Date of publication DEC-31, 2022, date of current version NOV-23, 2022.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.21.4.2777

DMUAS-IoT: A Decentralised Multi-Factor User Authentication Scheme for IoT Systems

IKENNA RENE CHIADIGHIKAOBI, NORLIZA KATUK, BAHARUDIN OSMAN
School of Computing, Universiti Utara Malaysia Sintok, Kedah, Malaysia
chiadighikaobiikenna@yahoo.com, k.norliza@uum.edu.my, bahaosman@uum.edu.my

Corresponding author: Norliza Katuk (e-mail: k.norliza@uum.edu.my).

This work was funded by the Ministry of Education Malaysia under the Fundamental Research Grant Scheme (Ref: FRGS/1/2018/ICT03/UUM/02/1,UUM S/O Code: 14208).

ABSTRACT The Internet of Things (IoT) has become the fundamental infrastructure of many intelligent applications, such as smart homes. IoT applications store distributes various information, including user authentication information, over a public channel that exposes it to security threats and attacks. Therefore, this study intends to protect authentication data communication through a decentralised multi-factor user authentication scheme for secure IoT applications (DMUAS-IoT). The scheme is secure and enables efficient user registration, login and authentication, and the user profile updating process where legitimate users can access the IoT system resources. DMUAS-IoT adopted PRESENT for face image encryption and elliptic curve cryptography for data exchange. The scheme security was verified using ProVerif and AVISPA, and mutual authentication was checked with BAN-Logic. The results show that the scheme is secure against man-in-the-middle and impersonation attacks, provides mutual authentication and has a low computation cost. Hence, the outcomes of this study could help secure user authentication data from attacks on applications involved with IoT and resource constraint environments.

KEYWORDS Biometric authentication; cryptography; encryption; ECC; PRESENT; face image.

I. INTRODUCTION

The advancement in information technology and the Internet has catalysed the development of the Internet of Things (IoT) and its systems. Like other digital systems, user authentication is necessary for accessing and controlling various IoT systems, including smart homes. Most IoT systems comprise resource-limited devices and sensors, which challenges the type of encryption and protocol they adopt. IoT applications communicate within the network and external environments such as cloud servers where the entire system is accessible remotely using smartphones and desktop computers through an open channel such as Bluetooth, wireless network, and radio wave. Figure 1 visualises an example of an application of IoT in smart home systems. The sensors and devices in a smart home system are limited in communication power, computational power, and security [1]. Due to the impact of IoT, smart home systems are ranked as one of the fast-growing IoT applications in which they enable any device to communicate with each other and have the ability for seamless network establishment [2].

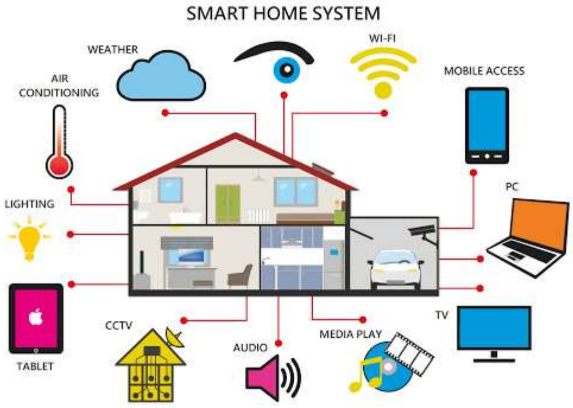


Figure 1. An example of a smart home system.

IoT device communication is provided through a combination of other technologies, like wireless sensor networks, Bluetooth, radio-frequency identification, cloud services, and machine-to-machine interfaces [3]. User authentication is an important aspect of IoT applications due to the transfer and data communication. The amount of data transmitted through the open channel of the IoT systems for communication makes it necessary for user authentication to combat attacks like impersonation and man-in-the-middle [4]. However, resource limitations of IoT devices have made preventing attacks one of the significant challenges, which has affected the type of security, encryption methods, and user authentication measures applied to this technology. Therefore, many researchers proposed a lightweight authentication scheme [5]. A lightweight authentication scheme embeds more suitable encryption algorithms to secure the user authentication information while maintaining low computation cost, communication cost, and memory usage.

Many researchers have designed different authentication schemes [1, 6–13] and methods which comprise different symmetric algorithms such as the Tiny Encryption Algorithm, Advanced Encryption Standard, PRESENT, and Hummingbird, to name a few [14, 15–17]. These studies proved that many of the implemented authentication schemes have a pitfall in computation cost and security.

A. RESEARCH CONTRIBUTION

The contributions of this study are:

1. The design of a decentralised multi-factor user authentication scheme for IoT systems called DMUAS-IoT. The scheme uses PRESENT encryption for face image template encryption.
2. A formal security analysis of DMUAS-IoT was conducted using ProVerif and the evidence of the scheme mutual authentication and correctness using BAN-logic. Further, the DMUAS-IoT scheme was tested using AVISPA.
3. The performance of DMUAS-IoT was determined through a comparison of the scheme and other similar authentication schemes.

B. RELATED WORKS

This study focuses on developing an authentication scheme for IoT systems that consists of communication operations executed between operational entities. An authentication scheme principal activity is to authorise the system functional entities and unauthorised illegitimate entities. Additionally, the scheme designed for authorising the IoT systems requires protection against replay, man-in-the-middle, and impersonation attacks, to mention a few. A good authentication scheme should be able to provide a mutual authentication that is a necessary feature for IoT systems [18-20]. Furthermore, the resource constraint in IoT devices made designing authentication schemes more challenging, as cryptographic protocol requires the utilisation of resources such as power, memory cost, and communication cost. Due to the device limitations, many IoT systems were implemented and designed with a one-hash function, which is weak to security, as the inversion is hard to realise [21–23].

An authentication scheme specifies the authentication method, the protocol, the process for user authentication, and algorithms verifying the users' identity to access resources from a computer system [24]. The authentication scheme is one of the critical security mechanisms, as it provides and protects user authentication information from being hacked or revealed [9, 25]. Studies on authentication schemes have been around for a long time and they address the needs of various types of computer systems. For example, a password is the most widely used and straightforward method for user authentication in a distributed computer system environment [26, 27]. However, the technique is vulnerable to attack and easily fooled [28]. Therefore, a standalone authentication method is insufficient to provide secure user authentication. Chang et al. [29] proposed an authentication scheme incorporating Hypertext Transfer Protocol cookies and public-key cryptography using the elliptic curve cryptography (ECC) algorithm for embedded devices and servers on the cloud. However, the scheme lacks mutual authentication [30].

Gope et al. [31] employed an RFID tag with a hash function for encryption. The scheme has potential performance; however, RFID tags for authenticating IoT devices expose it to cloning attacks [32]. Next, Wazid et al. [33] proposed a lightweight authentication scheme for a cloud-based IoT system that allows authenticated users to access the IoT sensor data remotely using a bitwise exclusive-OR (XOR) operations and one-way hash function. The performance analysis shows that the scheme offers low communication and computation overheads but a weak security measure. Finally, using ECC, Kumari et al. [34] enhanced the key exchange mechanism for authenticating devices in the IoT environment. It addressed offline password guessing and obtained a lightweight mutual authentication and session key agreement.

Jeong et al. [35, 23] applied a one-way hash function operation to a one-time password-based user authentication scheme using smart cards. The scheme has no mutual authentication of the gateway networks, smart devices, and users. Furthermore, the scheme does not protect against a stolen smart card attack and the possibility of extracting user authentication information from the smart card. Vaidya et al. [36] designed a password-based remote user authentication scheme using a hash-chaining and hashed one-time password. Like Jeong et al. [35], it does not have a mutual authentication of the user and gateway network. An analysis by Kim [37] was carried out on the [36] scheme, which identified the vulnerability of password guessing attacks and the loss of a smart card [37]. The scheme failed to provide security against privileged-insider, user impersonation, and password guessing attacks [37].

II. SYSTEMS MODEL

This section describes the network and threat models of DMUAS-IoT. Table 1 shows the description of the notation used in the scheme.

