A robust and adaptable four factor mutual authentication and key agreement protocol for wearable sensing devices in WBAN

Major Project Submitted
in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

Submitted By

K.Tejaswar reddy (R170728)
S.K.Afrin (R170680)
S.K.Enus (R170585)
D.Gowtham Kumar (R170523)
S.K.Amer (R170862)
S.K.Afreed Hussain (R170670)
M.Siva Kumar (R170793)
K.Safiya (R170610)
P.Gowhathi Raj (R170611**)**
D.A.Mahesh (R170527)

Under the supervision of

Mr. T. Sandeep Kumar Reddy
Assistant Professor

`` **Department of Computer Science Engineering**

Rajiv Gandhi University Of Knowledge Technologies(RGUKT)
R.K Valley , Kadapa , Andhra Pradesh

Rajiv Gandhi University Of Knowledge Technologies
R.K Valley , Kadapa , Andhra Pradesh

CERTIFICATE

This is to certify that report entitled "***A robust and adaptable four factor mutual authentication and key agreement protocol for wearable sensing devices in WBAN***" submitted by **K.Tejaswar reddy** (R170728),**D.Gowtham Kumar** (R170523),**P.Gowhathi Raj (**R170611**)**,**S.K.Amer**(R170862), **D.A.Mahesh** (R170527),**M.Siva Kumar** (R170793),**S.K.Afreed Hussain** (R170670),**S.K.Afrin** (R170680),**S.K.Enus** (R170585),**K.Safiya** (R170610) in partial fulfilment of the requirements for the degree of award of bachelor of technology in computer science engineering is a bonafide work carried by him under my supervision and guidance.

The report has been not submitted previously in part or full to this university or any other university or institution for the award of any degree or diploma.

INTERNAL GUIDE
Mr. T Sandeep Kumar Reddy
Assistant Professor
Department of CSE
RGUKT RK Valley

HEAD OF THE DEPARTMENT
Mr. N Satyanandram
Head of the Department
Computer Science Engineering
RGUKT RK Valley

DECLARATION

We, hereby declare that this report entitled "***A robust and adaptable four factor mutual authentication and key agreement protocol for wearable sensing devices in WBAN***" submitted by me under the guidance and supervision of ***Mr. T Sandeep Kumar Reddy***, is a bonafide work. We also declare that it has not been submitted previously in part or in full to this university or any other university or institution for the award of any degree or diploma.

K.Tejaswar reddy (R170728)

D.Gowtham Kumar (R170523)

D.A.Mahesh (R170527)

S.K.Afreed Hussain (R170670)

S.K.Enus (R170585)

S.K.Amer (R170862)

P.Gowhathi Raj (R170611)

M.Siva Kumar (R170793)

S.K.Afrin (R170680)

K.Safiya (R170610)

Date: -

Place: - RK Valley

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and who's constant guidance and encouragement crown all the efforts success.

We would like to express my sincere gratitude to ***Mr. T Sandeep Kumar Reddy*****,** my project guide for valuable suggestions and keen interest throughout the progress of the project.

We are grateful to ***Mr. N. Satyanandaram*****,** HOD CSE**,** for providing excellent computing facilities and congenial atmosphere for progressing the project.

At the outset,we would like to thank Honourable Director Madam **, Mrs. K Sandhya Rani** , for providing all the necessary resources and support for the successful completion of my course work.

TABLE OF CONTENTS

Abstract:

The primary concern for every individual nowadays is health. But it is impossible to go to the hospital or a physical health centre at the time of emergency. Remote health care monitoring plays a major role in such types of situations. Remote health monitoring can be done by using applications based on wireless body area networks (WBANs). Remote health care is the main evolution in the Health Care industry. Wireless Body Area Networks (WBAN) are the recent trend in IOT communication which plays a vital role in remote health care. It contains wireless body sensors useful to monitor the person and record data of that person activities. But achieving secure communication between WBANs is the main challenge in remote health monitoring. There are so many schemes proposed for achieving security in WBAN communication. Every scheme has its limitations. As the sensors in WBAN can only use a limited set of resources, we propose a new four factor mutual authentication scheme that ensures safe and reliable communication in WBAN that has advantages over other schemes in terms of performance and security. All nodes in the network ensure secure mutual authentication between each other. Users can easily access the data collected by sensors by authenticating themselves using the registered four factors. The security and safety are validated by using Burrows-Abadi-Needham (BAN) logic and simulated by using the Automated Validation of Internet Security Protocol and Applications (AVISPA) tool. We concluded that our protocol achieves a secure and optimized performance compared to most of the existing schemes.

1. Introduction

Health-care is everyone's top priority, and having access to quality medical facilities is a crucial sign of a country's level of development. It is seen to be a crucial factor in deciding how the country develops. People are now more aware of and concerned about their health as a result of the current COVID19 outbreak. Because of their busy schedules, people don't have the time to care for their elderly relatives or for routine checkups. A physical visit to the hospital during an emergency is also not always practicable. In these situations, many would prefer to finish the task quickly on their phones. Rapid technical development results in the creation of several WBAN applications, such as remote health monitoring systems, Tele Medicine Information Systems, etc., that are useful for such reasons. Since doctors physically cannot constantly monitor their patients, so WBAN technology was developed. A typical architecture for WBAN is shown in Figure 1. The sensors gather physiological information such as body temperature, blood pressure, glucose levels, electrocardiograms, electromyograms, oxygen saturation levels, and motion activities. A coordinator node could be one of these sensor nodes, though. The coordinator node alerts the gateway right away if there is any anomaly. The gateway then compiles the data that was received and delivers it to the appropriate authority, such as a doctor, nurse, family member, hospital, or ambulance, so that they can take the appropriate action. The WBAN devices will be implanted inside of patients' bodies, allowing doctors to track their vitals even when they are not physically present with the patients. The sensors typically have a tiny size, a little computation power, and a short communication range. So, in order to cope with such a resource-constrained world, it is crucial to establish an energy-efficient communication paradigm. The system must be kept confidential, intact, and available at all times.

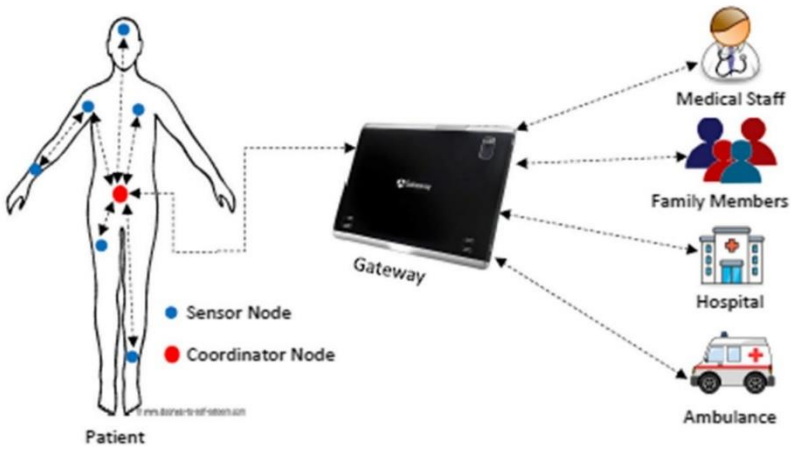

Fig. 1 Architecture of WBAN

1.1 Contribution:

As we seen earlier, most of the existing schemes have different limitations in counteraction to various security attacks and also using more energy, time, storage. Providing a secure communication is the main challenge in WBAN. They also need to communicate in less time with good performance. To communicate securely by using limited set of resources, we need a lightweight and adaptable mutual authentication protocol. Here are our contributions in achieving such protocol:

1. An improved four factor mutual authentication scheme for health-care implementation in WBAN with face recognition possess superior security features over similar schemes.
2. As sensor devices are resource-constrained, and the proposed method utilizes cryptographic hash functions, keyed hash functions, bit-wise XOR operations, and Asymmetric encryption functions and decryption functions and it minimizes computational cost. So that it is claimed to be robust and adaptable.
3. The Protocol's security is examined and validated by using BAN logic and AVISPA tool.
4. Here, we introduced SHA-256 algorithm to strengthen the security in mutual authentication and it sustains various security attacks.
5. We provide the comparative performance evaluation to show its efficiency over similar schemes.
6. This scheme sustains similar existing techniques in terms of communication overhead, and storage requirements.

1.2 Threat model:

For our proposed scheme, we consider two well-known threat models that are CK , and Dolev–Yao (DY) threat models. The assumptions considered therein are given below.

1. An Adversary ***A*** can eavesdrop, alter, discard, send, and receive the data packets sent over the public channel. However, ***A*** cannot thwart the working of a private channel.

2. ***A*** can extract the secret credentials stored in a smart device or a sensor node by using the side-channel attacks on the stolen hardware.

3. ***A*** can guess the identity or password of a user, but not both at the same time.

4. ***A*** can also be an insider of the system

2. Proposed Scheme:

This is the four-factor mutual authentication scheme for health-care based on WBANs which uses superior features that prevents the challenging security attacks. In this scheme we have implemented face recognition which makes this scheme a robust and secure over similar schemes. Below is the brief discussion on face recognition methodology. Initially user scans the face then face recognition algorithm implements the calculations on the distance between face points and creates a depth map of user's face. The sensors we used can obtain the data different types of sensors including RGB depth, EEG thermal and wearable inertial sensors are used to obtain the individuals data.

These sensors may provide extra information and help the face recognition systems to identify face images in both static images and video sequences also we use AI algorithms for face recognition. The main facial recognition methods are featuring analysis, neural network, Eigen faces, and automatic face processing. Although facial recognition technology has come a long way, there is still a need for enhancements to prove accuracy and reliability. Facial recognition security systems can help authorities monitor the flow of people, and alert the proper authorities for immediate action when a known criminal or unidentified adult approaches or enters the facility.

Face detection detects faces by identifying facial features in a photo or a video using machine learning algorithms. It first looks for an eye, and from there it identifies other facial features. It then compares these features to training data to confirm it has detected a face. This process is carried out during the user registration phase. We XOR the bit string received the user's password (UPW_a), after this process we get a face recognition template (UFT_a).

After that we can use passwords and templates when required. After completion of registration when the user again visits then algorithm can automatically detect the user's face and gives the result and stores the data. After enhancing the legitimate part of the system, user can log into the system to authenticate itself and access the sensor's data. The other communicating entities (gateway and sensor) get authenticated during the authentication process to secure robust security. After fortunate authentication among all the communicating entities, the user and the sensor establish a secure session key for upcoming data exchange.

A Face recognition system is a technology capable of matching a human face from a digital image or a video frame against a database of faces. Such a system is typically employed to authenticate users through ID verification services, and works by pinpointing and measuring facial features from an image.

