gateway is responsible for managing the communication between the home devices and user devices. The authentication request of the authorized user is sent to the *GWN* and then the *GWN* sends the request to target smart device. The smart device sends corresponding reply to the *GWN* and then the *GWN* forward the response to the user. Our scheme has six phases:

1. Pre-deployment phase
2. Registration phase
3. Login phase
4. Authentication phase
5. Password update phase

For convenience, the notations mentioned in the proposed scheme presented in the Table 1.

3.1. Pre-deployment phase

The pre-deployment phase takes place at the manufacturer's site before the devices are deployed. The mobile devices will be loaded with unique symmetric key K_{ur} shared between the registration authority and each mobile device. The smart devices also will be loaded with unique symmetric key K_{sr} shared between the registration authority and each smart device. Lastly, the gateway will be loaded with unique symmetric key K_{gr} shared between the registration authority and gateway.

Table 1
Notations used in our protocol.

Notation	Descriptions
U_i	Mobile User
ID_{Ui}	Identity of user
PW_i	Password of users
TID_{Ui}	Temporary identity of user
MD_i	Mobile device of i^{th} user
GWN	Gateway node
GID	Unique identity of GWN
SD_j	Smart device in the home
SID_j	Unique identity of SD_j
RA	Registration authority
K_{ur}	Symmetric key shared between the registration authority and each mobile device
L_P, L_C	The previous and current location, respectively
X_n	The history of all user locations
$HMAC$	keyed-hash message authentication code
N_1, N_2, N_3	Current nonces generated by U_i, GWN and SD_j, respectively
SK	Session key
$M_1 \| \| M_2$	Concatenate operation
$(X)\_K$	Message X encrypted with K
$h(\cdot)$	One-way hash function
$\oplus$	XOR operation

3.2. Registration phase

Each smart device in the smart home and the gateway have to be register with the *RA*. Moreover, each user needs to access a smart devices SD_j has to register with the *RA*. The registration phase consists of two parts.

3.2.1. Smart device and gateway registration step:

This step is done offline. Assuming in the smart home environment, there are j^{th} smart devices (SD_j) and one gateway. The *RA* selects unique identity for the gateway *GID* and each smart device SID_j.

3.2.2. User registration step:

Assume that there are i^{th} users U_i. Each user selects the identity ID_{Ui} and a generates password PW_i, the user then needs to supply his/her identity and password to the MD_i, which then encrypts the ID_{Ui} and PW_i using the symmetric key K_{ur} and send them to *RA*.

$$U_i \rightarrow RA : (ID_{Ui}||PW_i)\_K_{RA} \tag{1}$$

Upon receiving the ID_{Ui} and PW_i, *RA* will look them up in its database. If they exist, this indicates that the user is trying to update his/her credentials, the *RA* will then ask the user to re-submit the identity and password again.

The identity and password update phase are explained in Section 3.5. Otherwise, the *RA* generates a temporary identity for the user TID_{Ui} and computes the following parameters:

$$V_1 = h(ID_{Ui}||PW_i) \tag{2}$$

$$V_2 = h(GID||ID_{Ui} \oplus PW_i) \tag{3}$$

$$V_3 = h(GID \oplus ID_{Ui}||PW_i) \tag{4}$$

RA sends back TID_{Ui} and V_1 to the user, TID_{Ui} and V_3 to the smart device and TID, V_2 and V_3 to the gateway as shown in Fig. 2.

When MD_i receives the TID_{Ui} and V_1, it will generate variable *Counter* and set it to 0. Finally, the user, gateway and smart device store the received parameters in their databases.

3.3. Login phase

The user inputs his/her identity ID_{Ui} and password PW_i into the mobile device, which compute V_1^*, and check if $V_1^* \neq V_1$ then the mobile device terminates the login request, increments the counter and check if it reaches predetermined value for instance 3, this mean the mobile device is breached, then the mobile device terminates the login request immediately until the user re-register again. Otherwise, the user is authenticated and can access the application on his/her mobile device.

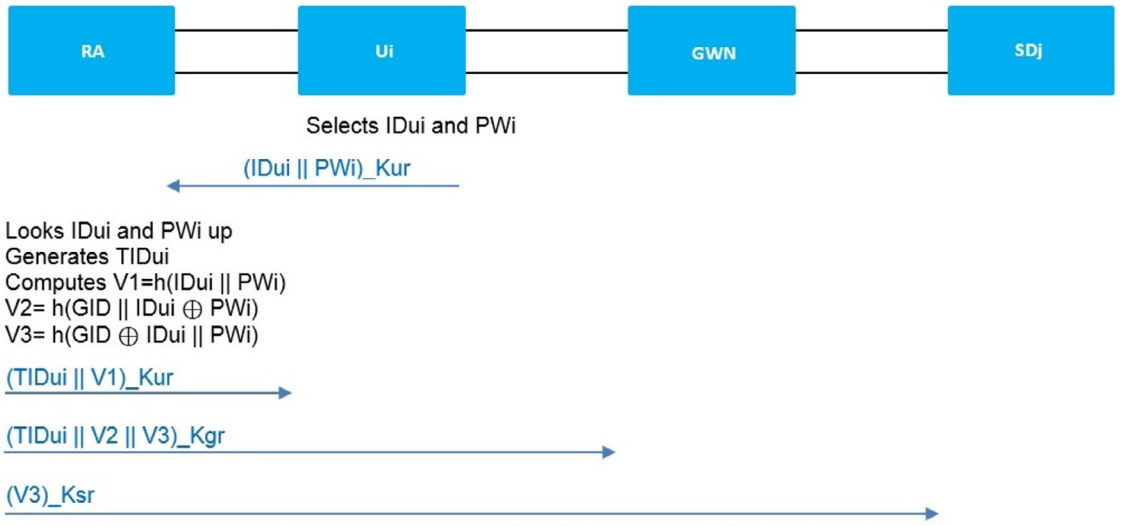

Fig. 2. Registration phase of the proposed scheme

3.4. Authentication phase

Referring to Fig. 3, the mobile device generates a nonce N_1, then computes the dynamic identity.

$$DID_{Ui} = TID_{Ui} \oplus N_1 \tag{5}$$

The DID_{Ui} will be unique in each session. Hence, the anonymity and untraceability proprieties are achieved. Next, the user chooses target smart device, defined by its identity SID_j.

The mobile device extracts its current location L_C and performs the following iterative hashing operation preformed at session n

$$X_n = h(X_{n-1}||L_C), \quad n > 0 \tag{6}$$

$$X_0 = 0 \tag{7}$$

where X_n represents the hash of cumulative locations or the hashed history of all user locations at session n. Next, the mobile device computes V_2 using Eq. (3). Using V_2, the MD_i computes UG.

$$UG = (SID_j||N_1||L_C||X_n) \oplus V_2 \tag{8}$$

Next, the MD_i computes HUG:

$$HUG = h(SID_j||N_1||L_C||X_n) \tag{9}$$

The mobile sends the following message to the gateway through the public channel:

$$U_i \rightarrow GWN : (DID_{Ui}||UG||HUG) \tag{10}$$

Upon receiving the message, the gateway will compute $(SID_j||N_1||L_C||X_n)$ using the stored value V_2:

$$(SID_j||N_1||L_C||X_n) = UG \oplus V_2 \tag{11}$$

GWN now has the values of SID_j, N_1, L_C and X_n. Using those values and from Eq. (9) the GWN computes HUG^* and check if $HUG^* = HUG$, then the integrity is verified. Otherwise, the GWN will terminate the session with the user because the message is modified before it reaches to GWN.

GWN then checks the freshness of received nonce N_1. This will prevent the replay attack.

Using N_1, GWN computes TID_{Ui}:

$$TID_{Ui} = DID_{Ui} \oplus N_1 \tag{12}$$

GWN checks TID_{Ui} and compares it with the stored value in its database. If TID_{Ui} does not match the stored value, GWN terminates the session with the user.