Table 1. The notation

Symbols	Description
U	The user of the system
IDU	The unique identity to identify the user U
SA	The system administrator that sets up the system
UD	The user U's device UD such as a smartphone and computer tablet with a device with an Internet browser
IDUD	The unique identity of user device UD
E	The email E the user U receives the email link EL
EL	A URL sent through user U's email
Fi	The facial image captured by the camera in the user device UD

IMf	The features of face image Fi of user U
Ft	The template derived from image features IMf
S	The Raspberry PI server is located within an IoT network that stores the face template Ft and faces image Fi
IDS	The unique identity of the PI Server S
RN	The random number RN used by secret key SK
SK	A randomly generated key by the PI server used for encrypting face template Ft
P(.)	The symmetric encryption algorithm for encrypting the face template Ft
EFt	An encrypted face template Ft using Present algorithm P(.) and secret key SK
R	The request or response is issued when an entity requests or responds to a resource or a process
CS	The cloud server is a private remote server that stores user authentication credentials, and data of an IoT network
IDCS	The unique identity of the cloud server CS
T	A timestamp is the time data generated by cloud server CS or PI server S embedded in a request or response R and authenticated email link Ael
Li	The Li is a variable that uses the hash function to combine IDU and Eft, then store it in the database D
WL	The weblink to access the authentication system
D	The database D, where user u information and authentication information are stored
$\|$	An operator used for combining two operations or variables
CC	The communication link between two entities
SK_{CSS}	The shared key between cloud server CS and PI server S
SK_{UDCS}	The shared key between the user device UD and cloud server CS
h(.)	The one-way hash function h(.) takes input information and restructures it to a fixed-size element
$\oplus$	The XOR is an operation that takes two operands and returns true if the data are different

A. NETWORK MODEL

DMUAS-IoT comprises six main entities: system administrator *SA*, cloud server *CS*, PI server *PI*, user *U*, user device *UD*, and IoT sensor *Sj*. This system is divided into fields consisting of *PI* and *Sj*. The cloud server enables the *U* to access the system; Figure 2 illustrates the network model. A user *U* needs to access the system requiring authentication and resource usage. In such a case, the user device *UD* and the *PI* need to authenticate each other mutually, and this requires a few mutual authentication steps (1) between *UD* and *CS*, (2) between *CS* and *PI*, and (3) between *UD* and *PI*. *UD* and *PI* establish a fresh session key for future secure communication. The communication between *UD* and *PI* is carried out via *CS,* which is accessible over the Internet.

The SA, the key player in the proposed model, creates U credentials using the U email and device ID. These credentials are considered secured and transmitted to the U over a secure channel. The created credentials by SA are stored in CS, and a section ID is created in PI for each U credential for user face image and email id identification. The U signs up into the system through a secure channel using a face image and the credential created by SA. The CS assigns the face image to the section ID created for the U, and the PI creates a face image template in which PRESENT encryption is applied and used for authentication. Figure 3 illustrates the DMUAS-IoT authentication framework.

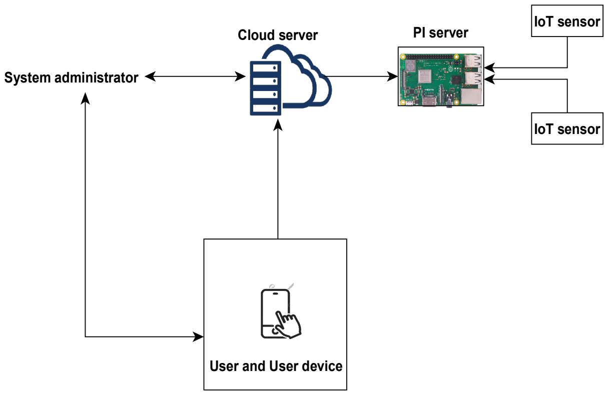


Figure 2. Network model of DMUAS-IoT.

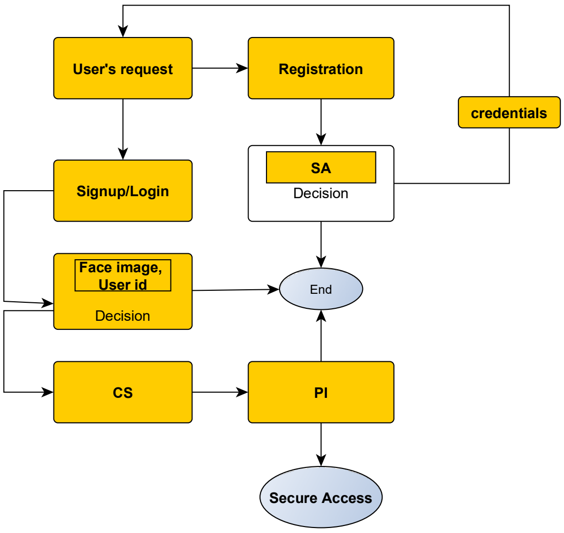


Figure 3. DMUAS-IoT authentication framework.

B. THREAT MODEL

DMUAS-IoT employed the Dovel-Yao threat model [38], assuming that PI and IoT sensors are not trusted as they operate in an insecure open communication channel. Based on the model, the adversary can eavesdrop and intercept the message and information communicated over the insecure channel to execute an active attack. Moreover, the adversary with physical access to the PI can retrieve sensitive information stored in them.

III. THE PROPOSED SCHEME

The centralised approach to user authentication protocol is a major challenge leading to security issues and computation costs in the IoT environment. Furthermore, the centralized control of the scheme makes the scheme vulnerable, with high computation costs despite the cryptographic algorithm adopted.

A. SETUP PHASE

In this phase, the system administrator SA establishes communication between cloud server *CS,* user device *UD*, and PI server *S*. Figure 4 illustrates the setup phase of this scheme.

Step S1: *SA* activates CS and S in a secure CC and chooses a secret key SK.

Step S2: SA assigns identity ID_X to (CS, S) in a secure CC and computes the secret key $SK_{CS\text{-}S}$ = h (ID_X || SK) for (CS, S).

Step S3: SA issues a communication signal to CS from S in a secure CC with a secret key SK, which randomly generates a random number RN, shared between CS and S. S stores <ID_X, $SK_{CS\text{-}S}$, RN>.

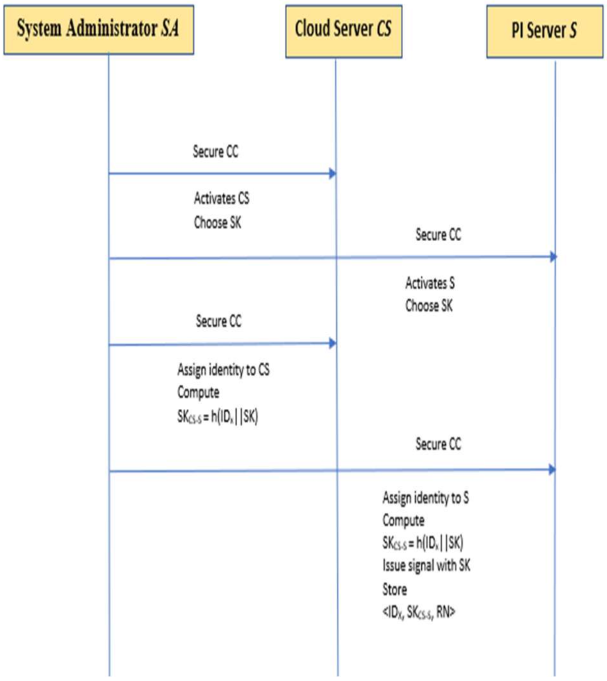


Figure 4. Setup phase.

B. REGISTRATION PHASE

This phase allows users to register once in the system and access the authentication system if an authorised user *U* initialises a registration request. Figure 5 illustrates the registration phase

Step R1: *SA* creates a user *U* profile with email and *IDUD* in a secure CC and stores it in *CS*, where *IDU*= h (email || IDUD).

Step R2: *U* enters the selected *IDU* on *UD,* and registration request R is sent to CS through secure communication channel CC, and CS checks, if entered IDU, is in database *D* and responds R with $K_{CS\text{-}UD}$ = h (IDUD ⊕ IDCS), if it is not in database, CS rejects the registration request.

Step R3: CS verifies IDU in the database, and EL sends to UD email E with timestamp T (3 minutes), U verifies within allowed T_{EL}, else reject if click EL > $T_{EL,}$ and registration is cancelled.

Step R4: *UD* captures *U* Fi (3 times) and the Fi is stored in S through CS. Then S computes Ft = IMf (Fi), EFt= P(Ft ⊕ SK), Li=h(IDU || EFt). Then S stores <Ft, EFt, Li>, and registration is successful.

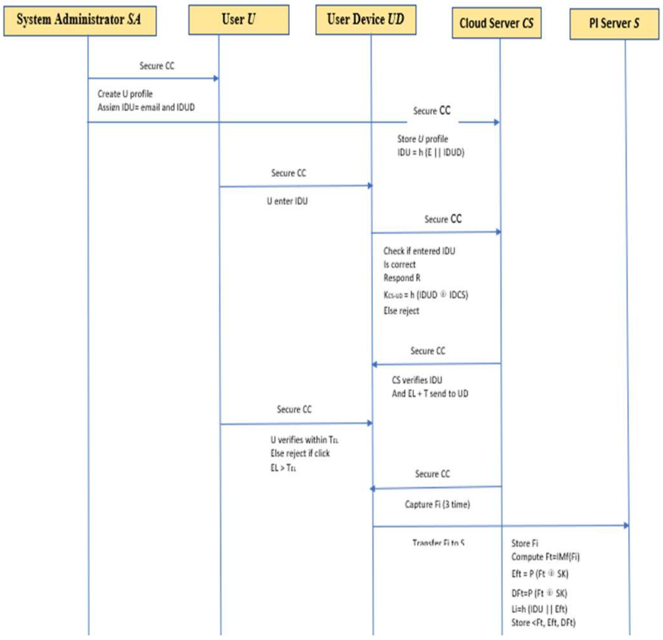


Figure 5. Registration phase.

C. AUTHENTICATION AND LOGIN PHASE

This phase is initiated when registered *U* needs access to the system. Figure 6 illustrates the authentication and login phase of the proposed scheme.