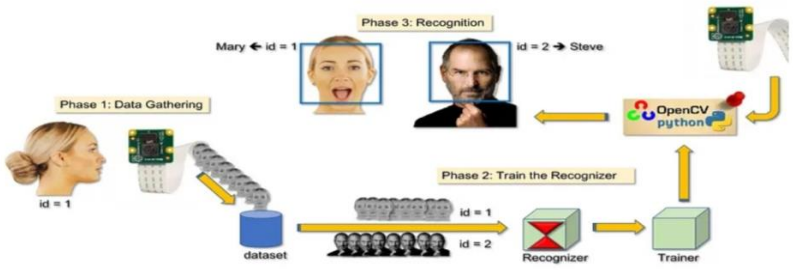

Fig.[2]

Face can be captured with inexperienced camera, so it decreases the cost. More features can capture such as eyes, nose, ears, lips, chin, teeth and cheeks. An Accuracy can be maintained through periodic updates in face. Variation in pose can causes effects, No one can copy the individual faces. It also needs an accuracy of face for scanning (Steady Picture for better results). There should be a problem when face swells or any changes in the face. Easy to integrate (lighting must). Face recognition is having more features, cost ineffective and flexibility to adopt in all situations so that it provides better accuracy in average case.

Table 1 Notations of our proposed protocol

Symbols	Meaning
SA	System administrator
U_a,UID_a,UPW_a ,$UBIO_a$,UFR_a , UFT_a	the User and his User identity,password,User biometric, User Face Recognition and User Face Recognition template.
SN_c , SID_c , $SSec_c$	cth Sensor, Sensor identity, secret.
GN_b , GID_b	bth Gateway, Gateway identity.
$UGSKey_{ab}$ & $SGSKey_{cb}$	shared key between sensor to gateway and user to gateway.
Ur_a, $URand_a$	Random numbers at User's end.
Gr_{1b}, Gr_{2b}, $GRand_b$	Random numbers at Gateway.
Sr_c, $SRand_c$	Random numbers at Sensor.
E(.) & D(.)	Asymmetric encryption functions and decryption functions.
h(.), h_k (.), $\oplus$, &||	Hash fucntion, keyed hash function, XOR operations and concatenation.
PT_{1a}, PT_{2b}, PT_{3c}, PT_{4b}, PT_{5a}	Present timestamps.
SK_{ac}	A fresh session key.
(UPr_a ,UPu_a), ($GPri_b$,$GPub_b$), (SPr_c ,SPu_c)	(Private,Public) key pair of user, gateway, sensor.
ΔT	Transmission delay tolerance.
I	Intruder.
$\mathbb{Z}^*_q$	Multiplicative prime cyclic group

2.1 Proposed protocol for four factor mutual authentication and key agreement protocol for wearable sensing devices

Initialization:

In this phase, the following parameters are initialized for system participants by SA before deployment of the sensors, which are described below:

1. Sensor identity SID_c.
2. Gateway identity GID_b and its public-private key pair ($GPri_b$, $GPub_b$) .
3. Shared symmetric key $SGKey_{cb}$ for sensor node and gateway that is kept into their respective memories.
4. Shared symmetric key $UGKey_{ab}$ for user and gateway that is kept into smart device and memory, respectively.
5. Asymmetric encryption and decryption algorithm.

User Registration:

User U_a has to register with gateway GWN_b to verify himself that he has a valid access to get into database by following the below mentioned steps:

Step 1: User U_a chooses his identification UID_a, password UPW_a, biometric imprint $UBIO_a$, and user face recognition UFR_a at random before taking a picture of his face and converting it to a bitstring. He uses the formula $UFT_a = UFR_a \oplus UPW_a$ to calculate the biometric template from face recognition.

Step 2: After creating his credentials, the user calculates $UR_{1a} = h(UID_a||UPW_a||UBIO_a||UFR_a)$, $UR_{2a} = h(GPub_b||UGSKey_{ab})$, and $UR_{3a} = h(UR_{1a}||UR_{2a}||(UGSKey_{ab} \oplus Ur_a))$ respectively. A new encrypted registration request message, designated $E_{GPubb}(UID_a, UR_{1a}, UR_{3a}$, and $Ur_a)$, is sent from U_a to GWN_b.

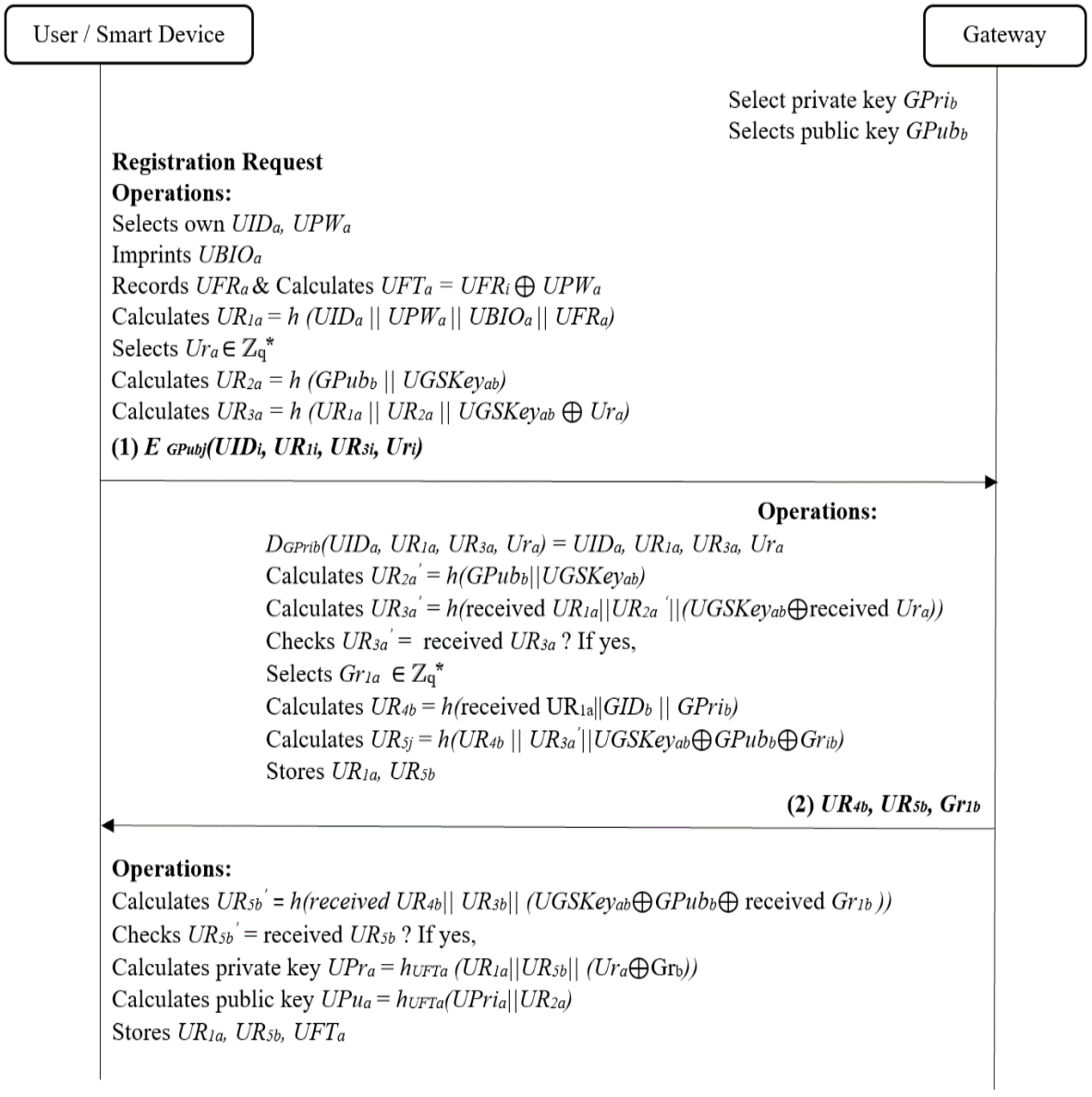

Fig. [3] User registration phase of proposed protocol

Step 3: GWN_b decrypts the registration request received as $D_{GPrab}(UID_a, UR_{1a}, UR_{3a}, Ur_a)$= UID_a ,UR_{1a}, UR_{3a}, Ur_a, and computes UR_{2a}=h($GPub_b$||$UGSKey_{ab}$), UR_{3a} =h(received UR_{1a}||UR_{2a} ||(UGSKey$_{ab}$ ⊕received Ur_a)) to see whether $UR_{3a}^{'}$ =? If the verification is successful, GWN_b chooses a number at random $Gr_{1b} \in Z_q$ to compute UR_{4b} = h(received UR_{1a} ||GID_b ||$GPrI_b$), and UR_{5b} = h(UR_{4b} ||UR_{3a} ||(UGSKey$_{ab}$ ⊕ $GPub_b$ ⊕ Gr_{1b})).GWN_j generates a reply to the registration request message as "UR_{4b}, UR_{5b}, Gr_{1b}" and stores UR_{1a}, UR_{5b} in its memory, which is indexed by the received UID_a.

Step 4: U_a determines whether UR_{5b} =? received UR_{5b} by computing UR_{5b} = h(received UR_{4b} ||UR_{3b} ||(UGSKey$_{ab}$ ⊕ $GPub_b$ ⊕ received Gr_{1b})). If so, U_a creates a private key called $UPri_a$ = h_{UFTa} (UR_{1a} ||UR_{5b} ||(Ur_a ⊕ Gr_b)), a public key called $UPub_a$ = h_{UFTa} ($UPri_a$ ||UR_{2a}), and retains all three in the smart device. After registering, U_a receives a list of the SID_c IDs of the sensors, which U_a can access.

Sensor Node Registration:

After the deployment of sensors into a person, they have to register with gateway GWN_b to make themselves as a valid user of the system and starts monitoring the body by following the mentioned steps:

Step-1: To compute SR_{1c} = h(SID_c||$SSec_c$||Sr_c), SR_{2c} = h($GPub_b$ ||SGSKey$_{cb}$), and SR_{3c} = h(SR_{1c}||SR_{2c}||(SGSKey$_{cb}$ ⊕ GID_b)), SN_c randomly selects its identity SID_c, secret $SSec_c$, and a random number Sr_c Z_q. Thereafter, SN_c sends GWN_b an encrypted registration request, designated EG_{Pubb}(SID_c, SR_{1c}, and SR_{3c}).

Step-2: GWN_b computes SR_{2c} = h($GPub_b$ ||SGSKey$_{cb}$) and SR_{3c} = h(received SR_{1c}||SR_{2c}||(SGSKey$_{cb}$ ⊕ GID_b)) to determine whether SR_{3c} =? received SR_{3c} after decrypting the message D_{GPrib}(SID_c, SR_{1c}, SR_{3c}) = SID_c, SR_{1c}, SR_{3c}. If so, GWN_b chooses a random number Gr_{2b} Z_q to compute SR_{4b} = h(received SR_{3c}||Gr_{2b} ||$GPri_b$) and SR_{5b}= h(SR_{4b} ||received SR_{4b}||(SGSKey$_{cb}$ ⊕ $GPub_b$ ⊕ GID_b)) if the condition is true. It maintains SR_{3c}, SR_{5b} in its database, which is indexed by the SID_c received, and creates SR_{4b}, SR_{5b} as a response to the request.