GWN then estimates the maximum radius of motion for the user given its current location L_c and using the linear motion equation to calculate the highest displacement for user in location L_p change to location L_c, where L_p and L_c are the previous and current location of the user, respectively.

$$\Delta L_{max} = V \Delta T \tag{13}$$

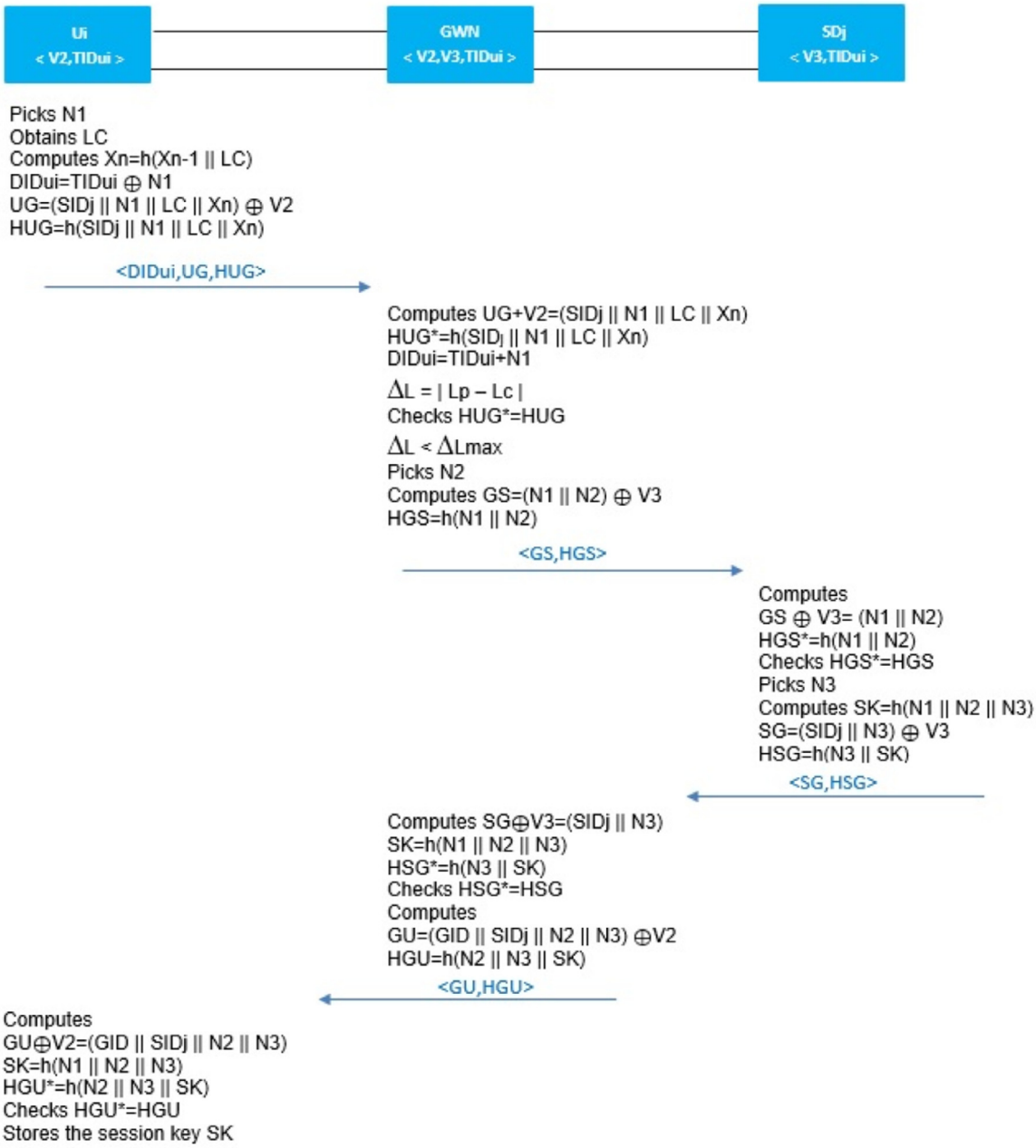

Fig. 3. Authentication phase of the proposed scheme

where ΔT represents the time needed by the user to move from location L_p to location L_c and V represents the maximum velocity that the user could have. For this work, we assume $V = 489.241$ km/h which represents the highest speed for Bugatti Chiron Super Sport recorded in 2019 [10].

Assuming the user at location A accessed a smart device and after 10 min, the user is trying to access a smart device at location B, L_p and L_c represents the the previous location and current location, respectively.

The distance between location A and location B is about 93 km, using the linear motion equation we get:

$$\Delta L_{max} = 447.19km/h \times 10min = 74.233km \tag{14}$$

Fig. 4, we consider the previous location at the center. The maximum radius will be ΔL_{max}, whereas the next authentication attempt is at 93 km from the previous location. This is going to be flagged as malicious access because the user cannot be 93 km away from the previous location in 10 min. However, if current location is less than maximum radius, the authentication attempt will be legitimate. Increasing the value of V reduces the false alarm rate. If the gateway checks

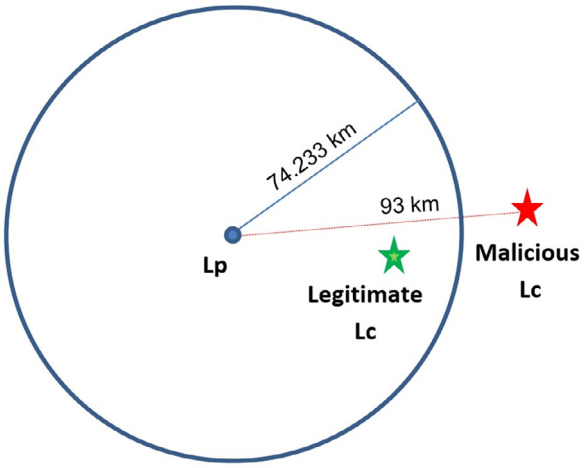

Fig. 4. Authentication based on location

that the location is within the expected range, it still needs to verify the consistency of the cumulative hash history of all previous locations. This can be done by computing X_n^* from Eq. (6) and comparing it to the received X_n. If the two values match, this confirms that the user has consistent locations with *GWN* and the user is authenticated by *GWN*.

Next, the gateway starts to prepare the message that will be sent to the smart device. It will first generate a nonce N_2 and forms the *GS* and *HGS*:

$$GS = (N_1||N_2) \oplus V_3 \tag{15}$$

$$HGS = h(N_1||N_2) \tag{16}$$

Finally, the gateway sends the message to the smart device SD_j:

$$GWN \rightarrow SD_j : (GS||HGS) \tag{17}$$

Once the smart device SD_j receives the message, it computes the following:

$$(N_1||N_2) = GS \oplus V_3 \tag{18}$$

First, SD_j verifies the *GID* and TID_{Ui}. Next, SD_j checks the freshness of N_1 and N_2. Next, to verify the integrity of the message, SD_j computes *HGS* using Eq. (16) and compare it with the received *HGS*, if it does not matches, the smart device terminates the session with the gateway because the smart device might communicates with rogue device as gateway. Otherwise, the smart device generates a nonce N_3 and computes the shared key as follows:

$$SK = h(N_1||N_2||N_3) \tag{19}$$

The smart device starts to prepare the reply to the gateway. First, it will compute *SG* and *HSG*:

$$SG = (SID_j||N_3) \oplus V_3 \tag{20}$$

$$HSG = h(N_3||SK) \tag{21}$$

Lastly, the smart device sends the following message to the gateway:

$$SID_j \rightarrow GWN : (SG||HSG) \tag{22}$$

Upon receiving the message, the gateway will extract N_3:

$$(SID_j||N_3) = SG \oplus V_3 \tag{23}$$

The gateway now has the value of N_3 and can calculate *SK* using Eq. (19). Then it will verify the integrity using Eq. (21). Finally, the gateway forwards N_2 and N_3 to the user:

$$GU = (GID||SID_j||N_2||N_3) \oplus V_2 \tag{24}$$

Next, the *GWN* computes *HGU*:

$$HGU = h(N_2||N_3||SK) \tag{25}$$

The *GWN* sends the following message to the user:

$$GWN \rightarrow U_i : (GU||HGU) \tag{26}$$

Upon receiving the message, the user will extract N_2 and N_3:

$$(N_2||N_3) = GU \oplus V_2 \tag{27}$$

Using N_2 and N_3 the mobile device will compute the session key using Eq. 19. Finally, the mobile device verifies the integrity by calculate *HGU* from Eq. (25) and compare it with the received *HGU*.

If for any reason the user failed to submit a correct location (ex. the user travels by airplane and he/she is a legitimate user), we add a challenge for instance the gateway sends a one of previous nonce (assume N_5) as follows:

$$c = N_5 \oplus V_2 \tag{28}$$

$$HMAC = h(c, N_5) \tag{29}$$

$$GWN \rightarrow U_i : (c, HMAC) \tag{30}$$

Upon receiving the challenge message, the mobile device computes the nonce:

$$N_5 = c \oplus V_2 \tag{31}$$

Next, the mobile device computes the *HMAC* and verify the integrity, if it matches the received *HMAC*. Then mobile device checks its database for the value of X_5 that meet N_5 and then sends back the response as follows:

$$R = X_5 \oplus V_2 \tag{32}$$

$$HMAC = h(R, N_5) \tag{33}$$

Upon receiving the response, the gateway will verify the *HMAC* and the value of X_5 and based on that whether the user is authenticated by the gateway or not.