Step A1: The *U* initiates the authentication system by entering the web link WL in *UD*, which sends a request R to CS, R1=h (IDUD || R) ⊕ K_{UDCS}, and CS responds R to UD, R2 =h (IDCS || R) ⊕ K_{UDCS}. CS and UD established a connection and exchanged key K_{UDCS} through a secure CC, and CS requested IDU through the web link in UD.

Step A2: *U* enters the *IDU*, *CS* decrypts R to obtain < E, ID, SK> and verifies whether IDU matches the value stored. If the two values do not match, then CS rejects the request; otherwise, CS responds R and computes $SK_{S\text{-}UD}$ = h(IDUD ⊕IDCS). Then, UD captures U face image, and S computes Ft* = IMf (Fi), EFt* = P(Ft ⊕ SK), Li* = h(IDU || EFt) and checks whether Ft* = Ft, EFt* = EFt and Li* = Li. If EFt* ≠ EFt and Li* ≠ Li, CS denies access; otherwise, login process continues, S computes Li* = h (IDU || EFt), and if Li* = Li,

Step A3: The S generates a random number of RN and timestamp T, computes EL* = E ⊕ h (IDU || UD), T = h (Li* || RN) ⊕SK and CS sends EL*, T to U email E. U verifies EL* within allowed T, else reject, otherwise access is granted through a secure communication channel CC.

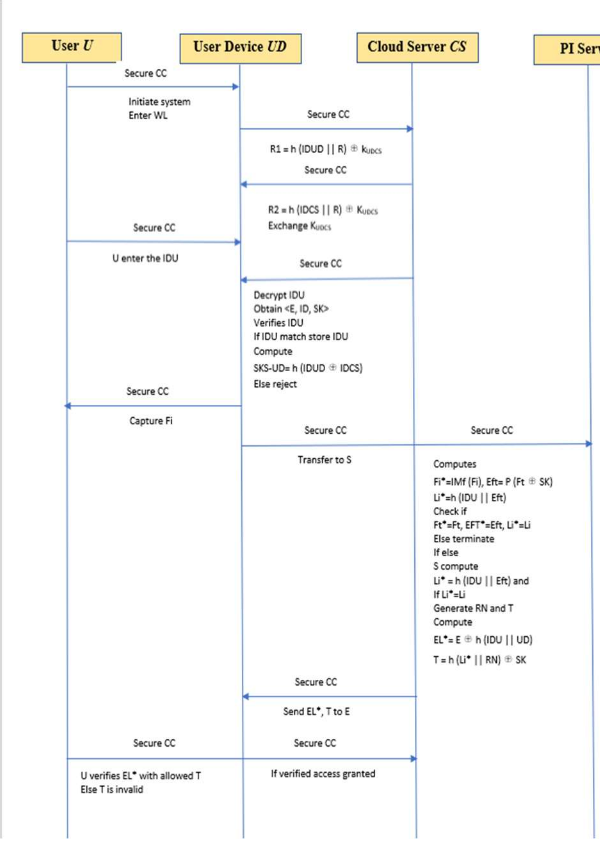


Figure 6. Authentication and Login Phase.

D. USER PROFILE RECOVERY

This phase occurs when registered U changes a new device and regains access to the existing profile through a secure CC. Figure 7 shows the user profile recovery steps for the proposed scheme.

Step P1: *U* sends request *R* to access the authentication system through a new UD1.

Step P2: *CS* detects request *R* from a new *UD1/unregistered/unauthenticated UD1*, and *CS* requests the *U* respond R by providing *IDU.*

Step P3: *CS* matches the provided *IDU* with the existing *IDU* in the database; if entered *IDU* = existing *IDU*, *CS* sends *EL* verification *R* with a *T*. Verification is rated valid if respond R email link verified time = *T*, invalid if time ≠ *T* and session terminates.

Step P4: *CS* receives the verified message, *CS* unregisters the existing *UD*, and *CS* sends communication request *R* through a secure CC to the new *UD1* and establishes a connection with $SK^1{}_{UDCS}$ assigning a new *IDUD1*.

Step P5: The new *UD1* computes $K^1{}_{UDCS} = H1(IDUD1 \oplus SK_{UDCS1})$, and verification between *CS* and new *UD1* communication is established.

Step P6: *CS* terminates the connection to new *UD1* and *CS* requests *U Fi1,* and if captured *U Fi1=* stored *Fi*, *S* sends a confirmation to *CS,* and *CS* issues an email authentication link with a *T*. Once verification occurs within the *T*, the system authenticates *U* if provided *Fi1* ≠ stored *Fi*, and email authentication link verification is not within the *T*, the system terminates the session.

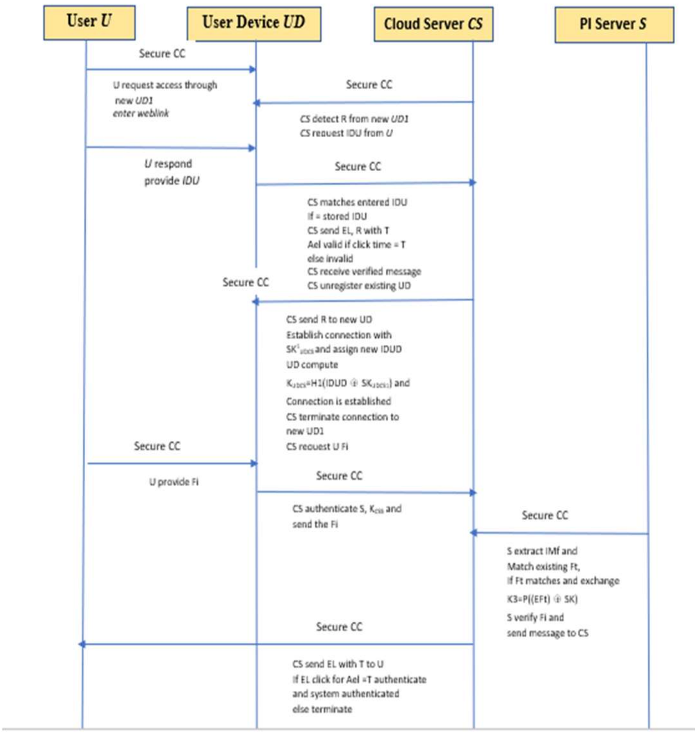


Figure 7. User profile recovery.

IV. SECURITY ANALYSIS

This study uses BAN logic, ProVerif, and AVISPA to evaluate the scheme. Further informal security analysis is carried out to ensure protection against security attacks.

A. BAN LOGIC

BAN logic [40, 9, 41, 42] is a method to verify the scheme achieved mutual authentication with the notations listed in Table 2. The BAN logic is used to check mutual authentication between UD and CS through S.

Table 2. BAN logic notations

Notation	Descriptions
A and B	Principals
$A \mid\equiv C$	Principal A believes the statement C
$A \blacktriangleleft C$	Principal A sees the statement C
$A \mid\Rightarrow C$	Principal A has jurisdiction over the statement C
$A \mid\sim C$	Principal A once said statement C
(E, F)	The statement E or F is one part of a message (E, F)
$< C >_F$	Statement E is encrypted with the key K
$(E)_k$	Statement E is hashed with the key K
$A \overset{k}{\leftrightarrow} B$	K is a secret parameter shared (or to be shared) between A and B
A B	C is a secret known only to A and B, and possibly to parties trusted by them
# (C)	The message C is fresh

The BAN logic rules for proving mutual authentication in DMUAS-IoT include:

i. Message-meaning rule:

If A believes that the key K is shared with B and A sees C encrypted under K, then A believes that b once said C.

$$\frac{P| \equiv B \overset{k}{\leftrightarrow} A, A \blacktriangleleft (E)}{A| \equiv B| \sim E}.$$

ii. Nonce verification rule:

If A believes C is fresh and A believes B once said C, then A believes B believes C.

$$\frac{A\,| \equiv \#(E), P| \equiv B|{\sim}E}{A| \equiv B| \equiv E}.$$

iii. Jurisdiction rule:

If A believes B has jurisdiction over E and A believes B believes E, then A believes E.

$$\frac{A\,| \equiv B| \Rightarrow E, A| \equiv B| \equiv E}{A| \equiv E}.$$

iv. Freshness conjuncatenation rule:

If one part of a statement is fresh, then the entire statement must also be fresh; so, if A believes E is fresh, then A believes E and F are fresh

$$\frac{A| \equiv \#(E)}{A| \equiv \#(E,F)}.$$

v. Belief rule:

If A believes X and F, then A believes B.

