Computers, Materials & Continua
DOI:10.32604/cmc.2021.015310
Article

Tech Science Press

Secure Multifactor Remote Access User Authentication Framework for IoT Networks

Mohammed Mujib Alshahrani*

College of Computing and Information Technology, The University of Bisha, Saudi Arabia
*Corresponding Author: Mohammed Mujib Alshahrani. Email: mmalshahrani@ub.edu.sa
Received: 14 November 2020; Accepted: 08 January 2021

Abstract: The term IoT refers to the interconnection and exchange of data among devices/sensors. IoT devices are often small, low cost, and have limited resources. The IoT issues and challenges are growing increasingly. Security and privacy issues are among the most important concerns in IoT applications, such as smart buildings. Remote cybersecurity attacks are the attacks which do not require physical access to the IoT networks, where the attacker can remotely access and communicate with the IoT devices through a wireless communication channel. Thus, remote cybersecurity attacks are a significant threat. Emerging applications in smart environments such as smart buildings require remote access for both users and resources. Since the user/building communication channel is insecure, a lightweight and secure authentication protocol is required. In this paper, we propose a new secure remote user mutual authentication protocol based on transitory identities and multi-factor authentication for IoT smart building environment. The protocol ensures that only legitimate users can authenticate with smart building controllers in an anonymous, unlinkable, and untraceable manner. The protocol also avoids clock synchronization problem and can resist quantum computing attacks. The security of the protocol is evaluated using two different methods: (1) informal analysis; (2) model check using the automated validation of internet security protocols and applications (AVISPA) toolkit. The communication overhead and computational cost of the proposed are analyzed. The security and performance analysis show that our protocol is secure and efficient.

Keywords: Internet of things; threats; smart building; attacks; remote access; authentication; smart buildings

1 Introduction

In the last few years, the world has witnessed a huge revolution in information and computing technologies of the 21st century. Internet of Things (IoT) is one of the most emerging releases of this revolution [1].

The core concept of IoT is adding sense to non-living objects to perform the information processing and take decisions automatically without any presence of human or living bodies'

functionality in the system, and bring all physical entities in one affordable integrated digital fold sharing defined communication system among it [2].

IoT network composes various physical entities together in the same networks, and most of these entities are not well manufactured for IoT systems. Hence, these embedded devices do not include and support high-security mechanisms. Thus, it will be exposed to various malicious attacks in both privacy and security dimensions. While the cybersecurity issues are gradually increasing with the increase of the IoT capacities of adapting entities, the heterogeneous entities introduce new security and privacy issues. Recently, many IoT devices were brought down by a distributed denial of service (DDoS) attack carried out using the Mirai IoT botnet [3]. Mirai propagates by brute-forcing IoT devices using a list of common, default passwords to attempt to log into the IoT devices. Thus, depending only on password-based authentication is a weak method as passwords are often easy to guess by attackers launching brute-force attacks.

1.1 Related Work

Several proposals have been published on user authentication [4–8]. Recent progress in user authentication has focused on incorporating context information. Many contributions incorporated various types of context information, namely physical contexts, device contexts, and historical information, among others.

Among these contributions, Jeong et al. [4] introduced a one-time password-based user authentication scheme using a smart card for a smart home. This scheme is lightweight because it depends on one-way hash function operations. However, the scheme failed to achieve some important security properties: mutual authentication between GWN and smart devices, traceability, unlinkability, and user anonymity.

Han et al. [9] proposed a secure key agreement scheme for smart home systems. The proposed scheme is suitable for consumer electronics devices in a smart home.

Li [10] proposed a lightweight key establishment protocol, and an initial session key was established between the nodes and control. The mutual authentication between the user and control was not provided in their scheme.

Santoso et al. [11] introduced a remote user authentication scheme for a smart home using elliptic curve cryptography (ECC). The authors did not achieve two key security properties, named anonymity and traceability. Furthermore, the scheme is susceptible to privileged-insider and stolen smart card attacks.

Kumar et al. [6] introduced a lightweight and secure session key establishment scheme for IoT smart homes. They capitalized on a short authentication token to establish a session key between and smart device.

Recently, Shuai et al. [12] proposed a remote authentication scheme for smart homes using ECC. The authors eliminate the need to store the verification table for authentication purposes. However, the authors failed to provide satisfactory performance.

1.2 Motivations and Contributions

Although researchers have proposed some remote user authentication schemes for smart buildings, they are not lightweight or secure enough to be suitable for the smart building nodes due to resources-constraints nature of such sensors. Besides, few mutual authentication and key agreement schemes have been proposed. Moreover, almost none of these contributions considered using physical context awareness (i.e., location awareness) and transaction history for

authentication. Motivated by the importance of authentication based on location awareness and transaction history for remote access in IoT smart building, this paper devoted to design a secure scheme not only provides mutual authentication but also achieves some important security properties such as anonymity, untraceability and unlinkability of transmitted information as well as authentication based on location awareness and transaction history. The contributions of this paper are as follows:

- We propose a new anonymous remote user mutual authentication protocol designed for the IoT smart building network. Our protocol guarantees key security properties: confidentiality, integrity, anonymity, unlinkability, and untraceability [13].
- Our protocol avoids the clock synchronization problem by not relying on timestamps to ensure safe protection against message reuse attacks, and it also can stop quantum computing attacks.
- We propose location awareness and transaction history to improve the authentication process of the user remote access of IoT smart buildings.
- The security of our protocol is proved by using the widely accepted Burrows–Abadi–Needham (BAN) logic, and assessed by using the AVISPA simulator tool. Besides, an informal security analysis of the proposed scheme is discussed.
- We compare our proposed protocol with other related protocols. Comparison results show that our protocol is more secure and efficient than the previously proposed related protocols.

1.3 Organization of the Paper

The remaining parts of this paper are structured as follows. In Section 2, we describe the proposed system model. In Section 3, we describe in depth our secure protocol. In Section 4, we evaluate the security of our protocol. In Section 5, we evaluate the performance of our protocol. Finally, Section 6 concludes the paper.

2 System Model

This section introduces the used network model and adversary model of the proposed scheme.

2.1 Network Model

The network model consists of four entities, namely registration authority (RA), the end-user, the smart building controller node (CRN), and the smart device (SD) in the building (see Fig. 1).

Every time the end-user wishes to access the IoT smart building devices remotely, they have to provide the building controller with their current location. Moreover, we employ transaction history in our system in a lightweight way. We cryptographically hash the transactions between the building controller and the user in a secure way. Then, we capitalize on these hashed transactions to improve the authentication process in our system. Thus, we combine location awareness and hashed transaction history to enhance user remote access to IoT smart buildings.