3.5. Password update phase

The proposed scheme offers a password update facility through which a permissible user U_i can update his/her password at any time after user registration without involving the *RA*. The User needs to provide his/her identity and old password into mobile device. The mobile device calculates V_1 from Eq. (2) and check if $V_1^* = V_1$. If it is not, the mobile device refuses the request to change the password. Otherwise, the mobile device believes that U_i is a legitimate user and enable him/her to change the password. The mobile device asks the user to re-submit his/her identity and new password then mobile will calculate the new V_1 and store it in the mobile device. Next, the mobile device computes the new V_2 and V_3, and send the new V_2 and V_3 to *GWN* and V_3 to SD_j after being encrypted with *SK*.

4. Security analysis of the proposed scheme

In this section, we use three different approaches to validate the security of our proposed protocol: formal validation using BAN logic, and formal model checking and simulation using AVISPA tool, informal security analysis.

4.1. Formal proof based on BAN logic

In this Subsection, we introduce a formal analysis for the proposed scheme using widely accepted model called BAN logic, this model has been used for a formal verification of security protocols which introduced in 1989 by Burrows et al. [11]. We begin our analysis by introducing the most important symbols and notations adapted from [25] which are given in Table 2.

In addition, the following BAN logic basic rules are used to prove that our authentication protocol provides secure mutual authentication and key agreement as follows:

- Message-meaning rule:
  If P believes that the key K is shared with Q and P sees X encrypted under K, then P believes that Q once said X.

$$\frac{P| \equiv Q \overset{K}{\leftrightarrow} P, P \triangleleft \langle X \rangle_K}{P| \equiv Q| \sim X}$$

- Nonce verification rule:
  If P believes X is fresh and P believes Q once said X, then P believes Q believes X.

$$\frac{P| \equiv \#(X), P| \equiv Q| \sim X}{P| \equiv Q| \equiv X}$$

- Jurisdiction rule:
  If P believes Q has jurisdiction over X and P believes Q believes X, then P believes X.

$$\frac{P| \equiv Q| \Longrightarrow X, P| \equiv Q| \equiv X}{P| \equiv X}$$

Table 2
Notations in BAN logic.

Notation	**Descriptions**
P and Q	Principals
$P \mid\equiv$ X	Principal P believes the statement X
$P \triangleleft$ X	Principal P sees the statement X
$P \mid\Rightarrow$ X	Principal P has jurisdiction over the statement X
$P \mid\sim$ X	Principal P once said statement X
(X, Y)	The statement X or Y is one part of message (X, Y)
$<X>_Y$	The statement X is encrypted with the key K
$(X)_K$	The statement X is hashed with the key K
$P \overset{K}{\longleftrightarrow} Q$	K is a secret parameter shared (or to be shared) between P and Q
$P \overset{K}{=\!=} Q$	X is a secret known only to P and Q, and possibly to parties trusted by them.
# (X)	The message X is *fresh*.

- Freshness conjuncatenation rule:
  If one part of a statement is fresh, then the entire statement must also be fresh; so if P believes X is fresh, then P believes X and Y are fresh.

$$\frac{P| \equiv \#(X)}{P| \equiv \#(X, Y)}$$

- Belief rule:
  If P believes X and Y, then P believes X.

$$\frac{P| \equiv (X, Y)}{P| \equiv X}$$

- Session keys rule:

$$\frac{P| \equiv \#(X), P| \equiv Q| \equiv X}{P| \equiv P \overset{K}{\leftrightarrow} Q}$$

The proposed scheme must achieve the following goals:

- Goal 1

$$GWN| \equiv U_i| \equiv U_i \overset{SK}{\leftrightarrow} GWN$$

- Goal 2:

$$GWN| \equiv U_i \overset{SK}{\leftrightarrow} GWN$$

- Goal 3:

$$GWN| \equiv SD_j| \equiv SD_j \overset{SK}{\leftrightarrow} GWN$$

- Goal 4:

$$GWN| \equiv SD_j \overset{SK}{\leftrightarrow} GWN$$

- Goal 5:

$$SD_j| \equiv GWN| \equiv GWN \overset{SK}{\leftrightarrow} SD_j$$

- Goal 6:

$$SD_j| \equiv GWN \overset{SK}{\leftrightarrow} SD_j$$

- Goal 7:

$$U_i| \equiv GWN| \equiv U_i \overset{SK}{\leftrightarrow} GWN$$

- Goal 8:

$$U_i| \equiv U_i \overset{SK}{\leftrightarrow} GWN$$

- Goal 9:

$$U_i| \equiv SD_j| \equiv U_i \overset{SK}{\leftrightarrow} SD_j$$

- Goal 10:

$$SD_j| \equiv U_i| \equiv U_i \overset{SK}{\leftrightarrow} SD_j$$

- Goal 11:

$$U_i| \equiv U_i \overset{SK}{\leftrightarrow} SD_j$$

- Goal 12:

$$SD_j| \equiv U_i \overset{SK}{\leftrightarrow} SD_j$$

The fundamental assumptions of the authentication protocol are as follows:

- A1:

$$GWN| \equiv \#(N_1)$$

- A2:

$$GWN| \equiv \#(N_3)$$

- A3:

$$SD_j| \equiv \#(N_2)$$

- A4:

$$U_i| \equiv U_i \overset{V_2}{\leftrightarrow} GWN$$

- A5:

$$GWN| \equiv U_i \overset{V_2}{\leftrightarrow} GWN$$

- A6:

$$SD_j| \equiv SD_j \overset{V_3}{\leftrightarrow} GWN$$

- A7:

$$GWN| \equiv SD_j \overset{V_3}{\leftrightarrow} GWN$$

- A8:

$$U_i| \equiv SD_j| \Rightarrow (N_3, SID_j, SK)$$

- A9:

$$U_i| \equiv GWN| \Rightarrow (N_2, V_2, SK)$$

- A10:

$$GWN| \equiv U_i| \Rightarrow (N_1, TID_{Ui}, V_2, SK)$$

- A11:

$$GWN| \equiv SD_j| \Rightarrow (N_3, SID_j, V_3, SK)$$

- A12:

$$SD_j| \equiv U_i| \Rightarrow (N_1, TID_{Ui}, V_2, SK)$$

- A13:

$$SD_j| \equiv GWN| \Rightarrow (N_2, V_3, SK)$$

Messages transferred in the authentication protocol:

- Msg 1:

$$U_i \rightarrow GWN : (TID_{Ui}||UG||HUG)_{U_i \overset{V2}{\leftrightarrow} GWN}$$

- Msg 2:

$$GWN \rightarrow SD_j : (GID||GS||HGS)_{GWN \overset{V_3}{\leftrightarrow} SD_j}$$

- Msg 3:

$$SID_j \rightarrow GWN : (SG||HSG)_{SD_j \overset{V_3}{\leftrightarrow} GWN}$$

- Msg 4:

$$GWN \rightarrow U_i : (GU||HGU)_{GWN \overset{V2}{\leftrightarrow} U_i}$$

Analysis of our authentication scheme:

- S1: According to Msg 1, we get:

$$GWN \ \triangleleft \ (TID_{Ui}, UG, HUG)_{U_i \overset{V2}{\leftrightarrow} GWN}$$

- S2: Based on Assumption A5, S1 and message-meaning rule, we have:

$$\frac{GWN| \equiv U_i \overset{V2}{\leftrightarrow} GWN, GWN \triangleleft (TID_{Ui}, UG, HUG)_{U_i \overset{V2}{\leftrightarrow} GWN}}{GWN| \equiv U_i| \sim (TID_{Ui}, UG, HUG)_{U_i \overset{V2}{\leftrightarrow} GWN}}$$

- S3: From A1 and freshness-conjuncatenation rule, we get:

$$GWN \mid \equiv \# \ (TID_{Ui}, UG, HUG)_{U_i \overset{V2}{\leftrightarrow} GWN}$$

- S4: From S3, S2 and nonce-verification rule, we get:

$$\frac{GWN| \equiv \#(TID_{Ui}, UG, HUG)_{U_i \overset{V2}{\leftrightarrow} GWN}, GWN| \equiv U_i| \sim (TID_{Ui}, UG, HUG)_{U_i \overset{V2}{\leftrightarrow} GWN}}{GWN| \equiv U_i| \equiv (TID_{Ui}, UG, HUG)_{U_i \overset{V2}{\leftrightarrow} GWN}}$$