$$\frac{A| \equiv (E,F)}{A| \equiv E}.$$

Based on the BAN logic principle, the proposed authentication scheme must achieve the following goals:

a. Goal 1: $CS| \equiv UD| \equiv UD \overset{SK}{\leftrightarrow} CS$

b. Goal 2: $CS \equiv UD \overset{SK}{\leftrightarrow} CS$

c. Goal 3: $CS \equiv S| \equiv S \overset{SK}{\leftrightarrow} CS$

d. Goal 4: $CS| \equiv S \overset{SK}{\leftrightarrow} CS$

e. Goal 5: $UD| \equiv S| \equiv S \overset{SK}{\leftrightarrow} UD$

f. Goal 6: $UD| \equiv S \overset{SK}{\leftrightarrow} UD$

The fundamental assumptions of the proposed scheme are as follows:

i. A1: CS believes UDID is a secured shared parameter between UD and CS, $CS| \equiv (UD \xleftrightarrow{UDID} CS)$

ii. A2: CS believes T is fresh, $S| \equiv \#(T)$

iii. A3: S believes CS believes UDID is a secured shared parameter between S and CS. $(S| \equiv CS| \equiv (CS \xleftrightarrow{SKCSS} S))$

iv. A4: S believes T is fresh. $CS| \equiv \#(T)$

v. A5: UD believes S believes CS is a secured shared parameter between UD and S. $UD| \equiv S| \equiv (S \overset{SK}{\leftrightarrow} UD)$

The proposed authentication scheme presents messages transferred in the authentication protocol:

a. M1: $UD \rightarrow CS\colon \left(UD \xleftrightarrow{SkUDCS} CS, RN, E, Fi\right) UD \overset{SK}{\leftrightarrow} CS$

b. M2: $CS \rightarrow S\colon (EL||SKcss||SID)CS \xleftrightarrow{SKCSS} S$

c. M3: $S \rightarrow UD\big(T\big||UDID\big||Ft\big)S \xleftrightarrow{SKSUD} UD$

Analysis of our authentication scheme: We analyse the proposed authentication scheme to prove that the scheme achieves mutual authentication between UD, CS, and S.

a. S1: According to M1 and A1, and by applying the message meaning rule, we get:

b.

$$\frac{CS| \equiv \left(UD \xleftrightarrow{SKUDCS} CS\right), CS \blacktriangleleft \left(UD \xleftrightarrow{SKUDCS} CS, RN, E, Fi\right) \\ UD \overset{SK}{\leftrightarrow} CS}{CS| \equiv UD{\sim}(UD \xleftrightarrow{SLUDCS} CS, RN, E, Fi)}.$$

c. S2: From assumption A2 and by applying the freshness rule, we get:

$$\frac{CS| \equiv \#(T)}{CS||\#(UD \xleftrightarrow{SKUDCS} CS, RN, E, Fi)}.$$

d. S3: From derivations S1 and S2 and by applying the nonce verification rule, we get:

$$\frac{CS| \equiv \left(UD \xleftrightarrow{SKUDCS} CS, RN, E, Fi\right), CS| \equiv UD|{\sim} \\ (UD \xleftrightarrow{SKUDCS} CS, RN, E, Fi)}{CS| \equiv UD \equiv (UD \xleftrightarrow{SKUDCS} CS, RN, E, Fi)}.$$

e. S4: From derivation S3 and by applying the belief rule, we get Goal 1:

$$\frac{CS| \equiv UD \equiv (UD \xleftrightarrow{SKUDCS} CS, RN, E, Fi)}{CS| \equiv UD \equiv (UD \xleftrightarrow{SKUDCS} CS)}.$$

f. S5: From S4, A2 jurisdiction rule, we get Goal 2:

$$\frac{CS| \equiv UD \Rightarrow \left(UD \overset{SK}{\leftrightarrow} CS\right), CS| \equiv UD \equiv (UD \xleftrightarrow{SKUDCS} CS)}{CS| \equiv UD \overset{SK}{\leftrightarrow} CS}.$$

g. S6: From assumptions A3 and M2 and by applying the message meaning rule, we get:

$$\frac{S| \equiv \left(CS \xleftrightarrow{SKcss} S\right), S \Leftarrow (EL, SKcss, SID)CS \xleftrightarrow{SKcss} S}{S| \equiv CS|{\sim}(EL, SK, SID)CS \xleftrightarrow{SKcss} S}.$$

h. S7: From assumption A4 and by applying the freshness rule, we get:

$$\frac{S| \equiv \#(T)}{S| \equiv (E, Fi, RN, S \overset{SK}{\leftrightarrow} CS}.$$

i. S8: From derivations S6 and S7, by applying the nonce verification rule, we get:

$$\frac{S| \equiv (E, Fi, RN, S \overset{SK}{\leftrightarrow} CS, S \overset{SKcss}{\longleftrightarrow} CS, S|\equiv CS|\sim \left(EL, SK, SID, CS \overset{SKcss}{\longleftrightarrow} S\right))}{S|\equiv CS| \equiv (EL, SK, SID), CS \overset{SKcss}{\longleftrightarrow} S}.$$

j. S9: From derivation S8 and by applying the belief rule, we get Goal 3:

$$\frac{S| \equiv CS| \equiv (EL, SK, SID, CS \overset{SKcss}{\longleftrightarrow} S)}{CS| \equiv S| \equiv S \overset{SK}{\leftrightarrow} CS}.$$

k. S10: From assumptions A4 and S9, by applying the jurisdiction rule, we get Goal 4:

$$\frac{S| \equiv CS \Rightarrow \left(CS \overset{Skcss}{\longleftrightarrow} S\right), CS| \equiv S \equiv (S \overset{SKcss}{\longleftrightarrow} CS)}{CS| \equiv S \overset{SK}{\leftrightarrow} CS}$$

l. S11: From assumption A5 and message M3 and by applying the message meaning rule, we get:

$$\frac{UD| \equiv \left(S \overset{SK}{\leftrightarrow} UD\right), UD \blacktriangleleft (T, UDID, CSID, Ft)S \overset{SK}{\leftrightarrow} UD}{UD \equiv S|\sim(T, UDID, CSID, Ft)S \overset{SK}{\leftrightarrow} UD}.$$

m. S12: From assumption A6 and by applying the freshness rule, we get:

$$\frac{UD \equiv \#(T)}{UD| \equiv (T, Ft, R, CSID, S \overset{SK}{\leftrightarrow} UD)}.$$

n. S13: From derivation S11 and S12, by applying the nonce verification rule, we get:

$$\frac{UD \equiv (T, Ft, RN, CSID, S \overset{SK}{\leftrightarrow} UD, S \overset{SK}{\leftrightarrow} UD, S|\sim \left(T, UDID, CSID, Ft, S \overset{Sk}{\leftrightarrow} UD\right))}{UD| \equiv S| \equiv (T, UDID, CSID, Ft)S \overset{SK}{\leftrightarrow} UD}.$$

o. S14: From derivation S13 and by applying the belief rule, we get Goal 5:

$$\frac{UD| \equiv S| \equiv (T, UDID, CSID, S \overset{SK}{\leftrightarrow} UD)}{UD| \equiv S| \equiv S \overset{SK}{\leftrightarrow} UD}.$$

p. S15: From S14, A5, and jurisdiction rule, we get Goal 6:

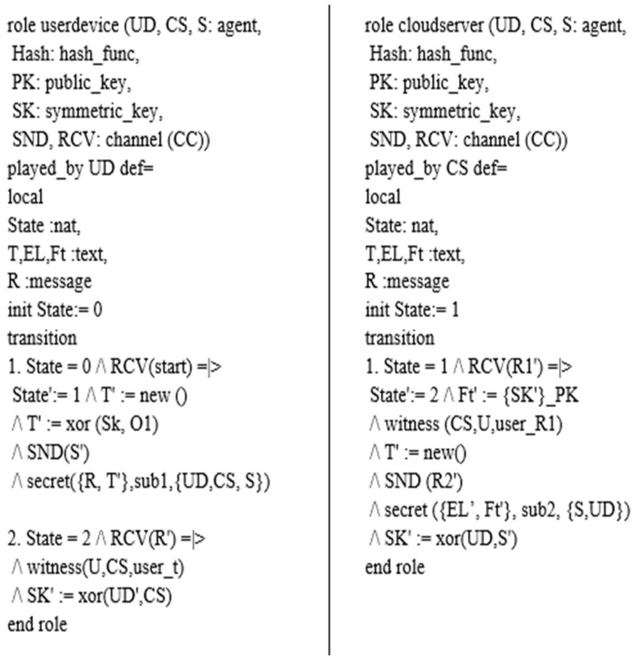
$$\frac{UD| \equiv S \Rightarrow \left(S \overset{SK}{\leftrightarrow} UD\right), UD| \equiv S \equiv (S \overset{SK}{\leftrightarrow} UD)}{UD| \equiv S \overset{SK}{\leftrightarrow} UD}.$$

The analysis proved that the DMUAS-IoT has mutual authentication.

B. SIMULATION USING THE AVISPA

The initial process to validate using AVISPA was to script the proposed scheme into HLPSL language, which began with the declaration of the basic roles (agents, cryptography operations, channel (Doblev-Yao). Then, it declares the processes that the agent executes, the composition roles that declare the legitimate entities that participated in the communication, the environment role declares the global entities, intruder knowledge, and all session that exists during the communication. Finally, the mutual authentication and key exchange were examined on AVISPA to validate and assess the strength of the proposed protocol. Figure 8 and Figure 9 show the AVISPA proposed protocol analysis and validation script, which lists the specification and role of the user device, cloud server, PI server, session, and environment goal of the protocol. Finally, the result of the protocol analysis using AVISPA is presented in Figure 10, which shows that the scheme is safe.