We introduced two different techniques that were derived from the transaction histories: first, a robust cumulative cryptographically-hashed historical transactions (CCHH) technique based on a one-way cryptographic hash function is used to validate and ensure the integrity of the data transmitted between the end-user and the building controller, and to maintain the anonymity of the communication parties; second, a robust cumulative location tracker (CLT) technique is used to ensure the genuineness of the user and the freshness of the temporary session keys. These CCHH and CLT techniques are briefly explained, as follows:

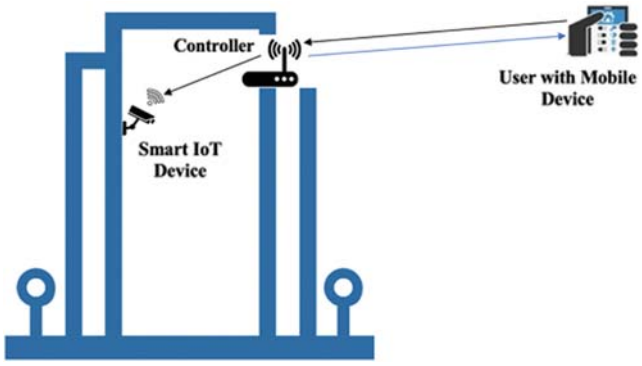

Figure 1: High-level architecture of smart building environment

- Cumulative cryptographically-hashed historical transactions (CCHH)

At the core of the proposed protocol is a lightweight challenge-response mechanism that relies on a chain of cryptographically-hashed historical transactions (CCHH) of all communication sessions between the user and the controller. The user and the controller maintain a synchronized database, called CCHH database, containing hashes generated during previous authentication sessions singed with the temporal session secret keys that change in every session. The synchronized CCHH database's values are utilized in generating the transitory identities of both the user and the controller, where transitory identities TID is constructed by hashing the real identity and the ^{x}CCHH. Note that x denotes the corresponding cumulative cryptographically-hashed historical transactions value that is stored in CCHH database. The use of the transitory identities improves the smart building's privacy by achieving key security properties, namely, anonymity, unlinkability and untraceability.

- Cumulative location tracker (CLT)

A physical context awareness, namely location, is employed in our system to check if a mobile device's current location is approximate to the previously recorded location within a given time. The location of the mobile device is checked using the linear motion equation (ΔL_{MAX} = V * ΔT) to calculate the time it would take a given user to move from a previous location to a current location, where ΔT denotes the time required for a user to move from the previous location Lp to the current location Lc. In contrast, V denotes the maximum velocity that the user could have. These locations from all sessions (previous locations + current locations) between the mobile device and the controller are cumulatively hashed and stored securely in a synchronized database, called CLT database, maintained by both the mobile device and the controller. These synchronized databases of cumulative hashes are capitalized on to achieve a challenge-response authentication between the mobile device and the controller to stop remote attacks such as Mirai attack. Meaning that, in case of suspicion of remote access attack, the controller challenges the knowledge of the mobile device about the previous locations stored in the database, the mobile device then has to send back the correct corresponding location; Otherwise, the session is terminated, and the mobile device is flagged as a malicious. Moreover, the synchronized CLT database's values can generate the temporal session secret keys between the mobile device and the controller. The session secret SK is constructed by hashing the secret key and the ^{x}CLT. Note that x denotes the corresponding cumulative cryptographically-hashed location tracker value that is stored in CLT database.

In our system, data security between the end-user and the IoT smart device is ensured by encrypting the payload using the AES 128 CCM algorithm (16 Bytes). According to a report on lightweight cryptography done by Kerry McKay from The National Institute of Standards and Technology [14], AES 128 CCM (Counter with CBC-MAC) mode is by far the most widely used-symmetric key algorithm. It can also be chosen which would provide additional benefit of data integrity.

2.2 Adversary Model

To evaluate the security properties of the proposed scheme, we define the adversarial model as follows.

- The CRN is assumed trustworthy. However, an attacker may be able to infiltrate HN's database. He may steal or manipulate database information.
- The attacker can eavesdrop on all communication links in the network. He can also damage or replace transmitted messages or replay previously sent messages.
- An attacker can capture any IoT node N.
- We consider the well-known Dolev–Yao threat model [15]. It assumes that two communicating parties communicate over an insecure channel. We rely on this model to provide the security analysis and simulation of our scheme.

3 Proposed Scheme

In this section, a secure remote user access authentication protocol based on transitory identities and multi-factor authentication is presented for IoT smart building systems, which resists all known attacks and supports the desirable security features. The abstract notations used to describe our authentication protocol are listed in Tab. 1. The proposed protocol consists of four phases: (1) initialization phase; (2) registration phase; (3) login and authentication phase; and (4) password change phase. These phases are explained as follows:

3.1 Initialization Phase

The manufacturer does the initialization phase before the devices are handed to the owner of the smart building. The mobile device will be loaded with a unique symmetric key *Kur* shared between the registration authority RA and the mobile device. Lastly, the controller will be loaded with a unique symmetric key *Kgr* shared between the registration authority and controller.

3.2 Registration Phase

When the user first turns on the mobile device, the user picks up an identity ID_{Ui} and a password PWi. Next, the user sends his/her identity and password to RA in a secure way, using the shared symmetric key Kur.

When RA receives the message, it will store the mobile device information, and generate a temporary identity for the user ID_{Ui} and temporary secret key TSK, and computes the following parameters:

$$\mathrm{S1} = \mathrm{h}(\mathrm{ID_{Ui}}, \mathrm{TSK}, \mathrm{SN}, \mathrm{DMN}, \mathrm{ESN}) \tag{1}$$

$$S2 = h(ID_{Ui}, Pwi, ID_{Ui}, TSK) \tag{2}$$

where SN, DMN, ESN are context information of the mobile device namely serial number, device manufacturer name, and unique equipment serial number, respectively.

Table 1: Notations used in our protocol

Notation	Description
Ui	Mobile User
CRN	Smart building controller
SD	Smart device
RA	Registration authority
ID_{Ui}	Real identity of user
IDCRN	Real identity of smart building controller
ID_{SD}	Real identity of sensor node N
TID_{Ui}	Transitory identity of user
DID_{CRN}	Dynamic identity of smart building controller
KMFR	Master secret key of Manufacturer MFR
Kur	Secret key shared between the registration authority and each mobile device
Kgr	Secret key shared between the registration authority and smart building controller
N	Random number picked by CRN used as a session key between Ui and SD
Lc	The current location
Lp	The previous location
TSK	Secret key picked by and sent to Ui and CRN
SK	Secret key
ISV1	Initialization Seed Value picked by RA for CCHH database
ISV2	Initialization Seed Value picked by RA for CLT database
HF	keyed-hash message authentication cod
h(·)	Hash function
${}^{X}CCHH$	Cumulative cryptographically-hashed historical transactions value
${}^{X}CLT$	Cumulative location tracker value
$M1 \| M2$	Concatenate operation
$\oplus$	Bitwise XOR operation
Ui $\rightarrow$ CRN: M	Ui sends the message M to controller CRN via a public channel

RA also generates INV1 and INV2 to be used as initial values by both mobile device and controller for CKH and CLT technique, respectively. Then, RA sends {ID_{Ui}, TSK, S1, S2, INV1, INV2} and {ID_{Ui}, TSK, SN, DMN, ESN, S1, INV1, INV2} to Ui and controller using the shared secret keys Kur and Kgr, respectively (see Fig. 2).