- S5: According to the Msg 2, we get:

$$SD_j \ \triangleleft \ (GID, GS, HGS)_{GWN \overset{V3}{\leftrightarrow} SD_j}$$

- S6: From A6, S5 and message-meaning rule, we have:

$$\frac{SD_j| \equiv (SD_j \overset{V_3}{\leftrightarrow} GWN), SD_j \triangleleft (GID, GS, HGS)_{GWN \overset{V3}{\leftrightarrow} SD_j}}{SD_j| \equiv GWN| \sim (GID, GS, HGS)_{GWN \overset{V3}{\leftrightarrow} SD_j}}$$

- S7: From A3 and freshness-conjuncatenation rule, we get:

$$SD_j \mid \equiv \# \ (GID, GS, HGS)_{GWN \overset{V3}{\leftrightarrow} SD_j}$$

- S8: From S6, S7 and nonce-verification rule, we get:

$$\frac{SD_j \mid \equiv \# \ (GID, GS, HGS)_{GWN \overset{V3}{\leftrightarrow} SD_j}, SD_j| \equiv GWN| \sim (GID, GS, HGS)_{GWN \overset{V3}{\leftrightarrow} SD_j}}{SD_j| \equiv GWN| \equiv (GID, GS, HGS)_{GWN \overset{V3}{\leftrightarrow} SD_j}}$$

- S9: According to the Msg3, we get:

$$GWN \triangleleft (SG||HSG)_{SD_j \overset{V3}{\leftrightarrow} GWN}$$

- S10: From A7, S9 and message-meaning rule, we have:

$$\frac{GWN| \equiv (SD_j \overset{V_3}{\leftrightarrow} GWN), GWN \triangleleft (SG||HSG)_{SD_j \overset{V3}{\leftrightarrow} GWN}}{GWN| \equiv SD_j| \sim (SG||HSG)_{SD_j \overset{V3}{\leftrightarrow} GWN}}$$

- S11: From A2 and freshness-conjuncatenation rule, we get:

$$GWN \mid \equiv \# \ (SG, HSG)_{SD_j \overset{V3}{\leftrightarrow} GWN}$$

- S12: From S10, S11 and nonce-verification rule, we get:

$$\frac{GWN \mid \equiv \# \ (SG, HSG)_{SD_j \overset{V3}{\leftrightarrow} GWN}, GWN| \equiv SD_j| \sim (SG, HSG)_{SD_j \overset{V3}{\leftrightarrow} GWN}}{GWN| \equiv SD_j| \equiv (SG, HSG)_{SD_j \overset{V3}{\leftrightarrow} GWN}}$$

- S13: According to the Msg4, we get:

$$U_i \triangleleft (GU||HGU)_{GWN \overset{V2}{\leftrightarrow} U_i}$$

- S14: From A4, S13 and message-meaning rule, we have:

$$\frac{U_i| \equiv U_i \overset{V_2}{\leftrightarrow} GWN, U_i \triangleleft (GU||HGU)_{GWN \overset{V2}{\leftrightarrow} U_i}}{U_i| \equiv GWN| \sim (GU||HGU)_{GWN \overset{V2}{\leftrightarrow} U_i}}$$

- S15: From A1, A2, A3 and freshness-conjuncatenation rule, we get:

$$U_i \mid \equiv \#\ (GU, HGU)_{GWN \overset{V_2}{\leftrightarrow} U_i}$$

- S16: From S14, S15 and nonce-verification rule, we get:

$$\frac{U_i \mid \equiv \#\ (GU, HGU)_{GWN \overset{V_2}{\leftrightarrow} U_i}, U_i| \equiv GWN| \sim (GU, HGU)_{GWN \overset{V_3}{\leftrightarrow} U_i}}{U_i| \equiv GWN| \equiv (GU, HGU)_{GWN \overset{V_2}{\leftrightarrow} U_i}}$$

- S17: From A10, S4 and jurisdiction rule, we get:

$$\frac{GWN| \equiv U_i| \Rightarrow (N_1, TID_{Ui}, V_2, SK), GWN| \equiv U_i| \equiv (TID_{Ui}, UG, HUG)_{U_i \overset{V_2}{\leftrightarrow} GWN}}{GWN| \equiv (TID_{Ui}, UG, HUG)_{U_i \overset{V_2}{\leftrightarrow} GWN}}$$

- S18: From S3, S4 and session keys rule, we get:

$$\frac{GWN| \equiv \#(TID_{Ui}, UG, HUG)_{U_i \overset{V_2}{\leftrightarrow} GWN}, GWN| \equiv U_i| \equiv (TID_{Ui}, UG, HUG)_{U_i \overset{V_2}{\leftrightarrow} GWN}}{GWN| \equiv U_i| \equiv U_i \overset{SK}{\leftrightarrow} GWN}$$

(Goal 1)

- S19: From S18, A10 and jurisdiction rule, we get:

$$\frac{GWN| \equiv U_i| \Rightarrow (N_1, TID_{Ui}, V_2, SK), GWN| \equiv U_i| \equiv U_i \overset{SK}{\leftrightarrow} GWN}{GWN| \equiv U_i \overset{SK}{\leftrightarrow} GWN}$$

(Goal 2)

- S20: From A11, S12 and jurisdiction rule, we get:

$$\frac{GWN| \equiv SD_j| \Rightarrow (N_3, SID_j, V_3, SK), GWN| \equiv SD_j| \equiv (SG, HSG)_{SD_j \overset{V_3}{\leftrightarrow} GWN}}{GWN| \equiv (SG, HSG)_{SD_j \overset{V_3}{\leftrightarrow} GWN}}$$

- S21: From S11, S12 and session keys rule, we get:

$$\frac{GWN \mid \equiv \#\ (SG, HSG)_{SD_j \overset{V_3}{\leftrightarrow} GWN}, GWN| \equiv SD_j| \equiv (SG, HSG)_{SD_j \overset{V_3}{\leftrightarrow} GWN}}{GWN| \equiv SD_j| \equiv SD_j \overset{SK}{\leftrightarrow} GWN}$$

(Goal 3)

- S22: From A11, S21 and jurisdiction rule, we get:

$$\frac{GWN| \equiv SD_j| \Rightarrow (N_3, SID_j, V_3, GWN| \equiv SD_j| \equiv SD_j \overset{SK}{\leftrightarrow} GWN}{GWN| \equiv SD_j \overset{SK}{\leftrightarrow} GWN}$$

(Goal 4)

- S23: From A13, S8 and jurisdiction rule, we get:

$$\frac{SD_j| \equiv GWN| \Rightarrow (N_2, V_3, SK), SD_j| \equiv GWN| \equiv (GID, GS, HGS)_{GWN \overset{V_3}{\leftrightarrow} SD_j}}{SD_j| \equiv (GID, GS, HGS)_{GWN \overset{V_3}{\leftrightarrow} SD_j}}$$

- S24: From S7, S8 and session keys rule, we get:

$$\frac{SD_j \mid \equiv \#\ (GID, GS, HGS)_{GWN \overset{V_3}{\leftrightarrow} SD_j}, SD_j| \equiv GWN| \equiv (GID, GS, HGS)_{GWN \overset{V_3}{\leftrightarrow} SD_j}}{SD_j| \equiv GWN| \equiv SD_j \overset{SK}{\leftrightarrow} GWN}$$

(Goal 5)

- S25: From A13, S24 and jurisdiction rule, we get:

$$\frac{SD_j| \equiv GWN| \Rightarrow (N_2, V_3, SK), SD_j| \equiv GWN| \equiv GWN \overset{SK}{\leftrightarrow} SD_j}{SD_j| \equiv GWN \overset{SK}{\leftrightarrow} SD_j}$$

(Goal 6)

- S26: From A9, S16 and jurisdiction rule, we get:

$$\frac{U_i| \equiv GWN| \Rightarrow (N_2, V_2, SK), U_i| \equiv GWN| \equiv (GU, HGU)_{GWN \overset{V_2}{\leftrightarrow} U_i}}{U_i| \equiv (GU, HGU)_{GWN \overset{V_2}{\leftrightarrow} U_i}}$$

- S27: From S15, S16 and session keys rule, we get:

$$\frac{U_i \mid \equiv \#\ (GU, HGU)_{GWN \overset{V_2}{\leftrightarrow} U_i}, U_i \mid \equiv GWN \mid \equiv (GU, HGU)_{GWN \overset{V_2}{\leftrightarrow} U_i}}{U_i \mid \equiv GWN \mid \equiv U_i \overset{SK}{\leftrightarrow} GWN}$$