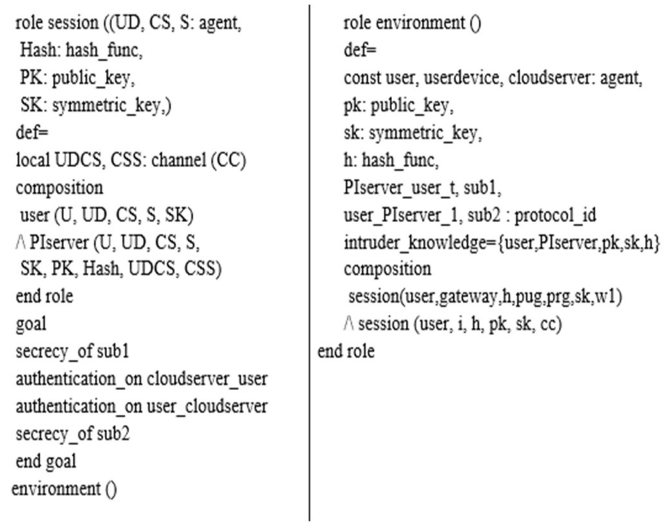
The basic role of the user device UD, cloud server CS, and PI server S comprises local agents (UD, CS, S), hash function operation, keys operations (SK, PK, etc.), and details of communication channels (CC). In addition, it described the messages Request or Response R used and exchanged during communication. UD gets activated in State = 0 and generates a timestamp T in State:=1 to validate the communication to CS and S. CS and S receive the request R in State = 1 and initiate the process of State':=2. The CS and S performed the decryption task of the message request R to avoid replay attacks. After successful decryption and validation of the message R, CS and S compute T'(=xor(SK, PK)) and T'(=xor(UD, SK). A fresh timestamp T is generated by CS and UD validate, as the goal is to have the privacy of the data which are communicated between the UD and S through CS. This results in a mutual authentication between UD, CS, and S.



```
role userdevice (UD, CS, S: agent,
 Hash: hash_func,
 PK: public_key,
 SK: symmetric_key,
 SND, RCV: channel (CC))
played_by UD def=
local
State :nat,
T,EL,Ft :text,
R :message
init State:= 0
transition
1. State = 0 ∧ RCV(start) =|>
 State':= 1 ∧ T' := new ()
 ∧ T' := xor (Sk, O1)
 ∧ SND(S')
 ∧ secret({R, T'},sub1,{UD,CS, S})

2. State = 2 ∧ RCV(R') =|>
 ∧ witness(U,CS,user_t)
 ∧ SK' := xor(UD',CS)
end role
```

```
role cloudserver (UD, CS, S: agent,
 Hash: hash_func,
 PK: public_key,
 SK: symmetric_key,
 SND, RCV: channel (CC))
played_by CS def=
local
State: nat,
T,EL,Ft :text,
R :message
init State:= 1
transition
1. State = 1 ∧ RCV(R1') =|>
 State':= 2 ∧ Ft' := {SK'}_PK
 ∧ witness (CS,U,user_R1)
 ∧ T' := new()
 ∧ SND (R2')
 ∧ secret ({EL', Ft'}, sub2, {S,UD})
 ∧ SK' := xor(UD,S')
end role
```

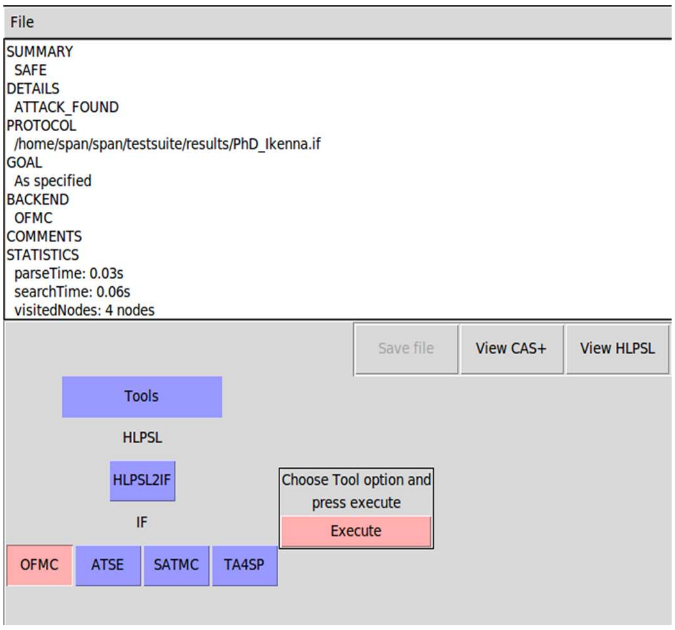
Figure 8. User device, cloud server, and PI server role specification for the proposed scheme.



```
role session ((UD, CS, S: agent,
 Hash: hash_func,
 PK: public_key,
 SK: symmetric_key,)
def=
local UDCS, CSS: channel (CC)
composition
 user (U, UD, CS, S, SK)
/\ PIserver (U, UD, CS, S,
 SK, PK, Hash, UDCS, CSS)
end role
goal
secrecy_of sub1
authentication_on cloudserver_user
authentication_on user_cloudserver
secrecy_of sub2
end goal
environment ()
```

```
role environment ()
def=
const user, userdevice, cloudserver: agent,
pk: public_key,
sk: symmetric_key,
h: hash_func,
PIserver_user_t, sub1,
user_PIserver_1, sub2 : protocol_id
intruder_knowledge={user,PIserver,pk,sk,h}
composition
 session(user,gateway,h,pug,prg,sk,w1)
/\ session (user, i, h, pk, sk, cc)
end role
```

Figure 9. The session, environment, and goal specification of the proposed protocol.



```
SUMMARY
  SAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/span/span/testsuite/results/PhD_Ikenna.if
GOAL
  As specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.03s
  searchTime: 0.06s
  visitedNodes: 4 nodes
```

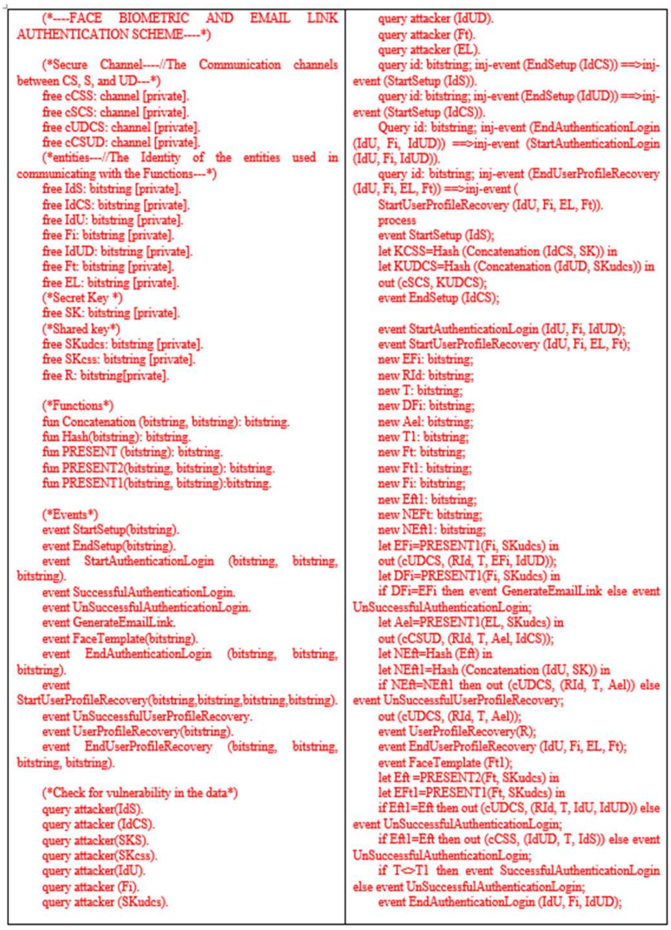
Figure 10. The AVISPA result of DMUAS-IoT.

The analysis and validation of the proposed scheme using AVISPA shows that the protocol is secured and marked as safe.

C. FORMAL VERIFICATION WITH PROVERIF

The proposed scheme uses a private channel (ChSec:). The secure channel was established between the user, user device, and cloud server for the login and authentication phase.