3.3 Login Phase

The user enters his/her identity ID_{Ui} and password PWi into mobile device, the mobile device computes *S1 = h(ID_{Ui}, TSK, SN, DMN, ESN) and checks if *S1 = S1. If they match, the user is considered legitimate and can access the application on his/her mobile device. Otherwise, the mobile device drops the login request, increments the value of the counter by 1, and check if it reaches the predetermined value, for instance, 3. If the number of attempts exceeds the predetermined value, the mobile device terminates the login request immediately until the user re-register.

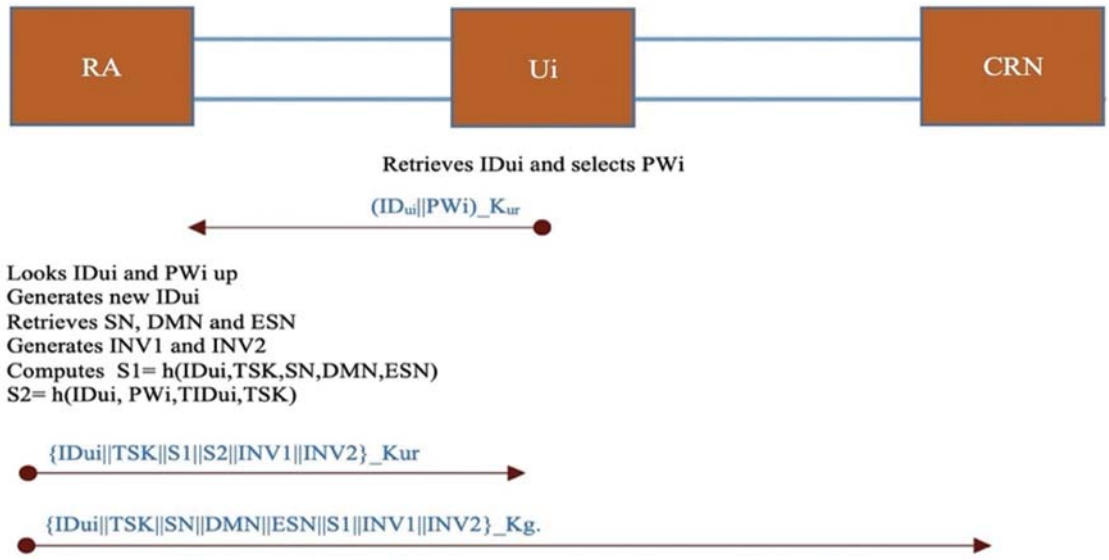

Figure 2: Registration phase of the proposed protocol

3.4 Authentication Phase

When Ui wishes to control any IoT device in the smart building, it will have to authenticate themselves with the controller first. Hence, they will follow the below steps:

A: At User Mobile Device:

(1) Ui accesses the mobile device and enter the correct username ID_{Ui} and password PW_i.
(2) Mobile device computes $*S1$ = h(ID_{Ui}, PW_i, SN, DMN, ESN).
(3) Mobile device checks if $*S1$ = S1.
(4) Mobile device computes TID_{Ui1} = h(ID_{Ui}, ISV1)
(5) Mobile device computes DID_{CRN1} = h(ID_{CRN}, ISV1)
(6) Mobile device computes SK1 = h(TSK, ISV1)
(7) Mobile device extracts current location Lc
(8) Mobile device creates the message UC = Lc ‖ $ID_{SD}\oplus$ S1
(9) Mobile device Prepare M={DID_{CRN1},TID_{Ui1}, UC}
(10) Mobile device computes HF = h(M, SK1, ISV1)
(11) Mobile device sends {DID_{CRN1},TID_{Ui1}, UC, HF} to the controller
(12) Mobile device updates ^{x}CCHH = h(HF, SK1)
(13) Mobile device updates ^{x}CLT = h(Lc, INV2)

B: At Smart Building Controller Device:

(1) Controller receives the message {DID_{CRN1}, TID_{Ui1}, UC, HF}
(2) Controller checks DID_{CRN1} and TID_{Ui1}
(3) Controller computes SK1 = h(TSK, ISV1)
(4) Controller computes M = {DID_{CRN1},TID_{Ui1}, UC}
(5) Controller checks *HF = h(M, SK1, ISV1)
(6) Controller updates ^{x}CCHH = h(HF, SK1)
(7) Controller computes Lc ‖ ID_{SD} = UC $\oplus$ S1

(8) Controller checks the genuineness of Lc using the formula (ΔL_{MAX} = V * ΔT). If it is legitimate, the controller will update the cumulative location tracker ^{x}CLT = h(Lc, INV2). Otherwise, controller challenges mobile devices knowledge of one of the previous CLT values.
(9) Controller generates a nonce N (session secret key between Ui and SD).
(10) Controller sends the {TID_{Ui1}, N} to SD in a secure way.
(11) Controller computes SK2 = h(SK1, ^{x}CCHH)
(12) Controller computes CU = DID_{SD} || N $\oplus$ S1
(13) Controller prepares M = {TID_{Ui1}, DID_{CRN1}, CU}
(14) Controller computes HF = h(M, SK2, ^{x}CCHH)
(15) Controller sends {TID_{Ui1}, DID_{CRN1}, CU, HF} to Ui
(16) Controller updates $^{x+1}$CCHH = h(HF, SK2)
(17) Controller computes TID_{Ui2} = h(ID_{Ui}, $^{x+1}$CCHH) for upcoming session with Ui
(18) Controller computes DID_{CRN2} = h(ID_{CRN}, $^{x+1}$CCHH) for upcoming session with Ui

C: At User Mobile Device:

(1) Mobile device computes SK2 = h(SK1, ^{x}CCHH)
(2) Mobile device checks *HF = h(M, SK2, ^{x}CCHH)
(3) Mobile device computes CU $\oplus$ S1 = DID_{SD}|| N
(4) Mobile device updates $^{x+1}$CCHH = h(HF, SK2)
(5) Mobile device uses DID_{SD} and N to establish a secure communication session with SD.
(6) Mobile device computes TID_{Ui2} = h(ID_{Ui}, $^{x+1}$CCHH) for upcoming session with CRN
(7) Mobile device computes DID_{CRN2} = h(ID_{CRN}, $^{x+1}$CCHH) for upcoming session with CRN

The above steps are summarized in Fig. 3. By the end of processing each message, CCHH database is updated (see Fig. 4), and by the end of each session, CLT table is updated (see Fig. 5).

3.5 Password Update Phase

In this section, the user Ui can change his/her password without any interaction with CRN by performing the following operations.

(1) Ui enters the identity ID_{Ui} and the password PWi into GUI of mobile device.
(2) Mobile device computes *S1 and checks if *S1 = S1. If it is not hold, the mobile device rejects the password change request. Otherwise, the mobile allows Ui to enter a new ID_{Ui} and Pwi.
(3) The mobile device then transmits the new ID_{Ui} and PWi to RA.
(4) The RA updates S1 and sends it to CRN.