(Goal 7)

- S28: From A9, S27 and jurisdiction rule, we get:

$$\frac{U_i \mid \equiv GWN \mid \Rightarrow (N_2, V_2, SK), U_i \mid \equiv GWN \mid \equiv U_i \overset{SK}{\leftrightarrow} GWN}{U_i \mid \equiv U_i \overset{SK}{\leftrightarrow} GWN}$$

(Goal 8)

- S29: From S27 and S21, we get:

$$\frac{U_i \mid \equiv GWN \mid \equiv U_i \overset{SK}{\leftrightarrow} GWN, GWN \mid \equiv SD_j \mid \equiv SD_j \overset{SK}{\leftrightarrow} GWN}{U_i \mid \equiv SD_j \mid \equiv U_i \overset{SK}{\leftrightarrow} SD_j}$$

(Goal 9)

- S30: From S24 and S18, we get:

$$\frac{SD_j \mid \equiv GWN \mid \equiv SD_j \overset{SK}{\leftrightarrow} GWN, GWN \mid \equiv U_i \mid \equiv U_i \overset{SK}{\leftrightarrow} GWN}{SD_j \mid \equiv U_i \mid \equiv SD_j \overset{SK}{\leftrightarrow} U_i}$$

(Goal 10)

- S31: From A8, S29 and jurisdiction rule, we get:

$$\frac{U_i \mid \equiv SD_j \mid \Rightarrow (N_3, SID_j, SK), U_i \mid \equiv SD_j \mid \equiv U_i \overset{SK}{\leftrightarrow} SD_j}{U_i \mid \equiv U_i \overset{SK}{\leftrightarrow} SD_j}$$

(Goal 11)

- S32: From A12, S30 and jurisdiction rule, we get:

$$\frac{SD_j \mid \equiv U_i \mid \Rightarrow (N_1, TID_{Ui}, V_2, SK), SD_j \mid \equiv U_i \mid \equiv SD_j \overset{SK}{\leftrightarrow} U_i}{SD_j \mid \equiv U_i \overset{SK}{\leftrightarrow} SD_j}$$

(Goal 12)

Hence, the above BAN logic analysis formally proves that the proposed scheme successfully achieves mutual authentication, and the session key SK is mutually established between the U_i and the SD_j through the GWN.

4.2. Simulation based on AVISPA tool

AVISPA, introduced by Armando et al. [12] is a toolkit based on the Dolev – Yao threat model [13]. In this model, the adversary has the ability to change, forward and modify messages. This toolkit is utilized to formally assess and validate Internet security protocols. AVISPA is a widely recognized tool used to evaluate the specifications of several industrial-scale security protocols [14]. Avispa uses a high level protocol specification language (HLPSL) to describe and define the security protocols. Protocol specifications in HLPSL are break down into roles. Some roles are used to define the actions of one single agent in a protocol run. Every agent plays a unique role during the execution of a given protocol. The main objective of HLPSL is to check security properties such as the message authentication, agent authentication and secrecy. The security protocol is checked indicating whether or not it is secure on the basis of the predefined goals. AVISPA includes four built-in model checkers, defined as follows:

4.2.1. Preliminaries

1. On-the-fly model checker (OFMC): uses lazy data types as an easy way to create an effective on-the-fly model for security protocols with infinite state spaces [15].
2. Constraint-logic-based attack searcher (CL-AtSe): The input of (CL-AtSe) is protocol defined as a set of rewriting rules (IF format) into In a set of constraints that help to detect the attacks on the security protocol [16].
3. SAT-based model checker (SATMC): produces a propositional formula based on a transitional state obtained from the IF specification. The propositional formula defines any breach of the security properties that can be turned into an attack [17].

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role role_Ui(Ui,RA,GWN,SD:agent,Kur:symmetric_key,H:hash_func,SND,RCV:channel(dy))
played_by Ui
def=
        local State:nat,GID,IDUi,PWi,IDsd,N1,N2,N3,Lc,X1,TID:text,UG,GU,M1,M4:message,HUG,HGU:hash(message),V1:hash
(text.text),V2:hash(text.message),X2:hash(text.message),SK:hash(text.text.text)
        init State := 0
        transition
                0. State=0 /\ RCV(start) =|> State':=2 /\ SND({IDUi.PWi}_Kur) /\ secret(IDUi,idui,{RA,Ui,GWN,SD}) /\ secret
(PWi,pwi,{RA,Ui,GWN,SD})
                2. State=2 /\ RCV({V1'.TID'}_Kur) =|> State':=4 /\ N1':=new() /\ Lc':=new() /\ X1':=new() /\ X2':=H(Lc.X1) /\
V2':=H(GID.xor(IDUi.PWi)) /\ M1':=(IDsd.N1.Lc.X2) /\ UG':=xor(M1,V2) /\ HUG':=H(M1) /\ SND(H(TID).UG.HUG) /\ secret(N1,sec_N1,
{Ui,GWN,SD}) /\ witness(Ui,GWN,ui_gwn_N1,N1) /\ secret(Lc,sec_Lc,{Ui,GWN}) /\ secret(X2,sec_X2,{Ui,GWN})
                4. State=4 /\ RCV(GU'.HGU') =|> State':=6 /\ M4':=xor(GU,V2) /\ SK':=H(N1.N2.N3) /\ HGU':=H(N2.N3.SK)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fig. 5. The role played by the user U_i

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role role_RA(RA,Ui,GWN,SD:agent,Kur,Kgr,Ksr:symmetric_key,H:hash_func,SND,RCV:channel(dy))
played_by RA
def=
        local State:nat,IDUi,GID,PWi,TID:text,V1:hash(text.text),V2:hash(text.message),V3:hash(message)
        init State := 1
        transition
                1. State=1 /\ RCV({IDUi'.PWi'}_Kur)  =|>  State':=3 /\ TID':=new() /\ V1':=H(IDUi.PWi) /\ V2':=H(GID.xor
(IDUi.PWi)) /\ V3':=H(xor(GID.(IDUi.PWi)))
                /\ SND({V1.TID'}_Kur) /\ secret(V1,sec_v1,{RA,Ui,GWN,SD}) /\ SND({V2.V3}_Kgr) /\ secret(V2,sec_v2,{RA,Ui}) /\
SND({V3}_Ksr) /\ secret(V3,sec_v3,{RA,Ui,GWN,SD})
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fig. 6. The role played by the registration authority RA

4. Tree automata-based on automatic approximations for the analysis of security protocols (TA4SP) model checker:It shows the vulnerability of the protocol and predicts the correctness of the protocol by accurately estimating the capabilities of the attacker.

4.2.2. Simulation details

We start writing the HLPSL script for our scheme by setting the simulation security goals. Our main objective is to ensure the secrecy of a number of values such as V_1, V_2, V_3, N_1, N_2, N_3. In addition, we define the six roles as follows: role(1) is role *RA* which is played by the registration authority, role(2) is role U_i which is played by the user, role (3) is role *GWN* which is played by the gateway, role(4) is role SD which is played by the smart device, role(5) is role session which combine the basic roles (*RA*, U_i, *GWN* and *SD*), role(6) is role environment which combines several sessions and contains global variables, functions and define the simulation security goals of the protocol.

Fig 5 shows the specification for role U_i which is played by the user. In this role, the U_i recognizes all the agents (U_i, *RA*, *GWN*, *SD*), the symmetric key K_{ur} which is shared between the user and the *RA*, the hash function H(.) and send/receive channels (SND, RSV). The (dy) notation shows that the channels follow the Dolev–Yao model. U_i receives a start message (RCV(start)) as a signal to start the run of the protocol at the first state (state 0), the user generate the identity ID_{Ui} and password PW_i, then user sends ID_{Ui} and PW_i after being encrypted with K_{ur} to the registration authority in order to register. At state 2, the U_i receives TID_{Ui} and V_1 encrypted with K_{ur} coming from the *RA*. At state 4, the U_i generates a fresh value as a nonce N_1. Next, the U_i generates a fresh value L_C which represent the location of U_i as if the U_i obtain it from the GPS. Next, the U_i generates the X_1 which has all the location's history of the U_i, this value is shared with *GWN*. Next, U_i computes X_2, V_2, *UG* and *HUG*. Next, U_i sends (TID_{Ui}, *UG*, *HUG*) to *GWN*. at the next transition, the U_i receives the message (GU, HGU) coming from the *GWN*. Next, the U_i computes *SK* using N_1, N_2 and N_3. Finally, U_i computes *HUG* and compare it with the received *HUG*.