Step-3: To determine whether SR_{5b} =? received SR_{5b}, SN_c calculates SR_{5b} = h(received SR_{4b} ||SR_{1c}||(SGSKey$_{cb}$ ⊕ $GPub_b$ ⊕ GID_b)). If the inequality is true, it creates a public key $SPub_c$ = h_{SSecc} ($SPri_c$||Sr_c), a private key $SPri_c$ = h_{SSecc} (SR_{1c}||SR_{5b} ||(Sr_c ⊕ $SSec_c$)), and keeps SR_{3c} and SR_{5b} in its memory.

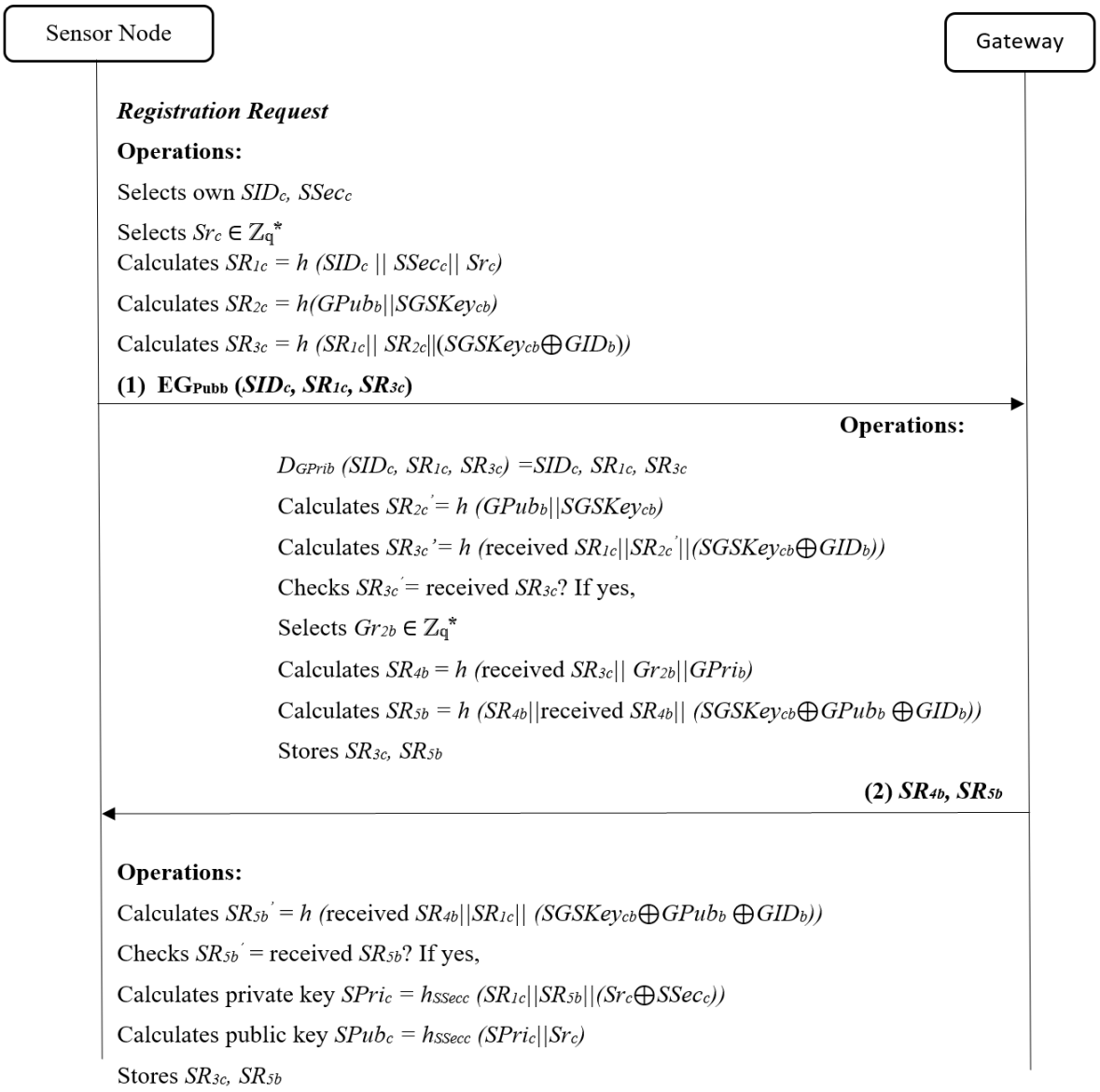

Fig. [4] Sensor node registration of the proposed protocol

Remote User Login:

After the successful registration of both user and sensor, User U_a has to login and enter the system by validating himself by using the registered parameters by using following steps:

Step-1: To access the system, U_a enters his UID_a, UPW_a, and $UBIO_a$ or UFR_a into the smart device.

Step-2: To determine if UR_{1a} =? stored UR_{1a}, the smart device computes URS_a = stored UFT_a $\oplus$ received UPW_a and UR_{1a} = h(received UID_a ||received UPW_a|| $UBIO_a$ ||UFR_a). U_a is provided system access if the condition is true.

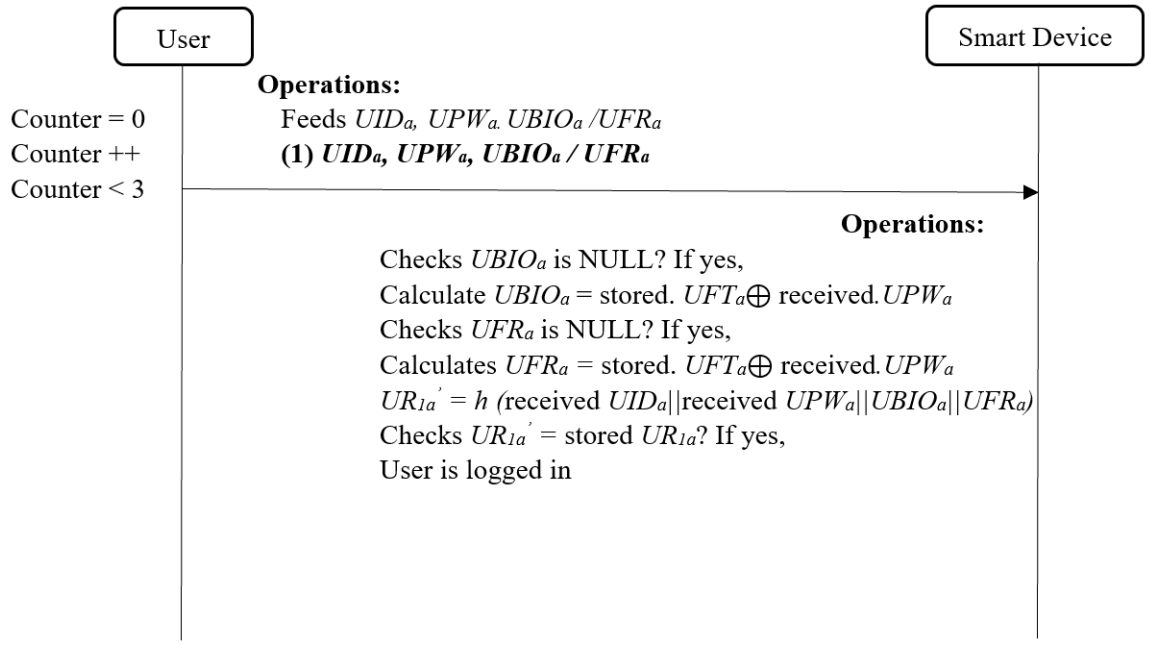

Fig. [5] Remote User Login of the proposed protocol

Mutual Authentication:

After logging in the user, sensor and gateway have to communicate securely by mutually authenticating themselves by executing the below mentioned steps.:

Step-1: U_a chooses a random number using $URand_a \in \mathbb{Z}_q^*$ For the purposes of computing $Access_a = E_{GPubb}$ (UID_a || $URand_a$||SID_c) and $Auth_{1ab}$ = h((stored UR_1 ⊕ stored UR_{5b}) || UPu_a ⊕ PT_{1a} || $URand_a$ ⊕ $UGSKey_{ab}$||$URand_a$). Next Ua creates a brand-new authentication request data packet DP1 = Accessa, Auth1ab, URanda, and PT1a for GWN_b.

Step-2: In accordance with ΔT, GWN_b confirms the data packet's freshness. If it is determined to be fresh, it computes D_{GPrib} ($Access_a$)=(UID_a || $URand_a$ || SID_c) to retrieve the appropriate UR_{1a} according to UID_a. It then determines whether $Auth_{1ab}'$ =? received $Auth_{1ab}$ by computing $Auth_{1ab}'$ = h((stored UR_1 ⊕ stored UR_{5b}) || UPu_a ⊕ received PT_{1a} || received $URand_a$ ⊕ $UGSKey_{ab}$ || received $URand_a$). The user is regarded as authentic if the variables' equality is guaranteed.