3.6 Challenge-Response Mechanism Based on Transaction History

As aforementioned in the previous section, Ui and CRN securely maintain two synchronized databases, namely CCHH and CLT, of cumulative hashed values. These values can be capitalized on to introduce a historical factor for authenticating Ui, as illustrated in Fig. 6. Authentication using a historical factor helps us achieve mutual authentication through a challenge-response process, where mutual authentication is so important in securing communication between devices. This two-way challenge/response allows the controller to verify the authenticity of Ui, so that it can stop malicious attacks such as the Mirai attack. Cumulative hash history-based authentication challenges Ui to show a proof of knowledge of past cumulative hash values. The approach involves securely storing the cumulative hash values related to the interaction over time between

the Ui and CRN in CCHH and CLT databases. Thus, when CRN receives an authentication request message from Ui, it triggers a challenge/response process. It generates a challenge c (information about random cumulative hash value ${}^{x}CCHH$ stored previously), hashes the challenge c with the secret key and ${}^{x}CLT$ h(SK, c, ${}^{x}CLT$), and sends it to Ui.

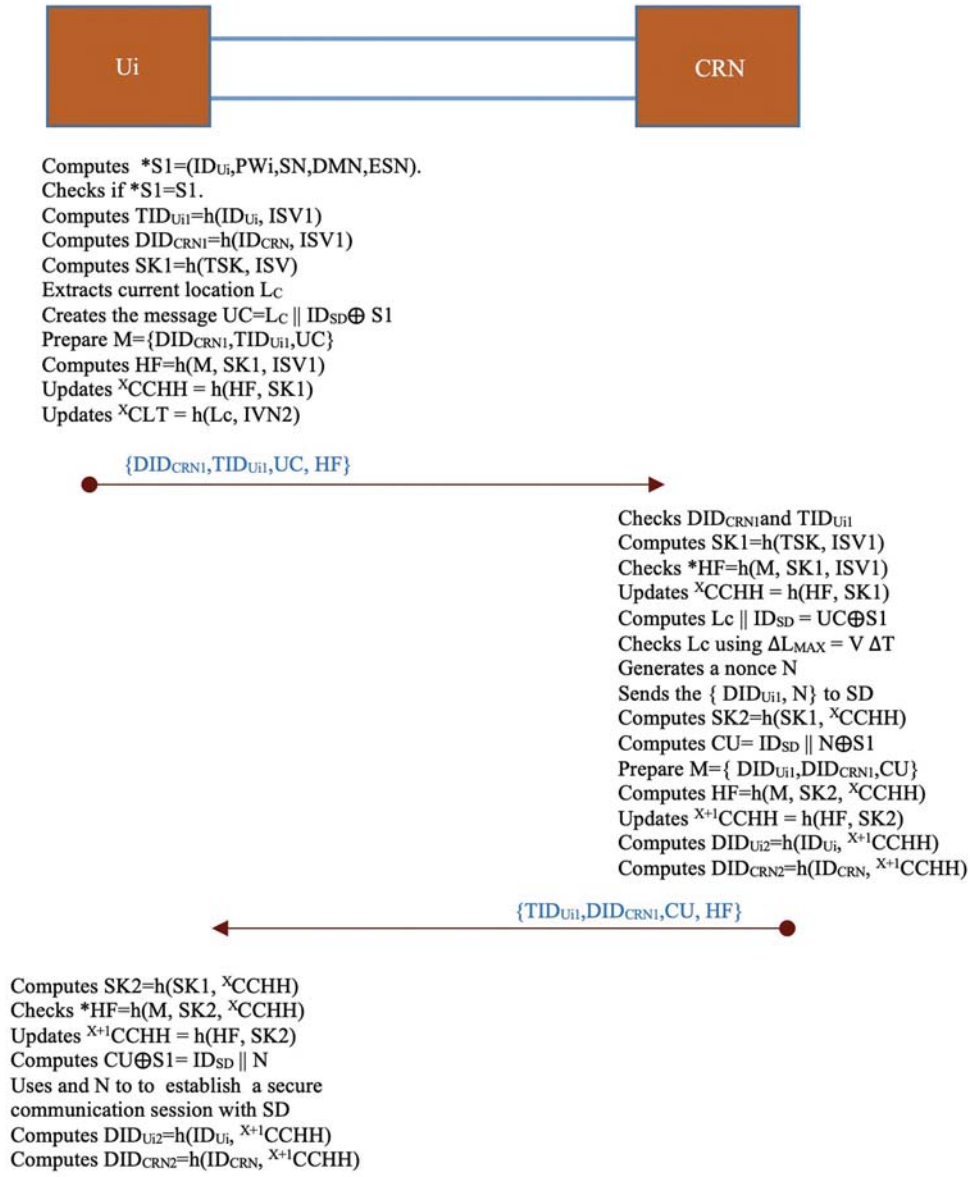

Figure 3: Authentication phase of the proposed scheme

Ui sends the response back using h(SK, r, ${}^{x}CLT$), where r is the response (cumulative hash value ^{x}CCHH). Once CRN receives the response, it checks if the received ${}^{x}CCHH$ value is correct. If yes, CRN will accept Ui and resume the authentication process. Otherwise, Ui will be rejected and flagged as malicious.

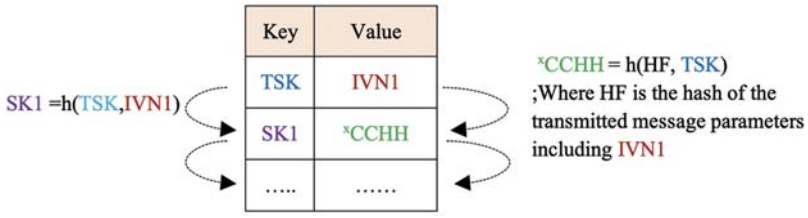

Figure 4: Cumulative cryptographically-hashed historical transactions (CCHH) table

Session	Key	Location	Value
1	N	Lc	IVN2
2	N1	Lc1	*CLT
....	....	....	....

*CLT = h(Lc, IVN2)

Figure 5: Cumulative location tracker (CLT) table

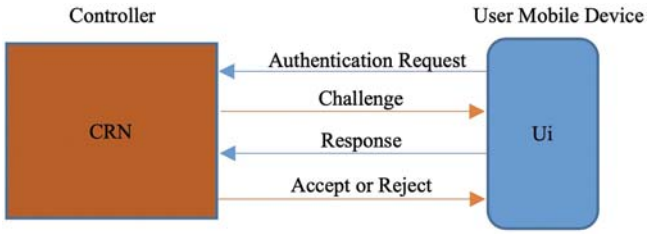

Figure 6: Historical authentication process

It is worth mentioning that the challenge-response mechanism is triggered by CRN when the Lc, that is provided by Ui, is not approximated.

4 Security Analysis

In this section, we discuss different known attacks, and we explain how our protocol successfully resists such attacks.

4.1 Informal Security Analysis

In the following, we analyze different important adversarial attacks/security properties and how our scheme stops these attacks and achieves these properties.

4.1.1 Replay Attack

The replay attack is defeated using CCHH and CLT's cumulative values that security change in every message. Furthermore, the mobile device and the controller use secure unique identities in every session. Besides, the keyed-hash message authentication cod (HF) value, which is attached in each message, changes in every single message. Hence, replay attack is detected.