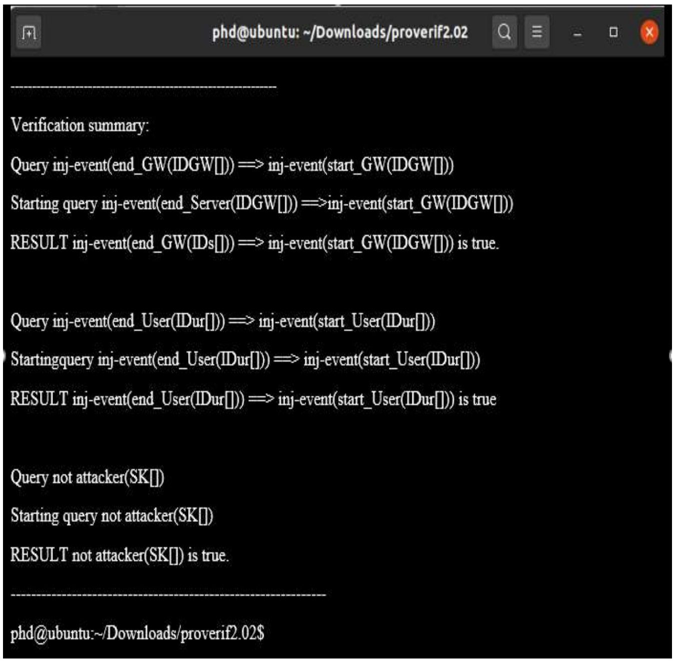
Based on the verification process in Figure 11, the result in Figure 12 was achieved, which shows that all four processes were successfully started and ended. It also shows that the attacker was unable to get the session key. Therefore, the DMUAS-IoT verification protocol was successfully executed, and the results show that it meets the security requirement.



```
    (*----FACE BIOMETRIC AND EMAIL LINK
AUTHENTICATION SCHEME----*)

    (*Secure Channel----//The Communication channels
between CS, S, and UD---*)
    free cCSS: channel [private].
    free cSCS: channel [private].
    free cUDCS: channel [private].
    free cCSUD: channel [private].
    (*entities---//The Identity of the entities used in
communicating with the Functions---*)
    free IdS: bitstring [private].
    free IdCS: bitstring [private].
    free IdU: bitstring [private].
    free Fi: bitstring [private].
    free IdUD: bitstring [private].
    free Ft: bitstring [private].
    free EL: bitstring [private].
    (*Secret Key *)
    free SK: bitstring [private].
    (*Shared key*)
    free SKudcs: bitstring [private].
    free SKcss: bitstring [private].
    free R: bitstring[private].

    (*Functions*)
    fun Concatenation (bitstring, bitstring): bitstring.
    fun Hash(bitstring): bitstring.
    fun PRESENT (bitstring): bitstring.
    fun PRESENT2(bitstring, bitstring): bitstring.
    fun PRESENT1(bitstring, bitstring):bitstring.

    (*Events*)
    event StartSetup(bitstring).
    event EndSetup(bitstring).
    event StartAuthenticationLogin (bitstring, bitstring,
bitstring).
    event SuccessfulAuthenticationLogin.
    event UnSuccessfulAuthenticationLogin.
    event GenerateEmailLink.
    event FaceTemplate(bitstring).
    event EndAuthenticationLogin (bitstring, bitstring,
bitstring).
    event
StartUserProfileRecovery(bitstring,bitstring,bitstring,bitstring).
    event UnSuccessfulUserProfileRecovery.
    event UserProfileRecovery(bitstring).
    event EndUserProfileRecovery (bitstring, bitstring,
bitstring, bitstring).

    (*Check for vulnerability in the data*)
    query attacker(IdS).
    query attacker (IdCS).
    query attacker(SKS).
    query attacker(SKcss).
    query attacker(IdU).
    query attacker (Fi).
    query attacker (SKudcs).
```

```
    query attacker (IdUD).
    query attacker (Ft).
    query attacker (EL).
    query id: bitstring; inj-event (EndSetup (IdCS)) ==>inj-
event (StartSetup (IdS)).
    query id: bitstring; inj-event (EndSetup (IdUD)) ==>inj-
event (StartSetup (IdCS)).
    Query id: bitstring; inj-event (EndAuthenticationLogin
(IdU, Fi, IdUD)) ==>inj-event (StartAuthenticationLogin
(IdU, Fi, IdUD)).
    query id: bitstring; inj-event (EndUserProfileRecovery
(IdU, Fi, EL, Ft)) ==>inj-event (
    StartUserProfileRecovery (IdU, Fi, EL, Ft)).
    process
    event StartSetup (IdS);
    let KCSS=Hash (Concatenation (IdCS, SK)) in
    let KUDCS=Hash (Concatenation (IdUD, SKudcs)) in
    out (cSCS, KUDCS);
    event EndSetup (IdCS);

    event StartAuthenticationLogin (IdU, Fi, IdUD);
    event StartUserProfileRecovery (IdU, Fi, EL, Ft);
    new EFi: bitstring;
    new RId: bitstring;
    new T: bitstring;
    new DFi: bitstring;
    new Ael: bitstring;
    new T1: bitstring;
    new Ft: bitstring;
    new Ft1: bitstring;
    new Fi: bitstring;
    new Eft1: bitstring;
    new NEFt: bitstring;
    new NEft1: bitstring;
    let EFi=PRESENT1(Fi, SKudcs) in
    out (cUDCS, (RId, T, EFi, IdUD));
    let DFi=PRESENT1(Fi, SKudcs) in
    if DFi=EFi then event GenerateEmailLink else event
UnSuccessfulAuthenticationLogin;
    let Ael=PRESENT1(EL, SKudcs) in
    out (cCSUD, (RId, T, Ael, IdCS));
    let NEft=Hash (Eft) in
    let NEft1=Hash (Concatenation (IdU, SK)) in
    if NEft=NEft1 then out (cUDCS, (RId, T, Ael)) else
event UnSuccessfulUserProfileRecovery;
    out (cUDCS, (RId, T, Ael));
    event UserProfileRecovery(R);
    event EndUserProfileRecovery (IdU, Fi, EL, Ft);
    event FaceTemplate (Ft1);
    let Eft =PRESENT2(Ft, SKudcs) in
    let EFt1=PRESENT1(Ft, SKudcs) in
    if Eft1=Eft then out (cUDCS, (RId, T, IdU, IdUD)) else
event UnSuccessfulAuthenticationLogin;
    if Eft1=Eft then out (cCSS, (IdUD, T, IdS)) else event
UnSuccessfulAuthenticationLogin;
    if T<>T1 then event SuccessfulAuthenticationLogin
else event UnSuccessfulAuthenticationLogin;
    event EndAuthenticationLogin (IdU, Fi, IdUD);
```

Figure 11. ProVerif verification.



```
-----------------------------------------------------------

Verification summary:

Query inj-event(end_GW(IDGW[])) ==> inj-event(start_GW(IDGW[]))

Starting query inj-event(end_Server(IDGW[])) ==>inj-event(start_GW(IDGW[]))

RESULT inj-event(end_GW(IDs[])) ==> inj-event(start_GW(IDGW[])) is true.


Query inj-event(end_User(IDur[])) ==> inj-event(start_User(IDur[]))

Startingquery inj-event(end_User(IDur[])) ==> inj-event(start_User(IDur[]))

RESULT inj-event(end_User(IDur[])) ==> inj-event(start_User(IDur[])) is true


Query not attacker(SK[])

Starting query not attacker(SK[])

RESULT not attacker(SK[]) is true.

-----------------------------------------------------------

phd@ubuntu:~/Downloads/proverif2.02$
```

Figure 12. ProVerif verification result of DMUAS-IoT

V. INFORMAL VERIFICATION

An informal analysis was done to prove that DMUAS-IoT is secure against man-in-the-middle and impersonation attacks.

A. REPLAY AND MITM ATTACKS

Each request transmitted during the authentication and login in the scheme contains a timestamp, random number, and response action. The timestamp T and random number RN are XORed with a secret key SK and face image and verified using an email link. An adversary cannot forge these message hash values; therefore, the proposed protocol is secure against man-in-the-middle and replay attacks.

B. IMPERSONATION ATTACK

An attacker may impersonate User U, send the authentication request R to CS, and obtain IDUD if the attacker has physical access to the U device; the first point of attack fails. If the attacker has access to the device and requests authentication to CS, compute SKS-UD = h(IDUD XOR IDCS) and request face image capture. The attacker fails to have the exact face image match. Therefore, DMUAS-IoT is protected from impersonation attacks.

VI. PERFORMANCE COMPARISON

The performance of DMUAS-IoT was compared with other schemes such as [33, 36, 40] in terms of six attacks. They are impersonation (A1), a man-in-the-middle (A2), password guessing (A3), insider (A4), session key discloser (A5), and replay (A6). The schemes were also compared for session key agreement (A7) and mutual authentication (A8). The scheme computation cost – the executed number of operations to accomplish the authentication process is also analysed in this study. Concatenation and XOR were excluded in the calculation as the operations use little execution time [44]. The computation cost, the operations, function execution, and the number of bits are counted [44, 45], and the result is shown in Table 4. A detailed description of the notation is shown in Table 3. This study excluded registration and password change phases as they were rarely used. Table 4 demonstrates that DMUAS-IoT has a lower computation cost than other schemes. Table 5 shows that DMUAS-IoT is also secure against the six attacks.

Table 3. Definition and conversion of operations

Notation	Definition and conversion
CC	Computation cost
TH	The computation cost of a single hash function
TSE	The computation cost of symmetric encryption
TSD	The computation cost of symmetric decryption
TECM	The computation cost of the ECC point multiplication operation.