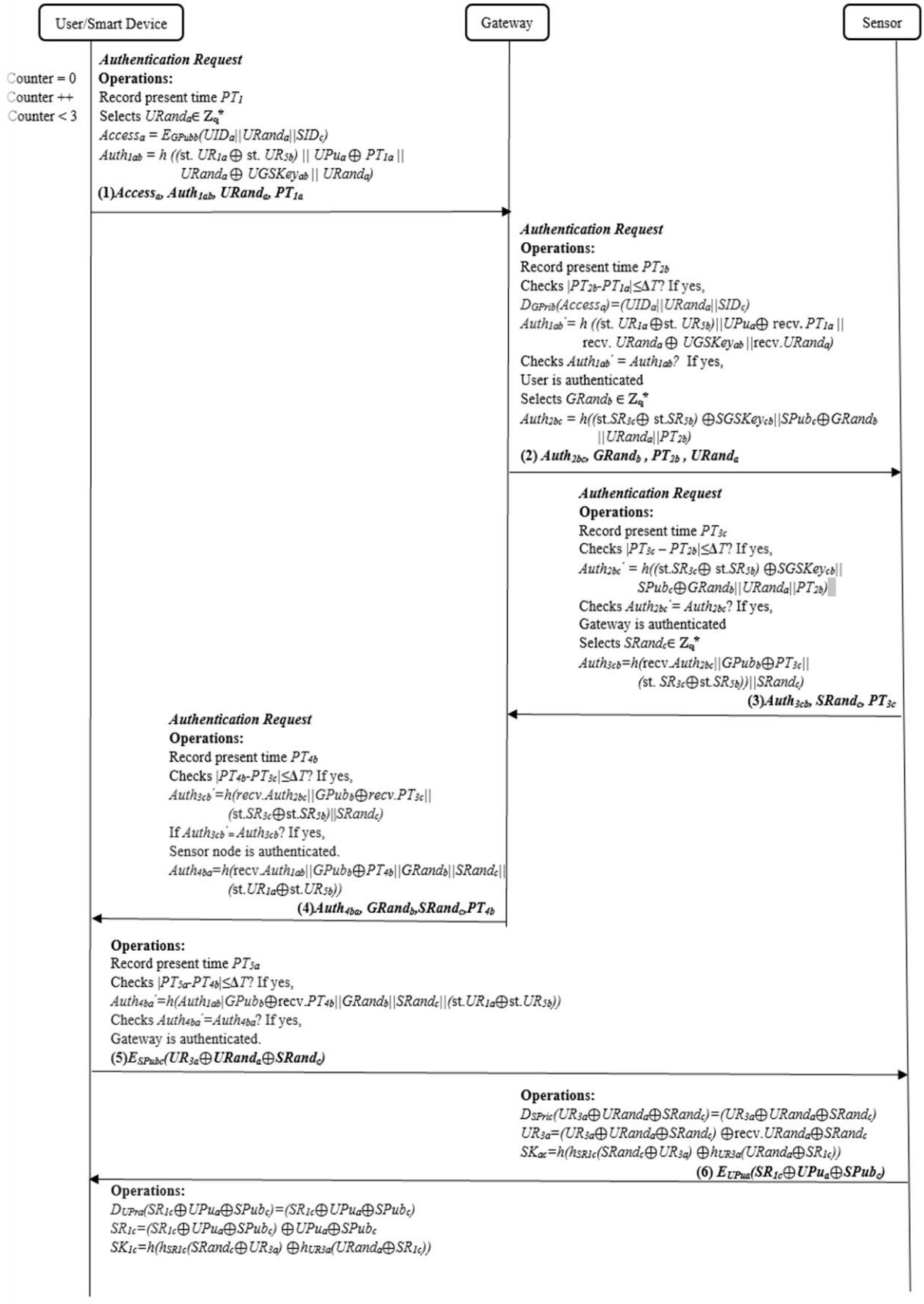

Fig. [6] Mutual authentication of the proposed protocol

Step-3: In order to calculate $Auth_{2bc}$ = h((stored SR_{3c} ⊕ stored SR_{5b}) ⊕ $SGSKey_{cb}$ || $SPub_c$⊕$Grand_b$|| $URand_a$ || PT_{2b}), GWN_b selects a random number $Grand_b \in \mathbb{Z}_q^*$. Then GWN_b creates a brand-new authentication request data packet for SN_c with the following components: $DP_2 = Auth_{2bc}$, $GRand_b$, PT_{2b}, and $URand_a$.

Step-4: If the received data packet is inside the ΔT limit, SN_c verifies this. If so, it determines whether $Auth_{2bc}'$=? received $Auth_{2bc}$ by computing $Auth_{2bc}$ = h((stored SR_{3c}⊕stored SR_{5b}) ⊕$SGSKey_{cb}$ || $SPUb_c$ ⊕$GRand_b$||$URand_a$||PT_{2b}).The gateway is authenticated if both variables are equal.

Step-5: To compute $Auth_{3cb}$ = h(received $Auth_{2bc}$||$GPub_b$⊕ PT_{3c}||(stored SR_{3c} ⊕stored SR_{3b})) || $SRand_c$), SN_c selects a random number $SRand_c \in \mathbb{Z}_q^*$. After then, SN_c composes a new response to GWN_b's authentication request as DP_3 = "$Auth_{3cb}$, $SRand_c$, PT_{3c}.".

Step-6: GWN_b determines whether the data packet received falls inside the ΔT time limitations. If so, it determines whether $Auth_{3cb}'$ =? received $Auth_{3cb}$ by computing $Auth_{3cb}$ = h(received $Auth_{2bc}$||$GPub_b$ ⊕received PT_{3c}||(stored SR_{3c} ⊕ stored SR_{5b}) || $SRand_c$). The sensor is regarded as an authentic entity if the requirement is satisfied.

Step-7:Using GWN_b, $Auth_{4ba}$ is calculated as h(received $Auth_{1ab}$||$GPub_b$⊕PT_{4b}||$GRand_b$||$SRand_c$||(stored UR_1⊕stored UR_{5b})). A new authentication request is created for U_a by GWN_b as $DP_4 = Auth_{4ba}$, $GRand_b$ $SRand_c$, and PT_{4b}.

Step-8: U_a verifies the received data packet's freshness in accordance with ΔT. In order to determine whether $Auth_{4ba}'$ =? received $Auth_{4ba}$, it computes $Auth_{4ba}'$ = h($Auth_{1ab}$|$GPub_b$⊕received PT_{4b}||$GRand_b$||$SRand_c$||(stored UR_1⊕stored UR_{5b})) if it is fresh. The gateway is regarded as a genuine entity if the equality of the variables is guaranteed. In order to establish the session key, the user speaks with the sensor directly by sending the encrypted data packet DP_5 = "E_{SPubc} (UR_{3a}⊕ $URand_a$ $SRand_c$)" to SN_c.

Step-9: UR_{3a} = (UR_{3a} ⊕ $URand_a$ ⊕ $SRand_c$) ⊕received $URand_a$ ⊕$SRand_c$ and session key SK_{ac} = h(h_{SR1c} ($SRand_c$ ⊕UR_{3a}) ⊕h_{UR3a} ($URand_a$ ⊕SR_{1c})) is used by SN_c to decode the data packet that was received. An encrypted data packet $DP_6 = E_{UPua}$ (SR_{1c} ⊕UPu_a ⊕$SPub_c$) is sent from SN_c to U_a.

Step-10: The received data packet is decrypted by U_a as D_{UPra} (SR_{1c} ⊕UPu_a ⊕$SPub_c$) = (SR_{1c} ⊕UPu_a ⊕$SPub_c$), and the session key SK_{1c} is created as h(h_{SR1c} ($SRand_c$ ⊕UR_{3a}) ⊕ h_{UR3a} ($URand_a$ ⊕SR_{1c})). With the agreed-upon session-key SK_{ac}, U_a and SN_c can now speak to each other directly over the public channel. This is a summary of the mutual authentication procedure.

3. Simulation Analysis:

In this chapter, we use a AVISPA tool to perform formal security verification [12,13]. AVISPA tool which makes use of modular and expressive High Level Protocol Specification Language (HLPSL) [14] for integration of source-code and to find out security weaknesses in a method. AVISPA tool translates HLPSL language into an Intermediary Form (IF) with the assistance of HLPSL2IF translator. The AVISPA tool's backends receive this intermediate form for safety checks, and the result indicates if the method is safe or unsafe to use in practice.

The following fields are included in the output format:
• SUMMARY suggests if the protocol for testing is SAFE / UNSAFE or that investigation was determined to be INCONCLUSIVE.
• DETAILS detailing the test protocol's safe settings, the assault findings, and also why the examination was found inconclusive.
• PROTOCOL displaying the protocol's title.
• GOAL stating the analysis's goal.
• BACKEND displaying the name of such back-end used.
• STATISTICS showing the parse time, search time, visited nodes, and depth of the nodes evaluated by the back-end during method execution.

Simulation Analysis

The AVISPA tool uses the HLPSL programming language to validate the security of our proposed approach.

In the AVISPA tool, the Security protocol animator (SPAN) application is utilized to animate the HLPSL requirements and create message sequence charts (MSC) relating to such HLPSL requirements. Fig.5 describes the fundamental MSC relating to proposed approach without taking the intruder into account. The MSC relating to the full protocol implementation with the intruder is depicted in Fig.6.

An overview of the OFMC and CL-AtSe back-end results are illustrated in Figs.7 and Figs8. The findings are inconclusive because the TA4SP and SATMC back-ends do not presently enable bit-wise XOR operations. As a result, they are not included in the paper. The OFMC and CL-AtSe back-end findings show that the proposed scheme is protected against a variety of common attacks like man-in-the-middle, reply, and impersonation attacks formulated on the DY-threat model, and session key privacy is maintained. As a result, our scheme can be used in practice.

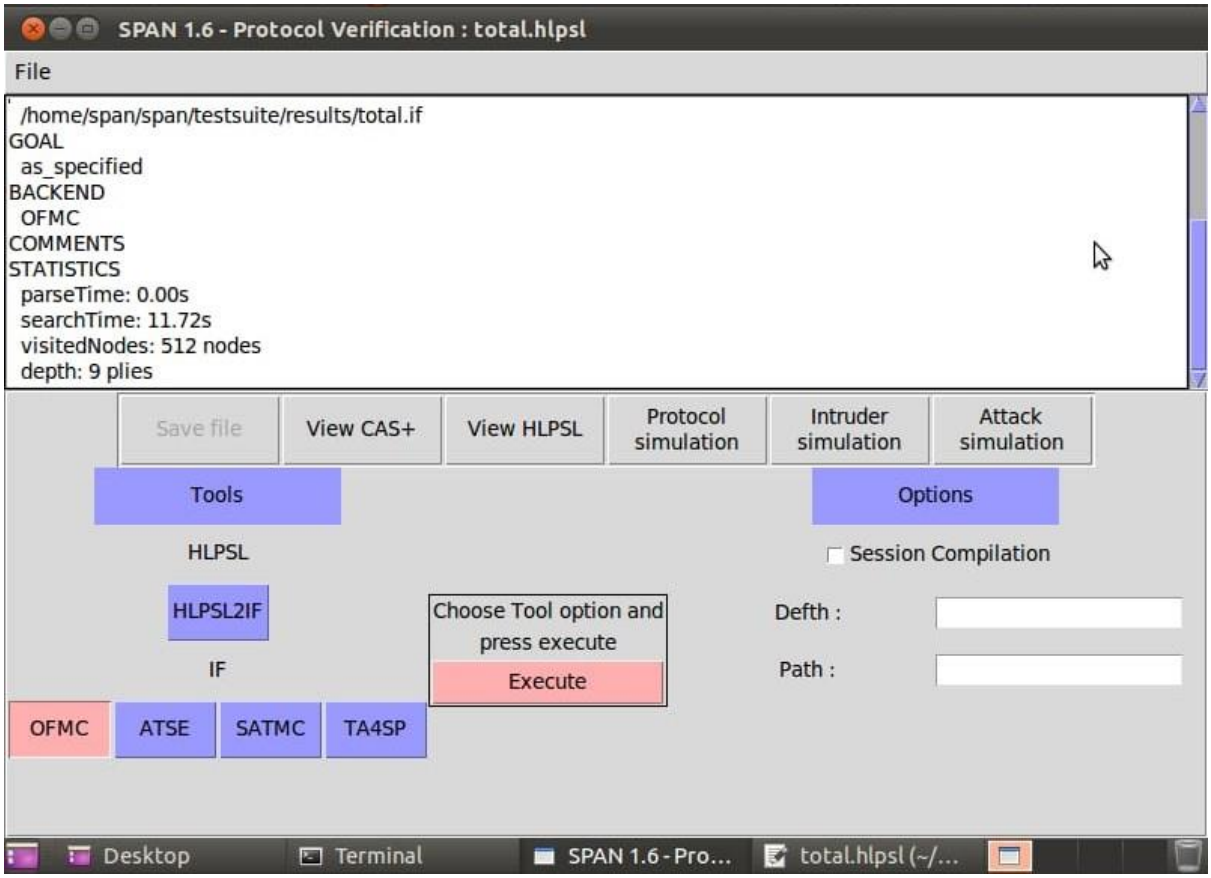

Fig.7(a). Overview of the OFMC results.