4.1.2 Eavesdropping Attack

In the authentication phase of our scheme, an adversary A can record all transmitted parameters between Ui and CRN. He collects the tuple $\langle DID_{CRN1}, TID_{Ui1}, UC, HF \rangle$ from Ui to CRN,

and the tuple ⟨TID_{Ui2}, DID_{CRN2}, CU, HF⟩ from CRN to Ui. Notice that the session key SK1 = h(TSK, ISV1). From the intercepted parameters, A cannot reach TSK and ISV1 because they are protected by the one-wayness of h (.). Moreover, A cannot compute ^{x}CCHH = h(HF, SK1) or ^{x}CLT = h(Lc, INV2) because he does know SK1 or Lc or INV2. The same is applied to the parameters sent from CRN to Ui. Therefore, the privacy of the SK1, SK2, ^{x}CCHH, ^{x}CLT, and Lc are preserved, and hence, the scheme protects against an eavesdropping attack.

4.1.3 Impersonation Attack

This attack is stopped using the TID_{Ui} and HF, which are protected using the one-way hash function. Besides, TID_{Ui} is constructed from different secure parameters $TID_{Ui=}$ h(ID_{Ui}, ISV1), and change in every message as ISV1 is updated in every message. Hence, the attacker is unable to create a valid temporary identity without the corresponding ID_{Ui}, ISV1.

4.1.4 Man-in-the-Middle Attack

Our protocol is protected against this attack using TID_{Ui}, N, and HF. So, this attack can be defeated.

4.1.5 Attack Against the Temporary Secret Key

This attack is defeated using the temporary secret key SK and N, which change in every session. Moreover, SK is constructed using secure parameters and protected using a one-way hash function.

4.1.6 Forward/Backward Security

The forward/backward security is an important security property, which means that any past or future sessions keys will not be affected when any temporary session key is exposed. Forward/backward security is achieved using the *SK* and N, which dynamically change in every session.

4.1.7 Session Key Guessing Attack

This attack is defeated using the *SK* and N, which dynamically change in every session.

4.1.8 Quantum Attacks

Recent advances in quantum computing put the security of the current IoT at risk using these cryptographic schemes. Grover's algorithm speeds up this process of brute force search dramatically using quantum computers. Thus, we rely on hash functions and symmetric schemes that are relatively easy to prevent quantum attacks by enlarging key and output sizes.

4.1.9 User Credentials Attack

In our proposed protocol, the user UI never stores its identity IDUi and password PWi credentials in its mobile device's memory because it stores the hash value S1, contributing to verifying IDUi and PWi entered by the user. When the attacker tries to obtain user credentials from S1 physically, they will fail as the one-way hash function protects S1. Hence, our proposed protocol can successfully stop the user credentials attack.

4.1.10 User Anonymity, Unlinkability and Untraceability

User anonymity unlinkability and untraceability are crucial security properties in the authentication. Anonymity ensures the mobile device's real identity is kept secure and the mobile device remains unidentifiable among the other set of devices. Thus, the attacker cannot identify the

devices' real identities as the real identity of the mobile device is kept secure and we use transitory identities that change in every session. We also ensured that an attacker cannot link the different sessions initiated by a particular mobile device to the same UI. Also, the adversary cannot relate two or more sessions to the same Ui. Hence, our protocol achieves anonymity, unlinkability, and untraceability of the conducted sessions.

4.1.11 Authentication Based on Cumulative Hashed Transaction History and Location

GPS location is utilized in our protocol to check whether mobile device's previous location is proximate to the current location. Tracking the GPS location of Ui will contribute to stopping remote cybersecurity attacks such as the as discussed in this Section Mirai attack. Additionally, both CRN and Ui maintain a synchronized database of cumulative hashes generated from the previous sessions, as discussed in this Section. These synchronized databases improve the overall security by applying the challenge/response mechanism and ensuring the uniqueness and freshness of the identities and established sessions; thus, securing the smart building system from known attacks.

4.2 Formal Proof Based on BAN Logic

The BAN logic was introduced by Burrows et al. [16] in 1989. It is a widely accepted model to describe and analyze authentication protocols. It has been widely employed to verify the protocols' security and provide proof of correctness of the authentication protocols [17]. Hence, we capitalize on it to formally prove that our authentication scheme achieves mutual authentication between an IoT device N and controller C.

We start by presenting a summarized introduction about the important symbols and the rules of BAN logic. Then, we will proceed with our formal proof.

4.2.1 BAN Logic Overview

Let N (client) and C (server) be participators, and let X and Y denote a parameter, formula or expression. We define the following notations:

- N $|\equiv$ X: N believes the statement X.
- # (X): X is fresh.
- N $|\Longrightarrow$X: N has jurisdiction over the statement X.
- N $\lhd$X: N sees the statement X.
- N $|\sim$ X: N once said the statement X.
- (X, Y): X or Y is one part of the formula (X, Y).
- $\langle X \rangle_Y$: X combined with Y.
- N $\overset{K}{\leftrightarrow}$ C: K is a secret parameter shared (or to be shared) between N and C.
- N $\rightleftharpoons$ C: X is a secret known only to N and C, and possibly to parties trusted by them.

Furthermore, the following commonly used BAN logic rules are utilized to prove that our authentication scheme ensures secure mutual authentication and key agreement, as follows:

- Message meaning rule: If N sees X encrypted with Y and if N believes Y is a secret key shared with C, then N believes C once said X.

$$\frac{N|\equiv N \overset{Y}{\leftrightarrow} C, N \lhd \langle X \rangle_Y}{N|\equiv C|\sim X} \tag{3}$$

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

$$\frac{N \mid\equiv \#(X), N \mid\equiv C \mid\sim X}{N \mid\equiv C \mid\equiv X} \tag{4}$$

- Jurisdiction rule: If N believes C has jurisdiction over X and N believes C believes X, then N believes X.

$$\frac{N \mid\equiv C \mid\Rightarrow (X), N \mid\equiv C \mid\equiv X}{N \mid\equiv X} \tag{5}$$

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

$$\frac{N \mid\equiv \#(X)}{N \mid\equiv \#(X, Y)} \tag{6}$$

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

$$\frac{N \mid\equiv (X, Y)}{N \mid\equiv X} \tag{7}$$

- Observation rule: If N sees X and Y, then N sees X.

$$\frac{N \lhd (X, Y)}{N \lhd X} \tag{8}$$

4.2.2 Goals of the Analysis of our Authentication Scheme

In this section, we define the main goals of the analysis of our authentication scheme as follows:

- Goal 1: CRN believes Ui believes $SK1$ is a secure, shared parameter between Ui and CRN.

$$CRN \mid\equiv Ui \mid\equiv (Ui \overset{SK1}{\leftrightarrow} CRN) \tag{9}$$

- Goal 2: CRN believes $SK1$ is a secure, shared parameter between Ui and CRN.

$$CRN \mid\equiv (Ui \overset{SK1}{\leftrightarrow} CRN) \tag{10}$$

- Goal 3: Ui believes CRN believes $SK2$ is a secure, shared parameter between Ui and CRN.