Table 4. Computation cost comparison

Studies	Number of operations	Cost (Bits)
[43]	7Th+8Tecm	1760
[36]	40Th+6Ted	1760
[39]	30Th+1Tecm	1696
DMUAS-IoT	7Th+4Ted+2Tecm	1460

Table 5. Comparison of security features

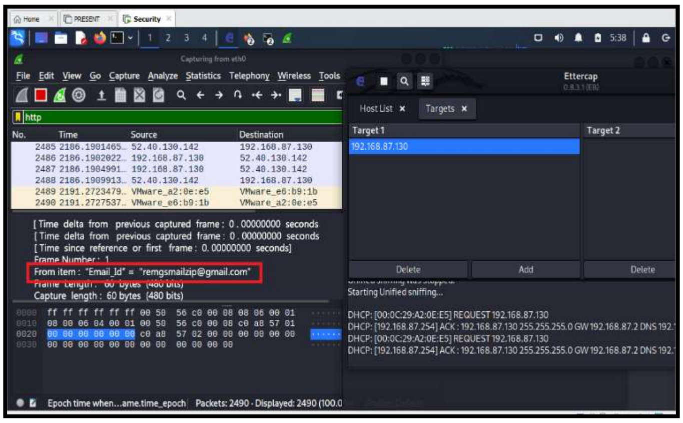
Studies	A1	A2	A3	A4	A5	A6	A7	A8
[43]	×	×	×	√	×	×	√	√
[36]	×	×	×	×	×	×	√	√
[39]	√	×	×	√	√	×	√	√
DMUAS-IoT	√	√	√	√	√	√	√	√

VII. REAL-LIFE EVALUATION

The proposed scheme was evaluated in a real-life environment to measure security performance. Kali Linux, Wireshark, and Ettercap tools were installed and configured on a computer device to perform this evaluation. First, the tools were used to build the real-life security evaluation environment for the implemented authentication scheme system. Then, the security evaluation was performed using the tools (Kali Linux, Wireshark, and Ettercap) to recover the user details. After the installation and configuration of Wireshark and Ettercap in the Kali Linux operating system, the authentication scheme (www.faceauthentication.tech) was accessed on the Kali Linux web browser.

The Wireshark was initiated during the scheme's access to capture the scheme traffic, communication, and message information. Based on the data captured by Wireshark, the Ettercap is initiated and issued a man-in-the-middle, impersonation, and database injection attacks. The Ettercap sniffed on all open channels, and Wireshark captured all information Ettercap discovered during the sniffing process. The security evaluation process on the authentication scheme detected and recovered a user email ID, but with other user information unknown to the attacker. The attacker failed to detect and recover the user device ID, face template, and encryption keys. Based on the achieved result, a conclusion can be drawn that the authentication system is secured against attacks. However, the email ID was recovered, and an attacker cannot use it to access the system, nor can the attacker modify the authentication access due to different layers (email link authentication and face recognition) of security. Furthermore, the scheme implementation approach in a decentralised manner makes it difficult for an attacker to gain access to the system as:

a. The access to the system administrator is not in an open channel.
b. The user face image, face template, and private key are stored in the PI server, and the authentication occurs in the PI server.
c. The system operates a decentralised method as each module manages its resources.



```
No.   Time                Source              Destination
 2485 2186.1901465…  52.40.130.142       192.168.87.130
 2486 2186.1902022…  192.168.87.130      52.40.130.142
 2487 2186.1904991…  192.168.87.130      52.40.130.142
 2488 2186.1909913…  52.40.130.142       192.168.87.130
 2489 2191.2723479…  VMware_a2:0e:e5     VMware_e6:b9:1b
 2490 2191.2727537…  VMware_e6:b9:1b     VMware_a2:0e:e5

[Time delta from previous captured frame: 0.00000000 seconds
[Time delta from previous captured frame: 0.00000000 seconds
[Time since reference or first frame: 0.00000000 seconds]
Frame Number: 1
From item: "Email_Id" = "remgsmailzip@gmail.com"
Frame Length: 60 bytes (480 bits)
Capture length: 60 bytes (480 bits)

0000  ff ff ff ff ff ff 00 50  56 c0 00 08 08 06 00 01
0010  08 00 06 04 00 01 00 50  56 c0 00 08 c0 a8 57 01
0020  00 00 00 00 00 00 c0 a8  57 02 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  00 00 00 00

Epoch time when...ame.time_epoch   Packets: 2490 · Displayed: 2490 (100.0
```

```
Target 1
192.168.87.130

Starting Unified sniffing...

DHCP: [00:0C:29:A2:0E:E5] REQUEST 192.168.87.130
DHCP: [192.168.87.254] ACK : 192.168.87.130 255.255.255.0 GW 192.168.87.2 DNS 192.
DHCP: [00:0C:29:A2:0E:E5] REQUEST 192.168.87.130
DHCP: [192.168.87.254] ACK : 192.168.87.130 255.255.255.0 GW 192.168.87.2 DNS 192.
```

Figure 13. Real-life security evaluation.

VIII. CONCLUSION

This study proposes a decentralised multi-factor user authentication scheme in IoT systems named DMUAS-IoT. The scheme consists of setup, registration, authentication, and login, and user profile recovery phases. The scheme protects authentication information communicated within IoT systems like smart homes. The authentication mechanism employs a face image encrypted using the PRESENT algorithm and ECC for key exchange. Furthermore, the email link is used to increase additional security and avoid attacks like impersonation, replay and man-in-the-middle attacks. DMUAS-IoT has been verified using BAN Logic, Avispa and ProVerif. The proposed scheme will be programmed and tested on the actual IoT systems in future work.

References

[1] S. Naoui, M. E. Elhdhili, and L. A. Saidane, "Lightweight and secure password based smart home authentication protocol: LSP-SHAP," *Journal of Network and Systems Management*, vol. 27, no. 4, pp. 1020–1042, 2019. https://doi.org/10.1007/s10922-019-09496-x.

[2] B. Ali, Int*ernet of Things based Smart Homes: Security Risk Assessment and Recommendations*, Master Thesis, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå Sweden, 2016. [Online]. Available at: https://www.diva-portal.org/smash/get/diva2:1032194/FULLTEXT02.pdf

[3] N. Katuk, K. R. Ku-Mahamud, N. H. Zakaria, and M. A. Maarof, "Implementation and recent progress in cloud-based smart home automation systems," *Proceedings of the 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE),* November 2019, pp. 71–77, 2018. https://doi.org/10.1109/ISCAIE.2018.8405447.

[4] R. Almadhoun, M. Kadadha, M. Alhemeiri, M. Alshehhi, and K. Salah, "A User Authentication Scheme of IoT Devices using Blockchain-Enabled Fog Nodes," *Proceedings of the 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, 2018, pp. 1-8. https://doi.org/10.1109/AICCSA.2018.8612856.

[5] H. Lin and N. W. Bergmann, "IoT privacy and security challenges for smart home environments," *Information*, vol. 7, no. 3, pp. 44, 2016. https://doi.org/10.3390/info7030044.

[6] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018. https://doi.org/10.1016/j.future.2016.12.028.

[7] K. Mahmood, S. Ashraf Chaudhry, H. Naqvi, T. Shon, and H. Farooq Ahmad, "A lightweight message authentication scheme for smart grid communications in power sector," *Computers & Electrical Engineering*, vol. 52, no. May, pp. 114–124, 2016. https://doi.org/10.1016/j.compeleceng.2016.02.017.

[8] M. Fakroon, M. Alshahrani, F. Gebali, and I. Traore, "Secure remote anonymous user authentication scheme for smart home environment," *Internet of Things*, vol. 9, p. 100158, 2020. https://doi.org/10.1016/j.iot.2020.100158.

[9] V. Ballal, K. Kumar, N. Megha, and S. R. Rai, "A study and comparison of lightweight cryptographic algorithm," *IOSR Journal of Electronics and Communication Engineering*, vol. 12, no. 4, pp. 20-25, 2017.

[10] L. Kou, Y. Shi, L. Zhang, D. Liu, and Q. Yang, "A lightweight three-factor user authentication protocol for the information perception of IoT," *CMC-Computers, Materials & Continua*, vol. 58, no. 2, pp. 545–565, 2019. https://doi.org/10.32604/cmc.2019.03760.

[11] M. Adeli, N. Bagheri, and H. R. Meimani, "On the designing a secure biometric-based remote patient authentication scheme for mobile healthcare environments," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 3075–3089, 2021. https://doi.org/10.1007/s12652-020-02465-2.

[12] H. Chen, C. Xu, Z. Xu, and X. Tu, "An enhanced lightweight biometric-based three-factor anonymous authentication protocol for mobile cloud computing," *Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 2019, pp. 1682–1691. https://doi.org/10.1109/HPCC/SmartCity/DSS.2019.00230.

[13] P. Kumar and L. Chouhan, "A secure authentication scheme for IoT application in smart home," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 420–438, 2021. https://doi.org/10.1007/s12083-020-00973-8.

[14] A. Gupta, M. Tripathi, T. J. Shaikh, and A. Sharma, "A lightweight anonymous user authentication and key establishment scheme for wearable devices," *Computer Networks*, vol. 149, pp. 29–42, 2019. https://doi.org/10.1016/j.comnet.2018.11.021.

[15] A. Diro, H. Reda, N. Chilamkurti, A. Mahmood, N. Zaman, and Y. Nam, "Lightweight authenticated-encryption scheme for Internet of Things based on publish-subscribe communication," *IEEE Access*, vol. 8, pp. 60539–60551, 2020. https://doi.org/10.1109/ACCESS.2020.2983117.