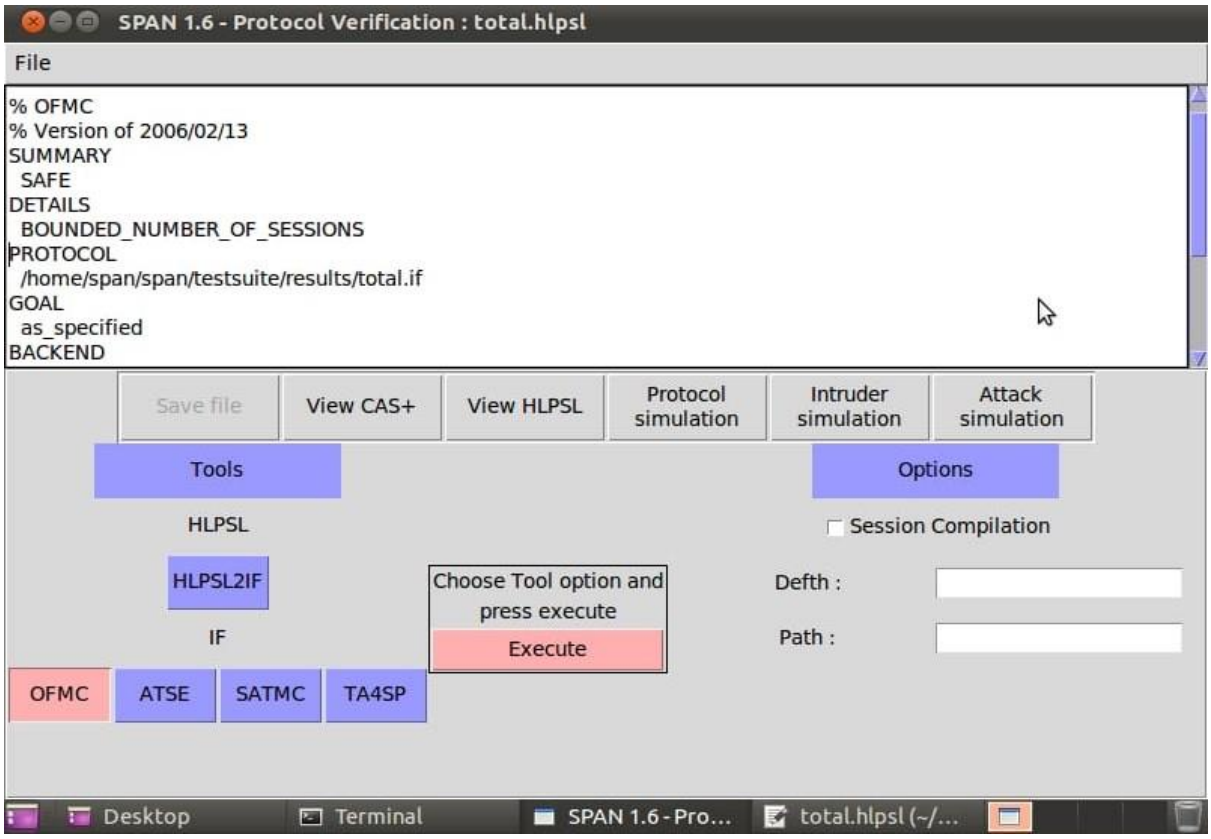

Fig.7b. Overview of the OFMC results.

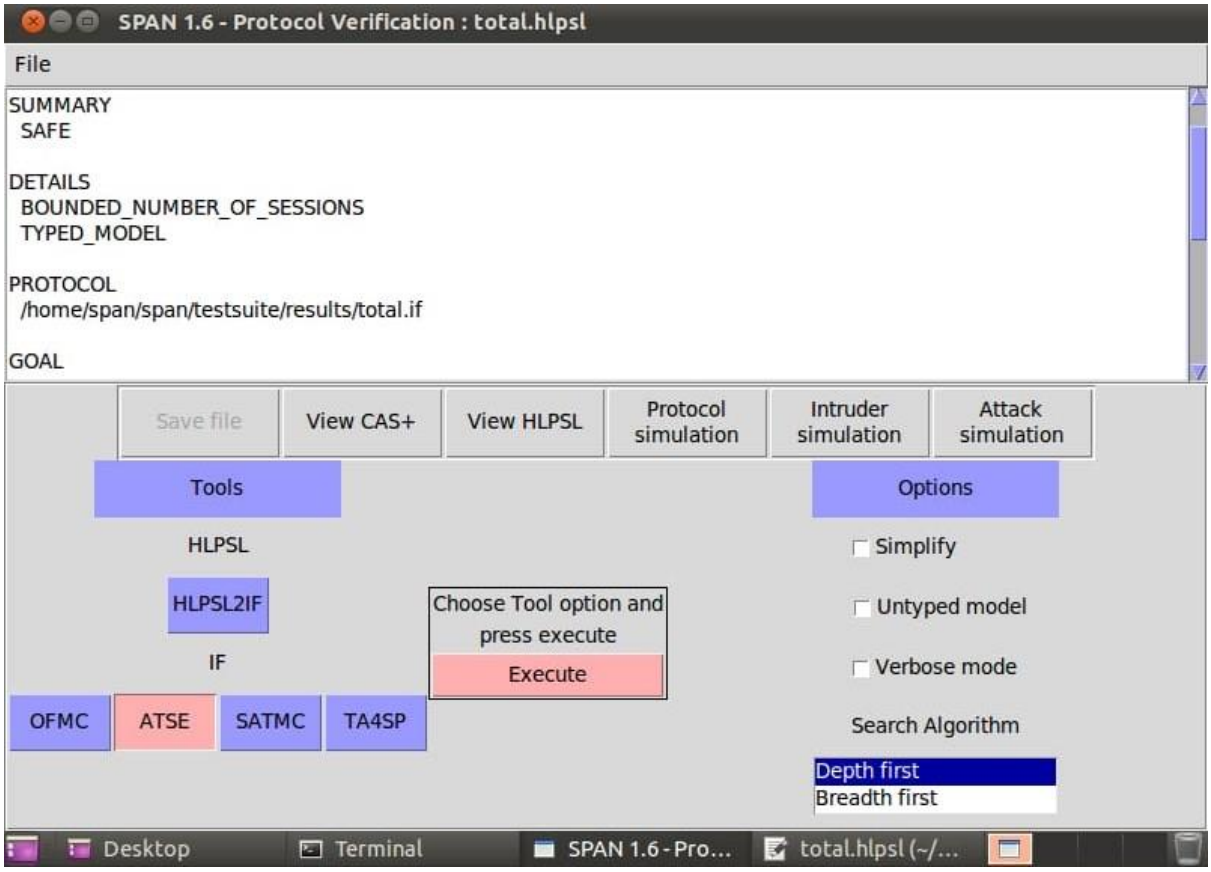

Fig.8a. Overview of the CL-AtSe results.

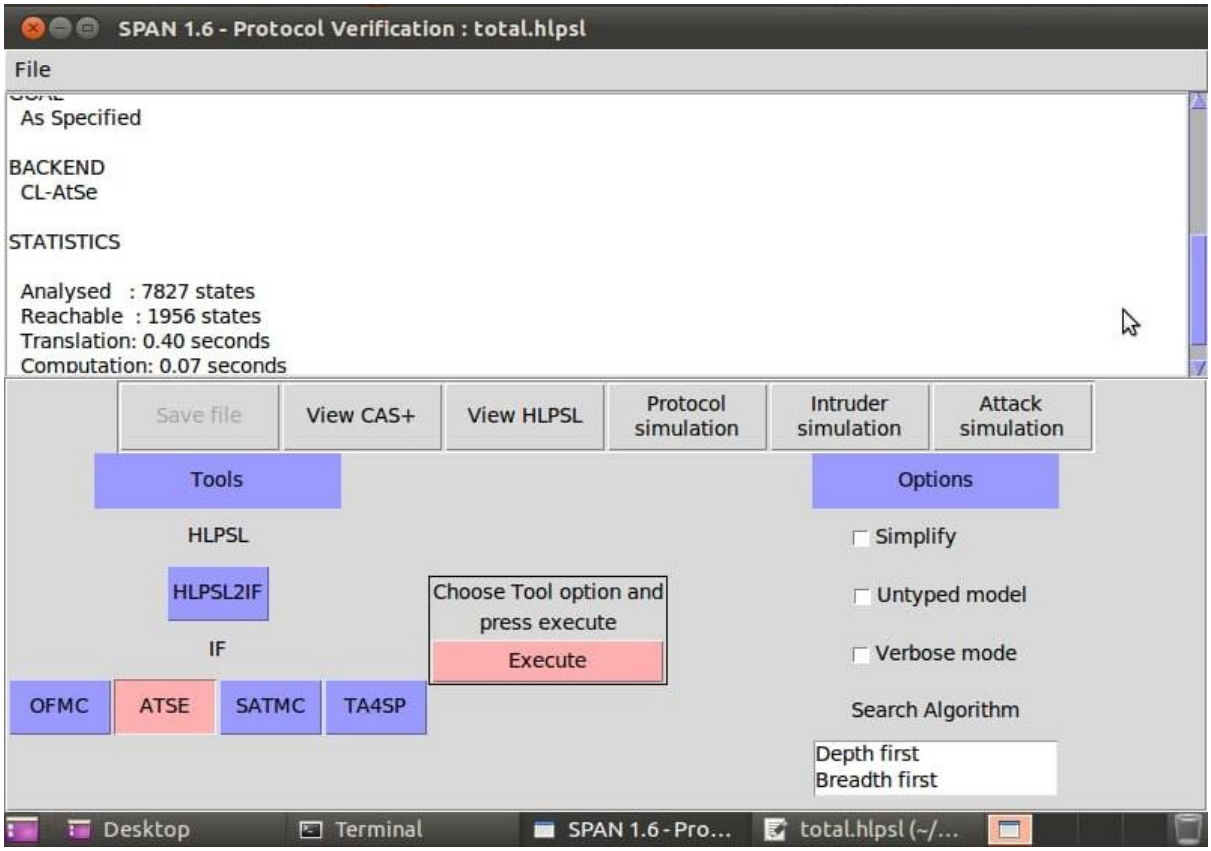

Fig.8b. Overview of the CL-AtSe results.