$$Ui \mid\equiv CRN \mid\equiv (Ui \overset{SK2}{\leftrightarrow} CRN) \tag{11}$$

- Goal 4: Ui believes $SK2$ is a secure, shared parameter between Ui and CRN.

$$Ui \mid\equiv (Ui \overset{SK2}{\leftrightarrow} CRN) \tag{12}$$

4.2.3 Messages Transferred in the Authentication

The idealized messages that are exchanged in the authentication phase between a user Ui and the controller CRN are listed below:

- M1: Ui $\rightarrow$ CRN: $\langle DIDCRN1, DIDUi1, \mathrm{UC}, \mathrm{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} CRN}$
- M2: CRN $\rightarrow$ Ui: $\langle DIDUi2, DIDCRN2, \mathrm{CU}, \mathrm{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} CRN}$

4.2.4 Introductory Assumptions

The fundamental assumptions of our authentication scheme are as follows:

- A1: CRN believes ISV1is fresh: CRN|$\equiv$ #(ISV1)
- A2: CRN believes *HF* is fresh: CRN|$\equiv$ #(HF)
- A3: Ui believes CCHH is a secure, shared parameter between Ui and CRN: Ui |$\equiv$ (Ui $\overset{\mathrm{CCHH}}{\leftrightarrow}$ CRN)
- A4: CRN believes *ISV* 1 is a secure, shared parameter between Ui and CRN: CRN |$\equiv$ (Uj $\overset{\mathrm{ISV1}}{\leftrightarrow}$ CRN)
- A5: Ui believes CRN has jurisdiction over CCHH, SK2, and HF: Ui|$\equiv$ CRN|$\Rightarrow${CCHH, SK2, HF}
- A6: CRN believes Ui has jurisdiction over ISV1, SK1 and HF: CRN|$\equiv$ Ui|$\Rightarrow${ISV1, SK1, HF}

4.2.5 Analysis of our Authentication Scheme

We now start analyzing our authentication scheme to prove that our scheme achieves mutual authentication between Ui and CRN.

S1: According to the M1, we get:

$$\mathrm{CRN} \ \triangleleft \langle \mathrm{DIDCRN1}, \mathrm{DIDUi1}, \mathrm{UC}, \mathrm{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \mathrm{CRN}}$$

S2: From assumption A4 and S1, and by applying the message meaning rule, we derive:

$$\frac{\mathrm{CRN} \ |\equiv (\mathrm{Uj} \overset{\mathrm{ISV1}}{\leftrightarrow} \mathrm{CRN}\ ), \mathrm{CRN} \ \triangleleft \langle \mathrm{DIDCRN1}, \mathrm{DIDUi1}, \mathrm{UC}, \mathrm{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \mathrm{CRN}}}{\mathrm{CRN} \,|\equiv \mathrm{Ui} \ | \sim \langle \mathrm{DIDCRN1}, \mathrm{DIDUi1}, \mathrm{UC}, \mathrm{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \mathrm{CRN}}}$$

S3: From assumption A1 and by applying the freshness rule, we derive:

$$\frac{\mathrm{CRN} \,|\equiv \#(\mathrm{ISV1})}{\mathrm{GWN} \,|\equiv \#\langle \mathrm{DIDCRN1}, \mathrm{DIDUi1}, \mathrm{UC}, \mathrm{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \mathrm{CRN}}}$$

S4: From derivations S3 and S2, and by applying the nonce verification rule, we derive:

$$\frac{\mathrm{GWN} \,|\equiv \#\langle \mathrm{DIDCRN1}, \mathrm{DIDUi1}, \mathrm{UC}, \mathrm{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \mathrm{CRN}}, \ \mathrm{CRN} \,|\equiv \mathrm{Ui}| \sim \langle \mathrm{DIDCRN1}, \mathrm{DIDUi1}, \mathrm{UC}, \mathrm{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \mathrm{CRN}}}{\mathrm{CRN} \,|\equiv \mathrm{Ui} \,|\equiv \langle \mathrm{DIDCRN1}, \mathrm{DIDUi1}, \mathrm{UC}, \mathrm{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \mathrm{CRN}}}$$

S5: According to the M2, we get:

$$\mathrm{Uj} \ \triangleleft \langle \mathrm{DIDUi2}, \mathrm{DIDCRN2}, \mathrm{HM}, \mathrm{HF}\rangle_{Ui \overset{SK2}{\leftrightarrow} \mathrm{CRN}}$$

S6: From assumption A3 and derivation S5 and by applying the message meaning rule, we derive:

$$\frac{\text{Ui} \mid\equiv (\text{Ui} \overset{\text{CCHH}}{\leftrightarrow} \text{CRN}),\ \text{Uj} \lhd \langle \text{DIDUi2}, \text{DIDCRN2}, \text{HM}, \text{HF}\rangle_{Ui \overset{SK2}{\leftrightarrow} \text{CRN}}}{\text{Ui} \mid\equiv \text{GWN} \mid\sim \langle \text{DIDUi2}, \text{DIDCRN2}, \text{HM}, \text{HF}\rangle_{Ui \overset{SK2}{\leftrightarrow} \text{CRN}}}$$

S7: From assumptions A1 and A2, and applying freshness-conjuncatenation rule, we get:

$$\text{Uj} \mid\equiv \#\langle \text{DIDUi2}, \text{DIDCRN2}, \text{HM}, \text{HF}\rangle_{Ui \overset{SK2}{\leftrightarrow} \text{CRN}}$$

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

$$\frac{\text{Uj} \mid\equiv \#\langle \text{DIDUi2}, \text{DIDCRN2}, \text{HM}, \text{HF}\rangle_{Ui \overset{SK2}{\leftrightarrow} \text{CRN}},\ \text{Ui} \mid\equiv \text{GWN} \mid\sim \langle \text{DIDUi2}, \text{DIDCRN2}, \text{HM}, \text{HF}\rangle_{Ui \overset{SK2}{\leftrightarrow} \text{CRN}}}{\text{Ui} \mid\equiv \text{CRN} \mid\equiv \langle \text{DIDUi2}, \text{DIDCRN2}, \text{HM}, \text{HF}\rangle_{Ui \overset{SK2}{\leftrightarrow} \text{CRN}}}$$

S9: From A6 and derivations S4 and applying jurisdiction rule, we get:

$$\frac{\text{CRN} \mid\equiv \text{Ui} \mid\Rightarrow \{\text{ISV1}, \text{SK1}, \text{HF}\},\ \text{CRN} \mid\equiv \text{Ui} \mid\equiv \langle \text{DIDCRN1}, \text{DIDUi1}, \text{UC}, \text{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \text{CRN}}}{\text{CRN} \mid\equiv \langle \text{DIDCRN1}, \text{DIDUi1}, \text{UC}, \text{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \text{CRN}}}$$