[16] X. Luo *et al.*, "A lightweight privacy-preserving communication protocol for heterogeneous IoT environment," *IEEE Access*, vol. 8, pp. 67192–67204, 2020. https://doi.org/10.1109/ACCESS.2020.2978525.

[17] M. Rana, Q. Mamun, and R. Islam, "Current lightweight cryptography protocols in smart city IoT networks: A survey," pp. 1–22, 2017. [Online]. Available at: https://arxiv.org/ftp/arxiv/papers/2010/2010.00852.pdf

[18] N. N. Mohamed, Y. M. Yussoff, M. A. Saleh, and H. Hashim, "Hybrid cryptographic approach for Internet of Things applications: A review," *Journal of Information and Communication Technology*, vol. 19, no. 3, pp. 279–319, 2020. https://doi.org/10.32890/jict2020.19.3.1.

[19] D. H. Lee and I. Y. Lee, "A lightweight authentication and key agreement schemes for IoT environments," *Sensors (Switzerland)*, vol. 20, no. 18, pp. 1–18, 2020. https://doi.org/10.3390/s20185350.

[20] S. Sahoo, S. S. Sahoo, P. Maiti, B. Sahoo, and A. K. Turuk, "A lightweight authentication scheme for cloud-centric IoT applications," *Proceedings of the 2019 6th International Conference of Signal Processing and Integrated Networks, SPIN 2019*, 2019, pp. 1024–1029. https://doi.org/10.1109/SPIN.2019.8711757.

[21] M. H. Afifi, L. Zhou, "Dynamic authentication protocol using self-powered timers for passive Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2927–2935, 2018. https://doi.org/10.1109/JIOT.2017.2757918.

[22] M. Sajjad *et al.*, "Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities," *Future Generation Computer System*, vol. 108, pp. 995-1007, 2017. https://doi.org/10.1016/j.future.2017.11.013.

[23] I. R. Chiadighikaobi and N. Katuk, "A scoping study on lightweight cryptography reviews in IoT," *Baghdad Science Journal*, vol. 18, no. 2, pp. 989–1000, 2021. https://doi.org/10.21123/bsj.2021.18.2(Suppl.).0989.

[24] P. K. Dhillon and S. Kalra, "A lightweight biometrics based remote user authentication scheme for IoT services," *Journal of Information Security and Applications,* vol. 34, pp. 255–270, 2017. https://doi.org/10.1016/j.jisa.2017.01.003.

[25] S. Emerson, Y. K. Choi, D. Y. Hwang, K. S. Kim, and K. H. Kim, "An OAuth based authentication mechanism for IoT networks," *Proceedings of the 2015 International Conference on Information and Communication Technology Convergence (ICTC)*, 2015, pp. 1072–1074. https://doi.org/10.1109/ICTC.2015.7354740.

[26] A. Canteaut, S. Carpov, C. Fontaine, B. Lac, and R. Sirdey, "End-to-end data security for IoT: from a cloud of encryptions to encryption in the cloud," *Cesar-Conference.Org*, [Online]. Available at: https://www.cesar-conference.org/wp-content/uploads/2017/11/CESAR2017215Carole Fontaine.pdf.

[27] H. Yang and V. Oleshchuk, "Attribute-based authentication schemes: a survey," *International Journal of Computing*, vol. 14, no. 2, pp. 86–96, 2015. https://doi.org/10.47839/ijc.14.2.805.

[28] A. De Santis, M. Flores, and B. Masucci, "One-message unilateral entity authentication schemes," *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017, pp. 1–6. https://doi.org/10.1145/3098954.3098982.

[29] C. C. Chang, H. L. Wu, and C. Y. Sun, "Notes on 'Secure authentication scheme for IoT and cloud servers'," *Pervasive Mobile Computing*, vol. 38, no. 100, pp. 275–278, 2017. https://doi.org/10.1016/j.pmcj.2015.12.003.

[30] C. C. Chang and H. D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Transaction on Wireless Communication*, vol. 15, no. 1, pp. 357–366, 2016. https://doi.org/10.1109/TWC.2015.2473165.

[31] P. Gope, R. Amin, S. K. Hafizul Islam, N. Kumar, and V. K. Bhalla, "Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localisation services for smart city environment," *Future Generation of Computer Systems*, vol. 83, pp. 629–637, 2018. https://doi.org/10.1016/j.future.2017.06.023.

[32] K. Mansoor, A. Ghani, S. A. Chaudhry, S. Shamshirband, S. A. K. Ghayyur, and A. Mosavi, "Securing IoT-based RFID systems: A robust authentication protocol using symmetric cryptography," *Sensors (Switzerland)*, vol. 19, no. 21, pp. 1–21, 2019. https://doi.org/10.3390/s19214752.

[33] M. Wazid, A. K. Das, V. Bhat K, and A. V. Vasilakos, "LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment," *Journal of Network and Computer Applications*, vol. 150, p. 102496, 2020. https://doi.org/10.1016/j.jnca.2019.102496.

[34] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, “A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers,” *Journal of Supercomputers*, vol. 74, no. 12, pp. 6428–6453, 2018. https://doi.org/10.1007/s11227-017-2048-0.

[35] J. Jeong, Y. C. Min, and H. Choo, “Integrated OTP-based user authentication scheme using smart cards in home networks,” *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008),* 2008, pp. 1–7. https://doi.org/10.1109/HICSS.2008.208.

[36] B. Vaidya, J. Park, S.-S. Yeo, and J. Rodrigues, “Robust one-time password authentication scheme using smart card for home network environment,” *Computer Communications*, vol. 34, pp. 326–336, 2011. https://doi.org/10.1016/j.comcom.2010.03.013.

[37] J. T. Kim, “Analyses of secure authentication scheme for smart home system based on Internet on Things,” *Proceedings of the 2017 International Conference on Applied System Innovation (ICASI)*, 2017, pp. 335–336. https://doi.org/10.1109/ICASI.2017.7988420.

[38] M. Backes, “Real-or-random key secrecy of the Otway-Rees protocol via a symbolic security proof,” *Electronic Notes in Theoretical Computer Science*, vol. 155, no. 1 special issue, pp. 111–145, 2006. https://doi.org/10.1016/j.entcs.2005.11.054.

[39] A. Y. F. Alsahlani and A. Popa, “LMAAS-IoT: Lightweight multi-factor authentication and authorisation scheme for real-time data access in IoT cloud-based environment,” *Journal of Network and Computer Applications*, vol. 192, no. August, p. 103177, 2021. https://doi.org/10.1016/j.jnca.2021.103177.

[40] M. Burrows, M. Abadi, and R. Needham, “A logic of authentication,” *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, pp. 233-271, 1989. https://doi.org/10.1098/rspa.1989.0125.

[41] P. Gope and B. Sikdar, “Lightweight and privacy-preserving two-factor authentication scheme for IoT devices,” *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 580–589, 2019. https://doi.org/10.1109/JIOT.2018.2846299.

[42] Y. Park, “A secure user authentication scheme with biometrics for IoT medical environments,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 11, pp. 607-615, 2018. https://doi.org/10.14569/IJACSA.2018.091185.

[43] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, “A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers,” *Journal of Supercomputers*, vol. 74, no. 12, pp. 6428–6453, 2018. https://doi.org/10.1007/s11227-017-2048-0.

[44] A. Ghani, K. Mansoor, S. Mehmood, S. A. Chaudhry, A. U. Rahman, and M. Najmus Saqib, “Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key,” *International Journal of Communication Systems*, vol. 32, no. 16, pp. 1–18, 2019. https://doi.org/10.1002/dac.4139.

[45] X. Li, J. Niu, S. Kumari, F. Wu, and K. K. R. Choo, “A robust biometrics based three-factor authentication scheme for global mobility networks in smart city,” *Future Generation of Computer Systems*, vol. 83, pp. 607–618, 2018. https://doi.org/10.1016/j.future.2017.04.012.

***IKENNA RENE CHIADIGHIKAOBI** received his B. Sc in Computer Science (Networking) and M. Sc in Computer Science from Universiti Malaysia Sarawak. From 2016 until 2019 he worked as a Network Security Engineer. In 2018 he started a Startup called MailZip, that focus on solving location identification problem in Nigeria and user credit record verification. Since 2019, he has been a Ph.D. student in Computer Science focusing on IoT. His research interest includes security and privacy, authentication and protocol design, blockchain technology, malware, and revise engineering.*

***NORLIZA KATUK** obtained her Bachelor’s degree in information technology from University Utara Malaysia in 2000 and her Master’s degree in computer science from Universiti Teknologi Malaysia in 2002. She obtained her Doctoral degree in information technology from Massey University, New Zealand, in 2012. She is a Senior Lecturer at Universiti Utara Malaysia. Her research interests cover multidisciplinary areas, including information security and privacy, web technology, disaster management and human-computer interaction. She is the chief editor of the Journal of Information and Communication Technology.*

***BAHARUDIN OSMAN** is currently a Senior Lecturer at School of Computing, Universiti Utara Malaysia. His research interests include topics in Information, System Security and Soft Computing in Programming Languages. He is currently doing research on Steganography, Cryptography and Network Security.*