4. Security Analysis of the proposed scheme:

Here, we do a formal examination of the proposed scheme's security properties using BAN logic and simulations using the AVISPA tool. It offers four-factor user authentication that is secure. The user U a has four secret credentials for uniquely identifying himself: identity (UID_a), password (UPW_a), biometric ($UBIO_a$), and Face scan(UFR_a). In order to circumvent the security of the user authentication process, attacker I must generate DP1 = {$Access_a$, $Auth_{1ab}$, $URand_a$, CT_{1a}} where $Auth_{1a}$ = h ((st. $UR_{1a} \oplus$ st. UR_{5b}) || $UPu_a \oplus PT_{1a}$ || $URand_a \oplus UGSKey_{ab}$ || $URand_a$). It is impossible to create this data packet without knowing the user's four secret credentials, as Ur_{1a} is a hash of all four credentials. Assume I properly estimates UID_a and UPW_a, but in order to compute Ur_{1a}, I need the user's other two biometric data. However, two parameters cannot be predicted simultaneously in polynomial time, according to the DY and CK threat models. Furthermore, unlike identification and password, these other two factors, the user's biometrics, cannot be predicted or fabricated. As a result, the creation of Ur_{1a} is impossible. so, without Ur_{1a} and the shared secret key $UGSKey_{ab}$, forming $Auth_{1ab}$ and so building $Auth_{2ab}$ is impossible. As a result, our system provides safe four-factor user authentication.

Here we describe how our proposed approach can resist the assaults given below:

Stolen Smart Device Attack:

It is resistant to stolen smart device attacks. Assume that attacker I steals the user's smart device and retrieves Ur_{1a}, Ur_{5b}, and UBT_a. I must generate DP 1 = ($Auth_{1ab}$, $URand_a$, CT_{1a}) to attack the system, which is not practicable as explained in point (I). If I attempts to construct the session key SK_{ac} = h(hSR_1 ($SRand_c$,Ur_{3i}) hUr_{3i} ($URand_a$,SR_{1c})) using the parameters contained in the smart device, the attempt will be pointless because the session key is not composed of any of these parameters. As a result, taking smart device I will not disrupt the system's operation. As a result, our system is resistant against stolen smart device attacks.

User Impersonation Attack:

It is resistant to user impersonation attacks. To conduct a user impersonation attack [25], attacker I create the data packet DP_1 = $Access_a$, $Auth_{1ab}$, $URand_a$, CT_{1a}. I can morph any of the three variables while forging this data packet. Assume I modifies $Auth_{1ab}$, with $Auth_{1a}$ = h ((st. $UR_{1a} \oplus$ st. UR_{5b}) || $UPu_a \oplus PT_{1a}$ || $URand_a \oplus UGSKey_{ab}$ || $URand_a$). This change is detected at the gateway during the legitimacy check since $Auth_{2ab}$ is made up of $Auth_{1ab}$, public variables, and the values saved with the gateway. As a result, the parameters do not match when the gateway composes $Auth_{2ab}$ and tries to match it with the received value. As a result, morphing $Auth_{2ab}$ is pointless. Furthermore, when I attempt to morph $URand_a$ or CT_{1a}, the gateway detects changes in these public variables since these values are needed to compute $Auth_{2ab}$ and any differences in timestamp result in a failed freshness check. As a result, any changes to these public values will modify the hash of $Auth_{2ab}$, which will be discovered during gateway integrity checks. As a result, our technique is resistant to user impersonation attacks.

Gateway Impersonation Attack:

It is resistant to gateway impersonation attacks. To carry out a gateway impersonation attack [25], attacker I must fake data packets DP_2 = $Auth_{2bc}$, $GRand_b$, CT_{2b} , $URand_a$ or DP_4 = $Auth_{4ba}$, $GRand_b$,$SRand_c$, CT_{4b} .I must change $Auth_{2bc}$ to perform impersonation by changing DP_2 , where $Auth_{2bc}$ = h((st.SR_{3c}⊕ st.SR_{5b}) ⊕ $SGSKey_{cb}$ || $SPub_c$ ⊕ $GRand_b$ || $URand_a$ || PT_{2b}). The sensor detects any forgery in Auth 4bc during the authenticity check since it is composed of Auth 3bc, a value kept at sensor, and two public parameters GRand b and CT_{2b}, where $Auth_{3bc}$ is constructed using the values saved at sensor and the shared secret key. Morphing $GRand_b$ or CT_{2b} changes the hash of $Auth_{4bc}$, which is detected by the sensor. Variations in timestamp will also result in a failed freshness check. I must change $Auth_{4ba}$, where $Auth_{4ba}$ = h(recv. $Auth_{1ab}$||$GPub_b$⊕PT_{4b}||$GRand_b$||$SRand_c$||(st.UR_{1a}⊕st.UR_{5b})). Because it is made up of three public parameters ($GRand_b$, $SRand_c$, and CT_{4b}) and $Auth_{7ba}$, any forgery in $Auth_{8ba}$ will be detected by the user during the authenticity check. $Auth_{7ba}$ is created by combining the values stored by the user and the shared secret key in the form of $Auth_{1ab}$. Morphing $GRand_b$, $SRand_c$, or CT_{4b} will modify the hash of $Auth_{2bc}$, which the user will see. Furthermore, any differences in the timestamp will result in a failed freshness check. As a result, neither data packet can be transformed. As a result, our method is resistant against gateway impersonation attacks.

Impersonation Attacks / Sensor Node Capture:

It is resistant to sensor node impersonation attacks. To launch a sensor node impersonation attack [25], attacker I must fake the data packet DP_3 = $Auth_{3cb}$, $SRand_c$, CT_{3c} . I must morph one of the three variables when forging this data packet. Assume I modifies $Auth_{3cb}$ such that $Auth_{3cb}$ = h(recv.$Auth_{2bc}$||$GPub_b$⊕PT_{3c}||(st. SR_{3c}⊕st.SR_{5b}))||$SRand_c$). Because $Auth_{3cb}$ is made up of public variables and values saved at the gateway, this change is detected during the legitimacy check. As a result, when the gateway constructs $Auth_{3cb}$ and attempts to match it with the incoming value, the parameters do not match. As a result, morphing $Auth_{3cb}$ is pointless. If I modify $SRand_c$ or CT_{3c}, the gateway will notice since these values are used to calculate $Auth_{3cb}$. As a result, any changes to these public values will modify the hash of the $Auth_{3cb}$, which will be detected during gateway integrity checks. As a result, our method is resistant against sensor node impersonation attacks.

Man-in-the-Middle Attack:

It is resistant to Man-in-the-middle attacks. Assuming the man is between the user and the gateway, attacker I calculates $Auth_{1a}$ = h ((st. UR_{1a} ⊕ st. UR_{5b}) || UPu_a ⊕ PT_{1a} || $URand_a$ ⊕ $UGSKey_{ab}$ || $URand_a$). Even if I create all of these variables by overhearing or estimating theparameters, generating a valid Ur_{1a} is unfeasible due to the user's two biometric credentials, which cannot be predicted. As a result, when Ur_{1a} cannot be created, genuine $Auth_{2ab}$ cannot be formed. The gateway also saves Ur_{5b} , which is made up of several hash and XOR operations. The gateway will identify any changes in Ur_{5b} , resulting in a failed integrity check.

As a result, our approach is resistant to man-in-the-middle attacks.

Replay Attack:

It is resistant to replay attacks. Attacker I can replay any of the data packets in order to carry out a replay attack [26]. In that circumstance, the check on the freshness of the packet's permissible time delay cannot be passed. Even though I passes the timeliness check, I cannot pass the authenticity check on the data packet since the variables include the current timestamp in hash format. As a result, every change in the timestamp affects the hash value of the authenticating variable. As a result, our technique is resistant to replay attacks.

Privileged Insider Attacks/ Denial of Service:

It is resistant to privileged insider attacks. As an insider at the gateway, attacker I has access to the credentials stored with the gateway in order to carry out the privileged insider attack. As a result, I has access to the Ur_{1a} and Ur_{5b} data stored with the gateway during user registration, as well as the SR_{3c} and SR_{5b} data stored with the gateway during sensor registration. Because these credentials are stored in non-invertible hash format, it is impossible to extract any secret parameters from these stored values by reverse engineering. Furthermore, none of these variables are used in the session key's creation. As a result, generating the session key via privileged insider attack is unfeasible. As a result, our scheme is resistant to privileged insider attacks.

User Identification and Password Guessing Attacks:

It is resistant to offline user identification and password guessing attacks. Attacker I can either steal the smart device or eavesdrop on data packets sent over a public channel to guess the user's identity and password. If I steals the smart device, it will extract the Ur_{1a}, Ur_{5b}, and UFT_a stored on it by performing a side-channel assault on the device. I cannot find or forecast identity or password aftergetting Ur_{1a}, Ur_{5b}, and UFT_a because $Ur_{1a} = h(UID_a || UPW_a || UBIO_a || UFR_a)$, which is in one-way hash format, and reverse engineering on hash is infeasible. As a result, taking the smart gadget to guess the user identity and password appears to be pointless.Suppose I has eavesdropped the data packets sent over a public channel, which are $DP_1 = \{Access_a, Auth_{1ab}, URand_a, CT_{1a}\}$,$DP_2=\{Auth_{2bc}, GRand_b, CT_{2b}, URand_a\}$, $DP_3 = \{Auth_{3cb}, SRand_c, CT_{3c}\}$, $DP_4 = \{Auth_{4ba}, GRand_b, SRand_c, CT_{4b}\}$,$DP_5 = \{ESPu_c (Ur_{3a} \oplus URand_a \oplus SRand_c)\}$ and $DP_6 = \{EUPu_a (SR_{1c} \oplus UPu_i \oplus SPu_c)\}$. None of these data packets employ the user's identity or password in their raw form.Because all of these variables are produced by applying hash numerous times, no mathematical computations will work.As a result, our technique is resistant against offline user identification and password guessing attacks.

Session Key Computation Attack:

In this scheme, the session key [27] is defined as $SK_{ac} = h(h_{SR1c} (SRand_c \oplus UR_{3a}) \oplus h\ UR_{3a} (URand_a \oplus SR_{1c}))$, where $URand_a$ and $SRand_c$ are the user and sensor's respective random numbers, which are randomly selected for each new session. Attacker A will be unable to generate the session key using public parameters $URand_a$ and $SRand_c$ since UR_{3a} and SR_{1c} are unknown.

The formulation of the session key is similarly impossible if A try to predict just one of the two values because it requires both parameters to be known concurrently. According to the DY and CK threat models, it is impractical to predict two parameters at once in polynomial time. So, Our system ensures resistance to the session-key computation attack as a result.