S10: From derivations S3, S4 and applying session keys rule, we get:

$$\frac{\text{CRN} \mid\equiv \#\langle \text{DIDCRN1}, \text{DIDUi1}, \text{UC}, \text{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \text{CRN}},\ \text{CRN} \mid\equiv \text{Ui} \mid\equiv \langle \text{DIDCRN1}, \text{DIDUi1}, \text{UC}, \text{HF}\rangle_{Ui \overset{SK1}{\leftrightarrow} \text{CRN}}}{\text{CRN} \mid\equiv \text{Ui} \mid\equiv (\text{Uj} \overset{SK1}{\leftrightarrow} \text{GWN})} \text{ (Goal 1)}$$

S11: From derivation S10 and assumption A6 and applying jurisdiction rule, we get:

$$\frac{\text{CRN} \mid\equiv \text{Ui} \mid\Rightarrow \{\text{ISV1}, \text{SK1}, \text{HF}\},\ \text{CRN} \mid\equiv \text{Ui} \mid\equiv \left(\text{Uj} \overset{SK1}{\leftrightarrow} \text{GWN}\right)}{\text{CRN} \mid\equiv (\text{Ui} \overset{SK1}{\leftrightarrow} \text{CRN})} \text{ (Goal 2)}$$

S12: From derivations S7and S8 and applying session keys rule, we get:

$$\frac{\text{Uj} \mid\equiv \#\langle \text{DIDUi2}, \text{DIDCRN2}, \text{HM}, \text{HF}\rangle_{Ui \overset{SK2}{\leftrightarrow} \text{CRN}}, \text{Ui} \mid\equiv \text{CRN} \mid\equiv \langle \text{DIDUi2}, \text{DIDCRN2}, \text{HM}, \text{HF}\rangle_{Ui \overset{SK2}{\leftrightarrow} \text{CRN}}}{\text{Ui} \mid\equiv \text{CRN} \mid\equiv (\text{Uj} \overset{SK2}{\leftrightarrow} \text{CRN})} \text{ (Goal 3)}$$

S13: From assumption A5, and derivation S12, and by applying jurisdiction rule, we get:

$$\frac{\text{Uj} \mid\equiv \text{GWN} \mid\Rightarrow \{\text{N2}, \text{SK}, \text{V2}\}, \text{Uj} \mid\equiv \text{CRN} \mid\equiv \left(\text{Uj} \overset{SK2}{\leftrightarrow} \text{CRN}\right)}{\text{Ui} \mid\equiv (\text{Ui} \overset{SK2}{\leftrightarrow} \text{GWN})} \text{ (Goal 4)}$$

Hence, our authentication scheme achieves mutual authentication and key agreement between Ui and CRN.

4.3 Simulation Based on AVISPA Tool

Our protocol is evaluated using the Automated Validation of Internet Security Protocols and Applications (AVISPA) toolkit, which is widely used as a toolkit in the research community for security protocol validation [18]. Fig. 7 depicts the HLPSL code for role UI.

The simulation via AVISPA is done using two widely-accepted back-end model checkers: The On-the-Fly Model-Checker (OFMC) and the Constraint Logic-based Attack Searcher (CL-AtSe).

Fig. 8 shows the CL-AtSe back-end checker report that assures that our protocol is SAFE and free from attack. Fig. 9 shows the OFMC back-end checker report, which proves that our protocol is SAFE and satisfies the specified security goals.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role role_UI (UI,RA,CRN:agent,TSK,Kur:symmetric_key,H:hash_func,Snd,Rcv:channel(dy))

played_by UI

def=
            local
            State : nat,
            N,Lc, IDSD, IDUI, IDCRN,INV1,INV2, PWI, TIDUri,S1,S2: text, SK1,SK2,TIDUI1,DIDCRN1, TIDUI2,DIDCRN2,XCCHH,XCLT,
X1CCHH,X1CLT:hash(text.text), UC: :hash(text.message), ,HF:hash(message.text.text),M:message,
          init State := 0

           transition
                 0.   State  = 0 /\ RCV(start) =|> State':= 2 /\ Snd({IDUI.PWI}_Kur) /\ secret(IDUI,idui,{RA,UI,CRN}) /\
secret(PWI,pwi,{RA,UI,CRN})

                 2.  State  = 2 /\ Rcv({TIDUri ' .TSK ' .S1 ' .S2 ' .INV1 ' .NV2 ' }_Kur) =|> State':= 4 /\ TIDUI1 ' :=H(IDUI.
INV1) /\ DIDCRN1 ' :=H(IDCRN. INV1) /\ SK1 ' :=H(TSK . INV1) /\ Lc' :=new() /\ UC ' :=H(Lc.xor(IDSD.S1) /\ M ' :=( DIDCRN1.
TIDUI1. UC) /\ HF ' :=H(M. SK1. INV1) /\ XCCHH ' :=H(HF. SK1) /\ XCLT ' :=H(Lc. INV2) /\ Snd(DIDCRN1.TIDUI1.UC.HF) /\
secret(SK1,sec_ SK1,{UI,CRN}) /\ witness(UI,CRN,ui_crn_ SK1, SK1) /\ secret(Lc,sec_Lc,{UI,CRN})
                 4.  State  = 4 /\ Rcv(TIDUI1 ' . DIDCRN1 ' .CU ' .HF ' ) =|> State':= 6 /\ SK2 ' :=H(SK1. XCCHH) /\  HF ' :=H(M.
SK2. XCCHH)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Figure 7: HLPSL code for the role played by the user UI

However, we were not able to use the TA4SP back-end model checker due to its limitation in supporting XOR operation, while The SATMC model checker has reported NOT SUPPORTED.

```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/finaltry-2.if

GOAL
 As Specified

BACKEND
 CL-AtSe

STATISTICS

 Analysed   : 5 states
 Reachable  : 3 states
 Translation: 0.01 seconds
 Computation: 0.00 seconds
```

Figure 8: CL-AtSe validation results

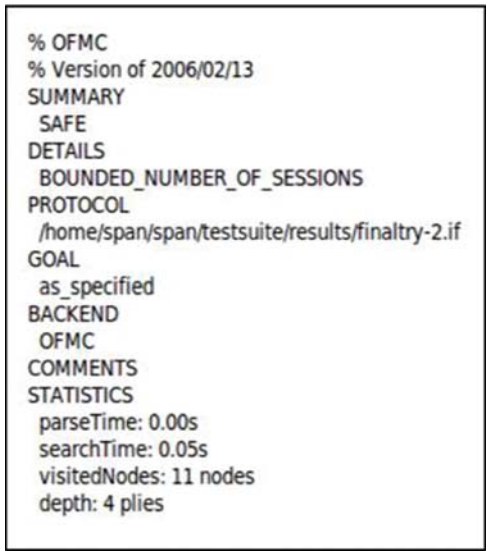

Figure 9: OFMC validation results

5 Performance Evaluation

In this section, we evaluate our protocol's performance in terms of communication overhead and computation costs.

5.1 Communication Overhead

The length of the parameters of the transmitted message DID_{CRN1}, TID_{Ui1}, UC, HF, CU are 128 bits, 128 bits, 256 bits, 160, and 256 bits, respectively.