Fig. 6 illustrates the specification for role *RA* which is played by the registration authority. In this role, the *RA* recognizes all the agents (U_i, *RA*, *GWN*, *SD*), the symmetric keys K_{ur}, K_{gr} and K_{sr} which are shared between the registration authority and User, gateway and smart device, respectively. In addition, *RA* knows the hash function H(.) and send/receive channels (SND, RSV). The (dy) notation shows that the channels follow the Dolev–Yao model. The reset part of the specification describes the different states of the protocol execution by *RA*.

Fig. 7 shows the specification for role *GWN* which is played by the gateway. In this role, the *GWN* knows all the agents (U_i, *RA*, *GWN*, *SD*), the symmetric key K_{gr} which is shared between the gateway and the *RA*, the hash function H(.) and send/receive channels (SND, RSV). The (dy) notation shows that the channels follow the Dolev–Yao model.

Fig. 9 shows the specification of the session and environment role. In the session role, all roles (role *RA*, role U_i, role *GWN*, role *SD*) combine together. In the environment role, one or more sessions are initiated. We defined the constants

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role role_GWN(Ui,RA,GWN,SD:agent,Kgr:symmetric_key,H:hash_func,SND,RCV:channel(dy))
played_by GWN
def=
        local State:nat,TID,IDsd,Lc,GID,N1,N2,N3,X1:text,V2:hash(text.message),V3:hash
(message),UG,GS,SG,GU,M1,M2,M3,M4:message,HUG,HGS,HSG,HGU:hash(message),X2:hash(text.message),SK:hash(text.text.text)
        init State := 30
        transition
                30. State=30 /\ RCV({V2'.V3'}_Kgr) =|> State':=32
                32. State=32 /\ RCV(H(TID').UG'.HUG') =|> State':=34  /\ M1':=xor(UG,V2) /\ HUG':=H(M1) /\ X2':=H(Lc.X1) /\
N2':=new() /\ M2':=(TID.GID.N1.N2) /\ GS':=xor(M2,V3) /\ HGS':=H(M2) /\ SND(GS.HGS) /\ secret(N2,sec_N2,{Ui,GWN,SD}) /\ request
(GWN,Ui,ui_gwn_N1,N1) /\ witness(GWN,SD,gwn_sd_N2,N2)
                34. State=34 /\ RCV(SG'.HSG') =|> State':=36 /\ M3':=xor(SG,V3) /\ SK':=H(N1.N2.N3) /\ HSG':=H(N3.SK) /\ request
(GWN,SD,sd_gwn_N3,N3)/\ M4':=(GID.TID.N2.N3) /\ GU':=xor(M4,V2) /\ HGU':=(N2.N3.SK) /\ SND(GU.HGU)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fig. 7. The role played by the gateway GWN

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role role_SD(Ui,RA,GWN,SD:agent,Ksr:symmetric_key,H:hash_func,SND,RCV:channel(dy))
played_by SD
def=
        local State:nat,TID,GID,IDsd,N1,N2,N3:text,V3:hash(message),GS,SG,M2,M3:message,HGS,HSG:hash(message),SK:hash
(text.text.text)
        init State := 60
        transition
                60. State=60 /\ RCV({V3'}_Ksr) =|> State':=62
                62. State=62 /\ RCV(GS'.HGS') =|> State':=64 /\ M2':=xor(GS,V3) /\ HGS':=H(M2) /\ N3':=new() /\ SK':=H
(N1.N2.N3) /\ M3':=TID.GID.IDsd.N3 /\ SG':=xor(M3,V3) /\ HSG':=H(N3.SK) /\ SND(SG.HSG) /\ secret(N3,sec_N3,{Ui,GWN,SD}) /\
request(SD,GWN,gwn_sd_N2,N2) /\ witness(SD,GWN,sd_gwn_N3,N3)

end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fig. 8. The role played by the smart device SD

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role session(RA,Ui,GWN,SD:agent,IDUi,PWi:text,Kur,Kgr,Ksr:symmetric_key,H:hash_func)
def=
        local SND4,RCV4,SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)
        composition
                role_RA(RA,Ui,GWN,SD,Kur,Kgr,Ksr,H,SND1,RCV1) /\ role_Ui(Ui,RA,GWN,SD,Kur,H,SND2,RCV2) /\ role_GWN
(RA,Ui,GWN,SD,Kgr,H,SND3,RCV3) /\ role_SD(RA,Ui,GWN,SD,Ksr,H,SND4,RCV4)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role environment()
def=
        const ra,ui,gwn,sd:agent,kur,kgr,ksr:symmetric_key,h:hash_func,iDUi,pWi:text,
idui,pwi,sec_v1,sec_v2,sec_v3,sec_N1,sec_N2,sec_N3,sec_Lc,sec_X2,ui_gwn_N1,gwn_sd_N2,sd_gwn_N3:protocol_id

        intruder_knowledge = {ra,ui,gwn,sd}
        composition
                session(ra,ui,gwn,sd,iDUi,pWi,kur,kgr,ksr,h)
end role
goal
        secrecy_of idui,pwi,sec_v1,sec_v2,sec_v3,sec_N1,sec_N2,sec_N3,sec_Lc,sec_X2
        authentication_on ui_gwn_N1,gwn_sd_N2,sd_gwn_N3
end goal

environment()
```

Fig. 9. Role session and environment

as (ra,ui,gwn,sd) represents the agents (RA, U_i, GWN, SD), respectively. (kur,kgr,ksr) represents the symmetric keys shared between RA and user, RA and gateway, RA and smart device, respectively. h represents the hash function H. In the intruder knowledge part, the relevant parameters that the intruder suppose to knows are defined. We assume that the intruder knows all the agents (RA, U_i, GWN, SD). The simulation goals are defined under goal keyword. We interested to check the secrecy of the following parameters: ID_{Ui}, PW_i, V_1, V_2, V_3, N_1, N_2, N_3.

4.2.3. Simulation results

In this section, we present the simulation results of our proposed scheme. The results is based on AVISPA back-end model checker OFMC. The Security Protocol Animator (SPAN) is used to interactively create a message sequence chart (MSC) of the

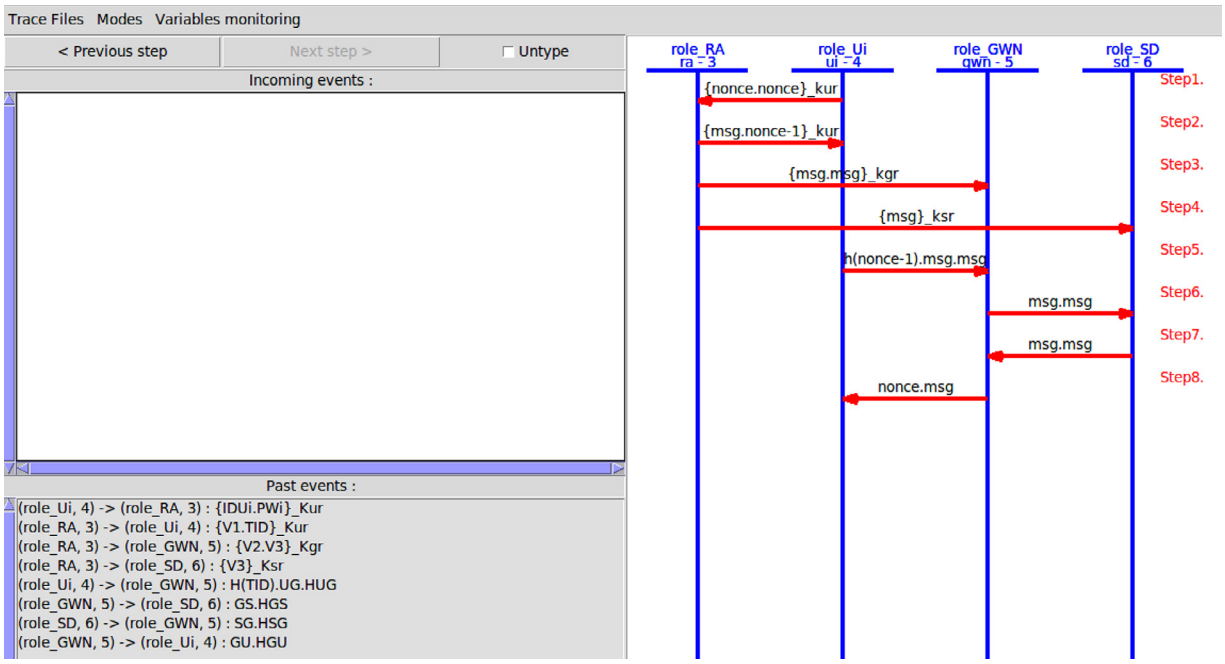

Fig. 10. Result based on model checker OFMC

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/proto10.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.23s
  visitedNodes: 112 nodes
  depth: 9 plies
```

Fig. 11. Result based on model checker OFMC

HLPSL specification protocol previously described. Furthermore, SPAN automatically generates attacks using the Dolev-Yao intruder model.