Temporary Information Attack:

Considering that attacker I obtains access to one of the temporary [28] variables ($URand_a$, $SRand_c$, $GRand_b$, $UGSKey_{ab}$, and $SGSKey_{cb}$) that are particular to a certain session. Since the session key is calculated using four separate user and sensor characteristics, is $SK_{ac} = h(h\ SR_{1c}\ (SRand_c \oplus UR_{3a}) \oplus h\ UR_{3a}\ (URand_a \oplus SR_{1c})$,If I attempts to formulate the session key by employing temporary variables that are particular to a certain session, then it will be an unsuccessful effort. These four parameters must be present at the same time; leaking any one temporary variable or anticipating another variable will be ineffective. Our system ensures resistance to known session-specific temporary information attacks as a result.

User Anonymity Attack/ Sensor Anonymity Attack:

In this scheme, no participating entity's identity is ever provided over a public channel in the raw format; instead, it is sent in one-way hashed format, which is impossible to decipher. Therefore, the identities of all involved parties are kept a secret. Therefore, this scheme completely protects user anonymity.

Untraceability Attacks:

In this scheme, random integers and the current timestamps are used to create hash format of the data packets parameters and broadcast over a public channel. In order to aid in authenticity checks, the data packets also include the same timestamp and random integers in raw format. Every new session uses different timestamps and random integers to prevent attackers from identifying patterns in the data packets that would identify the source of the data packets. Hence, Our system guarantees untraceability as a result.

Perfect Forward Secrecy:

In this scheme, the long-term keys $UGSKey_{ab}$ and $SGSKey_{cb}$ aid in key establishment and negotiation. The session key will be established as $SK_{ac} = h(h\ SR_{1c}\ (SRand_c \oplus UR_{3a}) \oplus h_{UR3a}\ (URand_a \oplus SR_{1c}))$. Let's say one of the long-term keys is stolen. Since, these keys are not used in the creation of the session key, the session key's resilience is still preserved. The user's and the sensor's random numbers, which are selected randomly for every session, are used to create the session key. Hence, it is impossible to compromise the current session key by acquiring any parameter from the prior session. So, this scheme guarantees perfect forward secrecy.

Allows Mutual Authentication:

In this scheme, the user first sends a data packet for authentication request to the gateway called DP_1 = {$Access_a$, $Auth_{1ab}$, $URand_a$ and PT_{1a}}. The gateway confirms the message's authenticity before creating a new data packet for authentication request for the sensor node, DP_2 = {$Auth_{2bc}$, $GRand_b$, PT_{2b}, $URand_a$}. The sensor node verifies the message and creates a response to the gateway's authentication request as DP_3 = {$Auth_{3cb}$, $SRand_c$, PT_{3c}}. The gateway creates a new data packet for authentication request for the user as DP 4 = { $Auth_{4ba}$, $GRand_b$, $SRand_c$, PT_{4b}} after receiving the response to the request. Finally, our protocol guarantees mutual authentication.

4.10 BAN logic:

In this section, we will go over the formal demonstration of the session key's confidentiality and reliability between user U_a and sensor SN_k. The notations used in the proof are defined in Table 3. The mathematical demonstration of BAN logic is well-known and is based on the notions in Table 4. Table 5 displays the four messages sent between the user and the sensor via the gateway in an idealised fashion, demonstrating how each parameter is produced. In the process, we consider a few real-world assumptions, which are stated in Table6.

Table 2 Ban Notations

Notations	Representations
$D \mid \equiv A$	D considers A trustworthy
$D \lhd A$	D sights A
$D \mid\sim A$	D previously stated A
#(A)	A is assumed to be afresh
$\langle A \rangle B$	A is composed of B
(A, B)	A/B belongs to a part of (A, B)
A, B_K	A/B has been encrypted with K
$\langle A \rangle_{K \mapsto D}$	A has been encrypted with K a parameter of D
$(A, B)_K$	A/B has been hashed by using K
$D \overset{K}{\leftrightarrow} C$	D and C exchange data via K
A/B	If A the numerator part is true, then B the denominator is also true
$D \Rightarrow A$	D has control over A

Table 3 Fundamental rules of proof

Principles	Description
Message meaning rule (MMR) $\frac{D\mid\equiv D \leftrightarrow C, D \lhd \langle M \rangle_N}{D\mid\equiv C\mid\sim M}$	If the numerator indicates that D believes N is shared with R and believes that M, which is made up of N, holds, then the denominator indicates that D believes C to be reliable, and that C previously claimed that M is true.
Nonce verification rule (NVR) $\frac{D\mid\equiv \#(M), D\mid\equiv C\mid\sim M}{D\mid\equiv C\mid\equiv M}$	The denominator indicates that D regards C as trustworthy, which further considers M to be trustworthy, is true if D thinks M to be afresh and C to be trustworthy, which earlier stated M holds.
Jurisdiction rule (JR) $\frac{D\mid\equiv C \Rightarrow M, D\mid\equiv C\mid\equiv M}{D\mid\equiv M}$	The denominator asserts that D considers M trustworthy to be true if D considers C trustworthy, which controls M and D considers C trustworthy, which considers M trustworthy hold.
Freshness conjuncatenation rule (FCC) $\frac{D\mid\equiv \#(M)}{D\mid\equiv \#(M,N)}$	The denominator asserts that D considers (M, N) to be a new is true if D considers M to be freshly holds in the numerator.
Belief rule (BR) $\frac{D\mid\equiv (M), D\mid\equiv (N)}{D\mid\equiv (M,N)}$	The denominator asserts that D thinks (M,N) to be trustworthy is true if D considers M to be trustworthy and it also considers N to be trustworthy holds.
Session key rule (SKR) $\frac{D\mid\equiv \#(M), D\mid\equiv C\mid\equiv M}{D\mid\equiv D \leftrightarrow^{k} C}$	If D believes that C is trustworthy and believes that M (a component of the session key) is true, as stated in the numerator, then D believes that the session key N is shared by D and C is true.

Table 4 Messages were delivered in optimal form.

1	$U_a \xrightarrow{viaGWNb} SN_c : PT_{1a}, URand_a, Access_a : \langle Access_a \rangle_{(UIDa \,||\, URanda \,||\, SIDc)}$, $Auth_{1ab} : UR_{1a} \oplus UR_{5b}, UPu_a \oplus PT_{1a}, URand_a \oplus UGSKey_{ab}, URand_a$
2	$GWN_b \rightarrow SN_k : PT_{2b}, URand_a, Grand_b, Auth2_{bc} : (SR_{3c} \oplus SR_{5b}) \oplus SGSKey_{cb}, SPub_c \oplus GRand_b, URand_a, PT_{2b}$
3	$SN_c \xrightarrow{viaGWNb} U_a : PT_{3c}, SRand_c, Auth_{3cb} : Auth_{2bc}, GPub_b \oplus PT_{3c}, SR_{3c} \oplus SR_{5b}, SRand_c$
4	$GWN_b \rightarrow U_a : PT_{4b}, GRand_b, SRand_c, Auth4_{ba} : Auth_{1ab}, Gpub_b \oplus PT_{4b}, Grand_b, Srand_c, UR_{1a} \oplus UR_{5b}$

Purpose of Objectives :

To demonstrate the trustworthiness and secrecy of the session key among the system's interacting entities, we must accomplish the following four goals:

1. $U_a \mid \equiv U_a \overset{SK_{ac}}{\leftrightarrow} Sn_c$
2. $U_a \mid \equiv SN_c \mid \equiv SN_c \overset{SK_{ac}}{\leftrightarrow} U_a$
3. $SN_c \mid \equiv SN_c \overset{SK_{ac}}{\leftrightarrow} U_a$
4. $SN_c \mid \equiv U_a \mid \equiv U_a \overset{SK_{ac}}{\leftrightarrow} SN$

Authentication of proof :

We see $Assertion_1$ from **Message 1** that is

$Assertion_1$: $SN_c \triangleleft URand_a$

In message meaning rule **MMR**, A_{11} and $Assertion_1$ are substituted, and the result is

$Assertion_2$: $SN_c| \equiv U_a | \sim URand_a$

With A_6 and $Assertion_2$ in place of the nonce verification rule **NVR,** we obtain

$Assertion_3$: $SN_c | \equiv U_a | \equiv URand_a$

A_6 and $Assertion_3$ are substituted in session key rule **SKR** to yield

$$Assertion_4 : \quad SN_c | \equiv SN_c \overset{SK_{ac}}{\leftrightarrow} U_a \qquad \text{...(Objective 3)}$$

We obtain the nonce verification rule **NVR** by changing A_6 and $Assertion_4$

$$Assertion_5 : \quad SN_c | \equiv U_a | \equiv U_a \overset{SK_{ac}}{\leftrightarrow} SN_c \qquad \text{... (Objective 4)}$$

We see $Assertion_6$ from **Message 3** that is

$Assertion_6$: $U_a \triangleleft SRand_c$

We get by putting A_{12}, $Assertion_6$ in message meaning rule **MMR**

$Assertion_7$: $U_a | \equiv SN_c | \sim SRand_c$

We obtain the nonce verification rule **NVR** by changing A_4 and $Assertion_7$

$Assertion_8$: $U_a | \equiv SN_c | \equiv Srand_c$

A_4 and $Assertion_8$ are substituted in session key rule **SKR** to produce

$$Assertion_9 : \quad U_a | \equiv U_a \overset{SK_{ac}}{\leftrightarrow} SN_c \qquad \text{... (Objective 1)}$$

We obtain the nonce verification rule **NVR** by changing A_4 and $Assertion_9$

$$Assertion_{10} : \quad U_a | \equiv SN_c | \equiv SN_c \overset{SK_{ac}}{\leftrightarrow} U_a \qquad \text{... (Objective 2)}$$

Hypothesis:

A_1 : $U_a \mid \equiv \#\{PT_{1a}\}$

A_2 : $GN_b \mid \equiv \#\{PT_{2b}, PT_{4b}\}$

A_3 : $SN_c \mid \equiv \#\{PT_{3c}\}$

A_4 : $U_a \mid \equiv \#\{Ur_a, Gr_{1b}, Sr_k, Gr_{2b}, URand_a, GRand_b, SRand_k\}$

A_5 : $GN_b \mid \equiv \#\{Ur_a, Gr_{1b}, Sr_k, Gr_{2b}, URand_a, GRand_b, SRand_k\}$

A_6 : $SN_c \mid \equiv \#\{Ur_a, Gr_{1b}, Sr_k, Gr_{2b}, URand_a, GRand_b, SRand_k\}$

A_7 : $U_a \mid \equiv U_a \overset{UGSKey_{ab}}{\leftrightarrow} GN_b$

A_8 : $GN_b \mid \equiv GN_b \overset{UGSKey_{ab}}{\leftrightarrow} U_a$

A_9 : $GN_b \mid \equiv GN_b \overset{SGSKey_{cb}}{\leftrightarrow} SN_c$

A_{10} : $SN_c \mid \equiv SN_c \overset{SGSKey_{cb}}{\leftrightarrow} GN_b$

A_{11} : $SN_c \mid \equiv SN_c \overset{URand_a,SRand_k}{\leftrightarrow} U_a$

A_{12} : $U_a \mid \equiv U_a \overset{URand_a,SRand_k}{\leftrightarrow} SN_c$

5. Performance Analysis:

Here, we discuss about the performance of the proposed scheme in terms of Computational cost, communication cost and storage requirements. We compared the performance the proposed protocol with similar schemes []. As the login and mutual authentication steps are used repetitively, we considered the calculation for these two phases only.