In our proposed protocol, the transmitted messages Ui $\rightarrow$ CRN and CRN $\rightarrow$ Ui require $(128 + 128 + 256 + 160) = 681$ bits, $(128 + 128 + 256 + 160) = 681$ bits, respectively. The communication overheads of our scheme are shown in Tab. 2. Also, it can be noticed that our protocol requires only two messages for a successful mutual authentication between Ui and CRN.

Table 2: Communication cost of our protocol

Communication between nodes	Communication cost
Ui $\rightarrow$ CRN	681 bits
CRN $\rightarrow$ Ui	681 bits

5.2 Computation Cost

Our protocol is computationally lightweight designed for IoT smart building. Our protocol ensures high security using only a simple hash function and XOR computations; Hence consuming little computation overheads. However, this protocol's novelty is adding multiple security layers (e.g., GPS location tracker, CCHH and CLT technique, challenge/response mechanism, and ensuring transitory identities, it also provides relatively low computation cost.

Our protocol uses two operations as aforementioned, namely XOR operation and one-way hash function. Let T_h and T_{xor} be the computation times of one hash and one XOR

operations, respectively. Considering the authentication steps involved in our protocol and outlined in Fig. 3, the Ui performs 7 hash and 1 XOR operations, which yields a total computation cost of $12 \times T_h + 2 \times T_{xor}$. The computation time of XOR operation is very trivial and can be ignored, so we can assume $T_{xor} \approx 0$. On the other hand, the controller node CRN performs 10 hash and 2 XOR operations, which yield a total computation cost of $10 \times T_h + 2 \times T_{xor}$. Therefore, the total computation cost of N is $10 \times T_h + 2 \times T_{xor} \approx 10T_h$, while the computation cost of CRN corresponds to $10 \times T_h + 2 \times T_{xor} \approx 10 \times T_h$. The computation cost is summarized in Tab. 3.

Table 3: Computation cost of our protocol

Node computation	Cost
Ui	12 Th
CRN	10 Th

5.3 Comparisons with Recent Schemes

We present a comparison between our proposed scheme and other most related schemes in terms of communication cost based on transmissions in both directions between IoT node and gateway. We use the number of exchanged messages for a successful authentication as the key to the communication cost comparison. As presented in Tab. 4, our scheme requires 4 messages and 2304 bits total number of bits for successful mutual authentication. In general, the comparison shows that our scheme is comparatively more cost-efficient than the other related works in terms of the number of exchanged messages and the total number of bits, and just a little less cost efficient than that of Kumar et al. [6] because our scheme adds additional functionality and security features are not provided by Kumar et al.; such as mutual authentication between the user and smart device, mutual authentication between user and gateway, password guessing attack, password change attack, stolen smartphone/smart card attack, and password change phase, physical context awareness (i.e., location awareness), and historical authentication.

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 [7]	4	3232
Shuai et al [12]	4	2944
Kumar et al [6]	3	1696
Proposed scheme	2	1362

It can be observed that our protocol is comparatively more cost-efficient.

6 Conclusion

We proposed, in the current paper, a secure remote mobile device authentication protocol. The proposed protocol allows only legitimate users to authenticate with the IoT devices via the smart building getaway and exchange a symmetric session key for future secure communications.

The security evaluation of the protocol, both through informal analysis and formal model checking (using AVISPA toolkit), shows that our scheme is secure against known attack techniques.

Despite the encouraging results, more work remains to be done. In our future work, we will implement the proposed protocol using OMNET++ and conduct live security tests using penetration testing tools such as Kali Linux. We will also explore how to strengthen our framework's securit to thwart impersonation by adapting continuous authentication schemes, such as the approach proposed by Tsai et al. [19]. In their work [19], Tsai et al. introduced a passive continuous authentication system based on physiological and soft biometrics technologies, where face recognition is mainly used to control the authentication process In contrast, soft biometric is used to prevent and deal with any potential security breach, such as account hijacking.

Acknowledgement: The authors would like to thank University of Bisha for support him doing this research.

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The author declares that he has no conflicts of interest to report regarding the present study.

References

[1] S. Pandikumar and R. S. Vetrivel, "Internet of things based architecture of web and smart home interface using GSM," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, no. 3, pp. 1721–1727, 2014.

[2] T. Sanchez, D. C. Ranasinghe, M. Harrison and D. McFarlane, "Adding sense to the internet of things—an architecture framework for smart object systems," *Pers Ubiquitous Compuer*, vol. 16, no. 3, pp. 291–308, 2012.

[3] C. Kolias, G. Kambourakis, A. Stavrou and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[4] J. Jeong, M. Y. Chung and H. Choo, "Integrated OTP-based user authentication scheme using smart cards in home networks," in *Proc. of the 41st Annual Hawaii Int. Conf. on System Sciences*, Waikoloa, HI, USA, pp. 294, 2008.

[5] B. Vaidya, J. H. Park, 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, 2016.

[6] 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, 2016.

[7] M. Wazid, A. K. Das, V. Odelu, N. Kumar and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 391–406, 2020.

[8] M. Alshahrani and I. Traore, "Secure mutual authentication and automated access control for IoT smart home using cumulative keyed-hash chain," *Journal of Information Security and Applications*, vol. 45, no. 2, pp. 156–175, 2019.

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

[10] Y. Li, "Design of a key establishment protocol for smart home energy management system," in *2013 Fifth Int. Conf. on Computational Intelligence, Communication Systems and Networks*, Madrid, Spain, pp. 88–93, 2013.

[11] F. K. Santoso and N. C. H. Vun, "Securing IoT for smart home system," in *Int. Symp. on Consumer Electronics*, Madrid, Spain, pp. 1–2, 2015.

[12] M. Shuai, N. Yu, H. Wang and L. Xiong, "Anonymous authentication scheme for smart home environment with provable security," *Computers & Security*, vol. 86, no. 2, pp. 132–146, 2019.

[13] A. Pfitzmann and M. Köhntopp, "Anonymity, unobservability, and pseudonymity—a proposal for terminology," in *Designing Privacy Enhancing Technologies*, vol. 2009. Berlin, Heidelberg: Springer, pp. 1–9, 2001.

[14] K. McKay, L. Bassham, M. S. Turan and N. Mouha, "Report on lightweight cryptography," *National Institute of Standards and Technology*, vol. 8114. pp. 1–23, 2016.

[15] 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.

[16] M. Burrows, M. Abadi and R. M. Needham, "A logic of authentication," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 426, no. 1871, pp. 233–271, 1989.

[17] J. Wen, M. Zhang and X. Li, "The study on the application of BAN logic in formal analysis of authentication protocols," in *Proc. of the 7th Int. Conf. on Electronic Commerce*, New York, NY, USA, pp. 744–747, 2005.

[18] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna *et al.,* "The AVISPA tool for the automated validation of internet security protocols and applications," in *Int. Conf. on Computer Aided Verification*, Berlin, Heidelberg: Springer, pp. 281–285, 2005.

[19] P. W. Tsai, M. K. Khan, J. S. Pan and B. Y. Liao, "Interactive artificial bee colony supported passive continuous authentication system," *IEEE Systems Journal*, vol. 8, no. 2, pp. 395–405, 2014.