Fig. 10 shows the protocol execution using SPAN software [12], where the all agents exchange the messages in registration and authentication phase.

Fig. 11 illustrates the report of model checker OFMC which clearly states that the proposed scheme is safe. hence, all the security goals are meet.

4.3. Informal security analysis

In this section, the security of the protocol is discussed against various well-known attacks. We explain how our protocol successfully resists these attacks.

4.4. Replay attack

The random number method is adopted to resist replay attack, so replay attack can be prevented by using the nonces which change in every session.

4.5. Eavesdropping attack

In our protocol the attacker can easily intercept the message in transit between U_i, *GWN*, and *SD* as the messages are all sent in plain-text. However, adversary cannot obtain any confidential information from any messages because the secret information are protected using secret parameters shared between the communicating parties securely (Eg, V_1 and V_2), and shielded using one-way hash function and XOR bitwise operator. Therefore, the attacker will be unable to unfold the transmitted parameters, and thus cannot obtain any useful information.

4.6. Smart-phone device loss attack

In the proposed scheme, the smart-phone of the user stores the secret parameters $V_1 = h(ID_{Ui}||PW_i)$. Suppose that an adversary A steals the smart-phone and extracts the stored secret value V_1, A cannot obtain the identity and password because V_1 is a hash value derived by application of a one-way hashing function. Hence, the proposed protocol is secure even if the smart-phone is lost or stolen.

4.7. Impersonation attack

Suppose if an adversary A tries to impersonate the user, A cannot succeed because does not know the user's identity ID_{Ui} and password PW_i as discussed in resisting smart-phone device loss attack. Thus, the proposed protocol is secure from user impersonation attack.

4.8. Man-in-the-middle attack

As discussed in BAN logic section 4.1, our proposed protocol provides mutual authentication. Additionally, the transmitted messages are protected by the secret values V_1, V_2, V_3 and nonces, and no one would be able to forge legal authentication messages without knowledge of these secret values. Therefore, the proposed protocol can stop the Man-in-the-Middle attack.

4.9. Forward/backward secrecy

The session key *SK* is constructed using three different random numbers, namely N_1, N_2 and N_3 which are randomly generated in each session by U_i, *GWN* and SD_j, respectively. Therefore, if the *SK* is compromised by an adversary A, it cannot compromise the confidentiality information of past or future communication sessions. Therefore, forward/backward secrecy is achieved in the proposed scheme.

4.10. User credentials attack

When U_i registers at *RA* as a legitimate user in the proposed protocol, U_i sends the registration message $(ID_{Ui}||PW_i)\_K_{RA}$ to *RA*. Next, *RA* computes V_1 from Eq. (2) and sends V_1 to U_i. U_i will never store its identity and password credentials, and instead it will store the hash value V_1. An insider attacker targeting user credentials cannot obtain user identity ID_{Ui} and password PW_i from V_1 as V_1 is protected by the one-way hash function. Thus, the proposed protocol can resist the user credentials attack.

4.11. Session key Guessing Attack

The session key *SK* is created by all communication participants, namely U_i, *GWN* and SD_j randomly chosen nonces. Therefore, *SK* depends on randomness of the Input values N_1, N_2 and N_3 and the one-way hash functions, which make it nearly impossible for an adversary to extract it from the protocol. The probability of an adversary to guess the correct *SK* key is so negligible, given that N_1, N_2 and N_3 are randomly chosen in every session.

4.12. User anonymity and untraceability

User anonymity and untracebility are two crucial security properties in authentication. Anonymity ensures the real identity of the mobile device is kept secure and the mobile device remain unidentifiable among the other set of devices. Thus, the attacker cannot identify the identities of the devices. Untraceability, on the other hand, ensures the different sessions established by a particular mobile device cannot be traced, so that an attacker cannot relate any sessions to the correct mobile device. We achieved these two key security properties by using the dynamic identity of the mobile user, where we use different ID in every session.

Table 3
The communication overheads of our scheme.

Transmitted messages	Communication cost in bits
$U_i \rightarrow GWN$	800
$GWN \rightarrow SD$	416
$SD \rightarrow GWN$	416
$GWN \rightarrow U_i$	672

4.13. Location-based authentication

The physical context awareness (location) that is used in our protocol involves verifying whether the previous location of mobile device is proximate to the current location. The location of the mobile device is checked using the linear motion equation to calculate the highest displacement for user in location L_P change to location L_C as explained in Section 3.

4.14. User authentication based on transaction history information

Each mobile device and the gateway maintain a synchronized database of cumulative hashes generated from the previous session based on the location as discussed in Section 3. Therefore, when the transaction from the mobile device is not approved (as discussed in Subsection 3.4, the gateway will challenge the knowledge of the mobile device about the previous locations stored from the previous session. The gateway will select one of previous nonce to represent one of the previous session, and challenge the mobile device to send back the correct corresponding location X_n. The gateway will approve the mobile device if it succeeds in sending the correct X_n; otherwise the mobile device's transaction is rejecting and flagged as malicious.

5. Performance comparison

In this Section, we evaluate the performance of our proposed scheme in terms of storage cost, communication overhead, and computation costs. We also compare our performance with other related works.

5.1. Storage cost

We analyze storage cost (in bits) for the three participants user U_i, gateway GWN, and smart device SD_j.

U_i is required to store GID, SID, TID_{Ui}, X_n and V_1. We use SHA-1 as an example of hash function, and the output of SHA-1 is 160 bits. By applying SHA-1, we obtain $X_n = 160$ bits [18]. While $TID_{Ui} = GID = SID = 128$ bits. Thus, the total storage required by U_i is $160 + (3 \times 128) + 160 = 704$ bits.

GWN is required to store GID, SID, TID_{Ui}, X_n, V_2, and V_3. By applying these settings, we obtain $X_n = 160$ bits. The $TID_{Ui} = GID = SID = 128$ bits, and $V_2 = 512$ and $V_3 = 256$ bits. Therefore, the total storage required by GWN is $160 + (3 \times 128) + (512) + (256) = 1056$ bits.

SD is required to store SID and V_3. $SID = 128$ bits, $V_3 = 256$ bits. Hence, the total storage required by U_i is $128 + 256 = 384$ bits.

5.2. Communication overheads

In this Subsection, we discuss the communication cost based on transmitted messages in both directions between these three participants. The communication costs of our scheme are shown in Table 3.

To make a reasonable comparison, we assume that the length of the transmitted messages' parameters TID_{Ui}, UG, HUG, GS, HGS, SG, HSG, GU, HGU are 128 bits, 512 bits, 160 bits, 256 bits, 160 bits, 256 bits, 160 bits, 512 bits, 160 bits, respectively.

In our proposed scheme, the transmitted messages $U_i \rightarrow GWN : (TID_{Ui}, UG, HUG)$, $GWN \rightarrow SD : (GS, HGS)$, $SD \rightarrow GWN : (SG, HSG)$ and $GWN \rightarrow U_i : (GU, HGU)$ require $(128 + 512 + 160) = 800$ bits, $(256 + 160) = 416$ bits, $(256 + 160) = 416$ bits, $(512 + 160) = 672$ bits, respectively.

Three existing relevant schemes, namely Wazid et al. [8], Shuai et al. [9]. Kumar et al. [7]. In terms of number of exchanged messages and total number of bits for a successful mutual authentication during authentication and key agreement phase. Considering our proposed scheme, the total communication cost turns out to be 4 messages in terms of number of exchanged messages, and to be (800+416+416+672) = 2304 bits in terms of total number of bits.

Table 4 shows a comparison of communication cost between the proposed scheme and other relevant schemes in terms of number of exchanged messages and total number of bits for a successful mutual authentication. our scheme requires 4 messages and 2304 bits total number of bits for a successful mutual authentication. The comparison, in general, shows that our scheme is comparatively more cost-efficient than the other related works in terms of number of exchanged messages and total number of bits, and just a little less cost efficient than that of Kumar et al.'s scheme [7] because our scheme adds

Table 4
Comparison of communication cost between the proposed scheme and other most related schemes.