5.1 Computation cost

Our scheme consists of different XOR operations, Hash operations and Encryption, Decryption operations. This scheme makes use of different hash operations such as T_A, T_h, T_H denotes the total time required for asymmetric decryption and encryptions, SHA-256 hash operation, keyed hash operation. Above table shows us the comparative study of the existing schemes with proposed schemes in terms of hash invocations. Our scheme performs better than all other schemes by using a smaller number of hash operations than other existing schemes.

Schemes	**User**	**Gateway**	**Sensor**	**Total Comparison Cost**
[5]	$2T_A+3T_h$	$3T_A+T_h$	$2T_A+T_h$	$7T_A+5T_h$
[6]	$3T_A+4T_h$	$5T_A+3T_h$	$2T_A+T_h$	$10T_A+8T_h$
[9]	$2T_A+10T_h$	$5T_A+6T_h$	T_A+4T_h	$8T_A+20T_h$
[10]	$3T_A+8T_h$	$2T_A+5T_h$	$2T_A+5T_h$	$7T_A+18T_h$
[8]	$3T_A+9T_h+T_F+4T_E$	$6T_E+2T_h$	$4T_E+4T_h$	$3T_A+15T_h+T_F+14T_E$
[11]	T_F+15T_h	$10T_h$	$10T_h$	T_F+35T_h
[7]	$11T_h$	$17T_h$	$6T_h$	$34T_h$
[1]	$3T_A+6T_h+2T_H$	T_A+8T_h	$2T_A+5T_h+5T_H$	$6T_A+18T_h+4T_H$
Ours	$3T_h+2T_H+3T_A$	T_A+4T_h	$2T_A+3T_h+2T_H$	$6T_A+10T_h+4T_H$

Table 5: Efficiency comparison

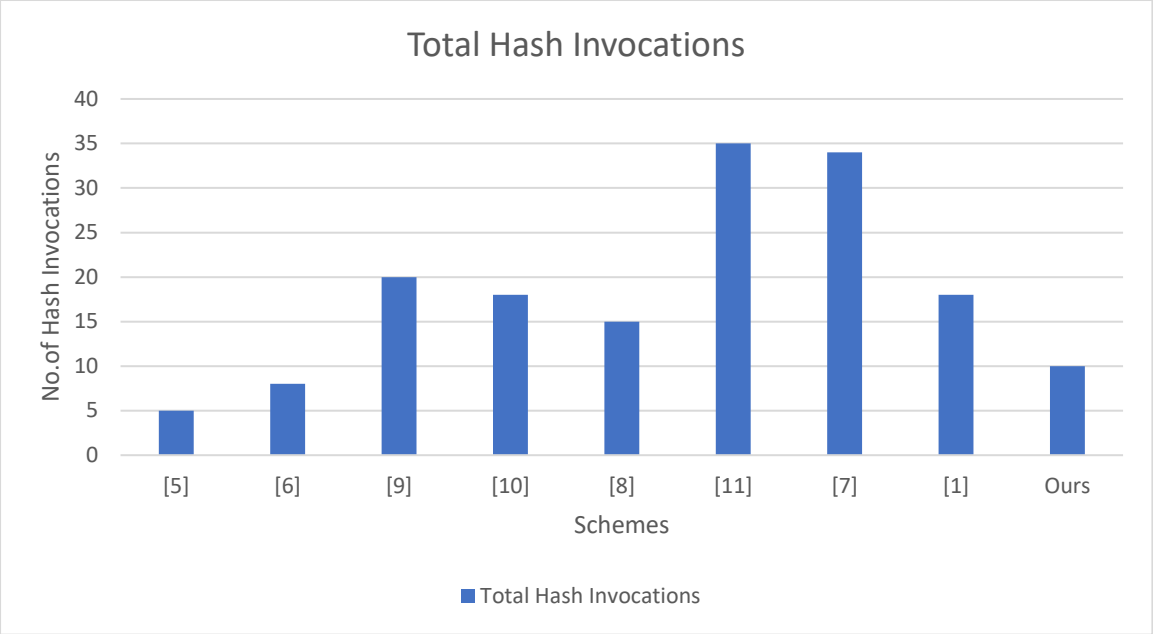

Fig 9: Comparison of Total Computation cost in terms of hash invocations

5.2 Communication Cost

Node to Node Communication	Cost of Communication
User to Gateway	832
Gateway to Sensor Node	352
Sensor Node to Gateway	320
Gateway to User	352
User to Sensor Node	512
Sensor Node to User	512

Table 6: Communication cost

Above table shows us about the communication cost used by mutual authentication phase. Here, we suppose that encrypted messages such as $|access_i| = 512$ bits, SHA-256 hashed values $|Auth_{ij}| = 256$ bits and timestamps, random numbers as 32 bits. As a result, the above messages/tuples can be communicated consuming as per their sizes.

Schemes	Number of messages	Total Cost
[5]	3	1632
[6]	3	2144
[9]	3	2816
[10]	3	2144
[8]	3	1696
[11]	3	1696
[7]	4	2208
[1]	6	2496
Proposed Scheme	6	2880

Table 7: Communication cost comparison

This table shows us about the comparison of communication costs of existing schemes with our proposed one. Our scheme has more communication cost than all as we used SHA-256 for hashing purpose which makes our scheme more robust than all other existing schemes.

5.3 Storage requirements:

In our proposed scheme, all nodes need to save different parameters such as public private key pairs, details of user and sensor and SHA-256 hashed values. Here, we suppose that key value pairs such as | UPr| = | UPu | = | SPri | = | SPub | = | GPub | = | GPri | = 160 bits and keys such as $|UGKey_{ab}|$ = 32 bits, some normal parameters such as ids, passwords, random numbers are considered as 32 bits, SHA-256 hashed values as 256 bits. Here in this scheme, User needs to store UR_{1a}, UR_{5b} , UBT_a , UR_{3a} , UID_a , $UGKey_{ab}$, UPr_a , UPu_b which can be considered as 256+256+32+256+32+32+160+160 = 1184. Sensor needs to store GID_b , SR_{3c} , SR_{5b} , SID_c , SR_{1c} , SGKey , SPub , SPri which can be calculated as 32+256+256+32+256+32+160+160 = 1184 bits.

Gateway needs to store UR_{1a} , UR_{5b} , GID_b , SR_{3c} , SR_{5b} , UGKey , SGKey , GPub , GPri which can be calculated as 256+256+32+256+256+32+32+160+160 = 1440 bits. The maximum storage needed by proposed scheme is 3808 bits. Our protocol needs more storage as it used SHA-256 for hashing which makes our protocol more robust and secure than other schemes.

6. Conclusion

This work proposes an efficient four factor mutual authentication scheme with face recognition for secure communication for health-care based applications in Wireless Body Area Networks (WBANs). The proposed protocol can sustain various types of security attacks. The safety and security of our protocol from different security attacks have been verified by using informal security analysis. The evaluation of secure communication has been verified by using BAN Logic and AVISPA tool. As the sensors used in WBAN are resource-limited, here we used XOR and encryption-decryption operations along with minimised hash operations by using SHA-256 algorithm to improve the security in mutual authentication. Our protocol performed better than the existing protocols in terms of computational time. Finally, a comparative study of results showed that our protocol is more optimized and secure for communication in WBANs.

7. References

1. Rangwani, D., & Om, H. (2022). Four-factor mutual authentication scheme for health-care based on wireless body area network. The Journal of Supercomputing, 1-35.
2. Bayat, M., Das, A. K., Pournaghi, M., Far, H. A. N., Fotuhi, M., & Doostari, M. (2020). A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT. Computer Networks: The International Journal of Computer and Telecommunications, 1(1).
3. Li, X., Ibrahim, M. H., Kumari, S., Sangaiah, A. K., Gupta, V., & Choo, K. K. R. (2017). Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. Computer Networks, 129, 429-443.
4. Gupta, A., Tripathi, M., & Sharma, A. (2020). A provably secure and efficient anonymous mutual authentication and key agreement protocol for wearable devices in WBAN. Computer Communications, 160, 311-325

5. Kumar, P., Lee, S. G., & Lee, H. J. (2012). E-SAP: Efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks. Sensors, 12(2), 1625-1647.
6. He, D., Kumar, N., Chen, J., Lee, C. C., Chilamkurti, N., & Yeo, S. S. (2015). Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. Multimedia Systems, 21, 49-60.
7. Wu, F., Li, X., Sangaiah, A. K., Xu, L., Kumari, S., Wu, L., & Shen, J. (2018). A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks. Future Generation Computer Systems, 82, 727-737.
8. Dodangeh, P., & Jahangir, A. H. (2018). A biometric security scheme for wireless body area networks. Journal of Information Security and Applications, 41, 62-74.
9. Wu, F., Xu, L., Kumari, S., & Li, X. (2017). An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks. Multimedia Systems, 23, 195-205.
10. Srinivas, J., Mishra, D., & Mukhopadhyay, S. (2017). A mutual authentication framework for wireless medical sensor networks. Journal of medical systems, 41, 1-19.
11. Wazid, M., Das, A. K., & Vasilakos, A. V. (2018). Authenticated key management protocol for cloud-assisted body area sensor networks. Journal of Network and Computer Applications, 123, 112-126.
12. Rovai, M. (2021). Real-time face recognition: an end-to-end project.
13. Ioan Buciu, Cristian Grava and Alexandru Gacsadi (2019). Facial Biometric Template Post-processing by Factorization.
14. 1. Kira Alex R (2013) The factors afecting gross domestic product (gdp) in developing countries: The case of Tanzania
15. 2. Vani Rajasekar J, Sathya Premalatha K, Muzafer S (2021) Secure remote user authentication scheme on health care, iot and cloud applications: a multilayer systematic survey. Acta Polytechnica Hungarica 18(3):87–106
16. 3. S.M.R. Islam, D. Kwak, M.H. Kabir, M. Hossain, K. Kwak, “The internet of things for health care: A comprehensive survey”, IEEE Access 3 (2015) 678–708, URL 10.1109/ACCESS.2015.2437951.
17. 4. A.K. Sangaiah, D.V. Medhane, T. Han, M.S. Hossain, G. Muhammad, “Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics”, IEEE Trans. Ind. Inf. 15 (7) (2019) 4189–4196.
18. 5. D. Tse, P. Viswanath, “Fundamentals of Wireless Communication”, Cambridge university press, 2005.