Authentication scheme	Number of exchanged messages	Total number of bits
Wazid et al. [8]	4	3232
Shuai et al. [9]	4	2944
Kumar et al. [7]	3	1696
Santoso et al. [6]	3	4416
Kim et al. [5]	2	4352
Proposed scheme	4	2304

Table 5
Crypto-operations and the computational times needed

Crypto-operations	Computational time
Modular exponentiation operation (T_{exp})	19.2 ms
Hash function (T_h)	0.32 ms
Symmetric encryption or decryption (T_E/T_D)	5.6 ms

Table 6
Comparison of computation cost between the proposed scheme and other most related schemes in ms.

Authentication scheme	Total cost	Rough estimation
Wazid et al. [8]	$4T_E/T_D + T_{fe} + 22T_h$	46.54 ms
Shuai et al. [9]	$3T_{exp} + 16T_h$	162.72 ms
Kumar et al. [7]	$2T_E/T_D + T_{mac} + T_{hmac} + 2T_h$	12.48 ms
Santoso et al. [6]	$3T_{exp} + 2T_h$	58.24 ms
Kim et al. [5]	$3T_E/T_D + 30T_h$	26.40 ms
Proposed scheme	$10T_h$	3.2 ms

Table 7
Security and functionality features comparison.

Functionality features	**Wazid** et al.	**Shuai** et al.	**Kumar** et al.	**Santoso** et al.	**Kim** et al.	Proposed scheme
Mutual authentication	Yes	Yes	No	No	No	Yes
Session key agreement	Yes	Yes	Yes	Yes	Yes	Yes
User anonymity	Yes	Yes	No	No	No	Yes
Untraceability	Yes	Yes	No	No	No	Yes
Forward security	No	Yes	Yes	No	No	Yes
Avoid clock synchronization problem	No	Yes	No	No	No	Yes
No verification table	No	Yes	Yes	No	No	Yes
Password guessing attack	Yes	Yes	No	No	No	Yes
Mobile device loss attack	Yes	Yes	No	No	No	Yes
Privileged insider attack	Yes	Yes	Yes	Yes	No	Yes
Impersonation attack	Yes	Yes	No	No	No	Yes
Replay attack	Yes	Yes	Yes	Yes	Yes	Yes
Man-in-the-middle attack	Yes	Yes	Yes	No	No	Yes
Password change phase	Yes	Yes	No	No	Yes	Yes
Formal proof (BAN logic)	Yes	Yes	No	No	No	Yes
Formal verification (AVISPA)	Yes	No	Yes	No	No	Yes
Authentication based on contextual factors	No	No	No	No	No	Yes
Authentication based on transaction history	No	No	No	No	No	Yes

additional functionality and security features are not provided by Kumar et al.'s scheme [7] such as mutual authentication between user and smart device, mutual authentication between user and gateway, password guessing attack, password change attack, stolen smart phone/smart card attack and password change phase, physical context awareness (i.e., location awareness), and transaction history authentication.

Therefore, the communication cost analysis shows that our proposed scheme is effective and feasible for smart homes.

5.3. Computational cost

In this Subsection, we conduct the computation cost analysis of our proposed protocol. In order to ensure a precise computation cost of our protocol, the experimental data reported in [19] [20] are applied. They defined the terms T_{exp}, T_h, and T_E/T_D as the computational time for modular exponentiation operation, hash function h(.), and symmetric encryption/decryption, respectively.

Table 5 shows the time needed for executing those operations. However, the bitwise XOR operation execution time is negligible. Our protocol performs 10 hash invocations and 8 XOR operations, which yields a total computation cost ($10 \times T_h$). Hence, the computation cost of our proposed protocol is (10×0.32 ms) = 3.2 ms.

Table 6 shows a comparison of computation cost between the proposed scheme and other most related schemes in ms.

Table 7 provides security and functionality features compared to other existing related schemes.

In summary, our scheme achieves significantly better performance, security and functionality features as compared to those of other existing schemes.

6. Conclusion

Security and privacy issues are significant obstacles that impedes the large-scale applications of smart home. In previous research, there are almost no robust authentication schemes suitable for smart home ecosystem. As a step in the right direction, we proposed a lightweight and secure two-factor anonymous authentication scheme. The proposed scheme grants a legal user mutually authenticate with the smart device through *GWN*. By the end of successful mutual authentication, a symmetric session key *SK* for future secure communications is established between the user and the smart device. The security of the proposed scheme is formally proved using widely-accepted the BAN logic. Moreover, the informal security verification demonstrate that the proposed scheme resists most common attacks. Finally, the formal security is evaluated using the AVISPA tool and the results indicate that our scheme is safe. The following are our plans for future work:

We will use OMNet++ to implement the proposed scheme which is used to simulate computer networks protocols.

Declaration of Competing Interest

The authors declare that they do not have any financial or nonfinancial conflict of interests.

References

[1] R. Taylor, D. Baron, D. Schmidt, The world in 2025-predictions for the next ten years, in: 2015 10th International Microsystems, Packaging, Assembly and Circuits Technology Conference (IMPACT), IEEE, 2015, pp. 192–195.
[2] C. Kolias, G. Kambourakis, A. Stavrou, J. Voas, Ddos in the iot: Mirai and other botnets, Computer 50 (7) (2017) 80–84.
[3] J. Jeong, M.Y. Chung, H. Choo, Integrated otp-based user authentication scheme using smart cards in home networks, in: Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), IEEE, 2008. pp. 294–294
[4] B. Vaidya, J.H. Park, S.-S. Yeo, J.J. Rodrigues, Robust one-time password authentication scheme using smart card for home network environment, Computer Communications 34 (3) (2011) 326–336.
[5] H.J. Kim, H.S. Kim, Auth hotp-hotp based authentication scheme over home network environment, in: International Conference on Computational Science and Its Applications, Springer, 2011, pp. 622–637.
[6] F.K. Santoso, N.C. Vun, Securing iot for smart home system, in: 2015 International Symposium on Consumer Electronics (ISCE), IEEE, 2015, pp. 1–2.
[7] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttila, M. Sain, Lightweight and secure session-key establishment scheme in smart home environments, IEEE Sensors Journal 16 (1) (2015) 254–264.
[8] M. Wazid, A.K. Das, V. Odelu, N. Kumar, W. Susilo, Secure remote user authenticated key establishment protocol for smart home environment, IEEE Transactions on Dependable and Secure Computing (2017).
[9] M. Shuai, N. Yu, H. Wang, L. Xiong, Anonymous authentication scheme for smart home environment with provable security, Computers & Security (2019).
[10] M. Branman, 6 fastest cars in the world right now, 2019, (https://www.themanual.com/auto/fastest-cars-in-the-world/). Online; accessed 3 November 2019.
[11] M. Burrows, M. Abadi, R.M. Needham, A logic of authentication, Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 426 (1871) (1989) 233–271.
[12] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P.H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, et al., The avispa tool for the automated validation of internet security protocols and applications, in: International conference on computer aided verification, Springer, 2005, pp. 281–285.
[13] D. Dolev, A. Yao, On the security of public key protocols, IEEE Transactions on information theory 29 (2) (1983) 198–208.
[14] L. Viganò, Automated security protocol analysis with the avispa tool, Electronic Notes in Theoretical Computer Science 155 (2006) 61–86.
[15] D. Basin, S. Mödersheim, L. Vigano, An on-the-fly model-checker for security protocol analysis, in: European Symposium on Research in Computer Security, Springer, 2003, pp. 253–270.
[16] M. Turuani, The cl-atse protocol analyser, in: International Conference on Rewriting Techniques and Applications, Springer, 2006, pp. 277–286.
[17] A. Armando, L. Compagna, Satmc: a sat-based model checker for security protocols, in: European workshop on logics in artificial intelligence, Springer, 2004, pp. 730–733.
[18] O. Elkeelany, M.M. Matalgah, K.P. Sheikh, M. Thaker, G. Chaudhry, D. Medhi, J. Qaddour, Performance analysis of ipsec protocol: encryption and authentication, in: 2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No. 02CH37333), 2, IEEE, 2002, pp. 1164–1168.
[19] D. He, N. Kumar, J.-H. Lee, R.S. Sherratt, Enhanced three-factor security protocol for consumer usb mass storage devices, IEEE Transactions on Consumer Electronics 60 (1) (2014) 30–37.
[20] C.-C. Lee, C.-T. Chen, P.-H. Wu, T.-Y. Chen, Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices, IET Computers & Digital Techniques 7 (1) (2013) 48–55.