Biometric Based Novel and Dynamic Remote User Access control scheme for WBAN

Trilok Sharma(201206527)
International Institute of Information Technology, Hyderabad
trilok.sharma@students.iiit.ac.in

Ankit Choudhary(201206570)
International Institute of Information Technology, Hyderabad
ankit.choudhary@students.iiit.ac.in

***Abstract*—Wireless body area networks (WBANs) can be applied to provide health care and patient monitoring. However, patient privacy can be vulnerable in a WBAN unless security is considered. Access to authorized users for the correct information and resources for different services can be provided with the help of efficient user access control mechanisms. This paper proposes a new user access control scheme for a WBAN. The proposed scheme makes use of a group-based user access ID, an access privilege mask, a password, the biometrics and the smart card. Also we have introduced concept of group joining and leaving without changing smart card. We show that our scheme performs better than previously existing user access control schemes and needs minimal memory resources. Through a security analysis, we show that our scheme is secure against possible known attacks.**

I. Introduction

In a wireless body area sensor network (WBAN), miniature low-power sensor nodes are placed around a patient's body for monitoring their body functions and the neighboring environment (Ghasemzadeh and Jafari, 2011; Liang et al., 2012; Otto et al., 2006; Zois et al., 2012). With the help of a WBAN, a patient's health related information, including their temperature, respiration, heart rate, pulse oximeter, blood pressure, blood sugar, and pH can be remotely monitored (Ameen et al., 2012). To achieve the maximum benefit, this information must be continuously processed in real time. The medical information must be shared and accessed by various levels of

users such as healthcare staff, researchers, government agencies, and insurance companies to make important decisions such as clinical diagnoses and emergency medical responses for the patients (Li et al., 2010). The bio-sensors are placed on a patient's body to transmit sensing data through a secure channel to a small body area network gateway. The gateway then locally processes the data and resends it through a secure channel to the external net- work router and then onto the medical server at the hospital. The results are then observed and analyzed by the medical staff/doctors charged with monitoring patients. A typical example of a WBAN is shown in Fig. 1 (Li et al., 2010). In this scenario, a patient wears various bio-sensors. A centralized control device is used to transmit data in and out of the net- work. This control device can also be used as a gateway be- tween the internal network and the base station. The base station is connected with the external network. The communication of health related information between sensors on a patient's body in a WBAN over the Internet to medical servers must be strictly private and confidential (Alemdar and Ersoy, 2010; Kwak et al., 2009; Seyedi et al., 2013; Singelee et al., 2008; Venkatasubramanian et al., 2010). Authenticated medical data transmissions are essential requirements for a WBAN because false or unauthenticated medical information may lead to incorrect treatments or diagnoses for patients. Therefore, the transmitted information must be encrypted to protect patient privacy. In addition, the medical staff of the hospital that collects the data must be confident that the data are unaltered and indeed originate from the specified patient. The major challenges in a WBAN are security, robustness, and scalability. The size and resource constraints of the bio-sensors also play a crucial role in the success and reliability of a WBAN (Singelee et al., 2008). Health care staff can directly access data from the body area network of a patient after

successful authentication. A survey on wireless body area net- works can be found in Klaoudatou et al. (2011), Latre et al. (2011) and Otto et al. (2006). Scalability, in terms of number of sensors and patients, is an important factor in this type of network. User access control is an essential requirement in pro- viding security and data privacy for a WBAN. User access control is critical to the successful operation and extensive adoption of wireless body area network services. The security framework for a WBAN should consist of user authentication (identity verification), user authorization (access provided to user) and user accountability (monitoring activity and controlling access) to control user access and prevent different types of attacks. User access control can identify and impose different access privileges for different types of users. In a typical WBAN, different doctors, health care staff, and medical insurance company agents are the major users, but access to all medical information of a particular patient may not be required for all types of users. For example, a con- cerned doctor can retrieve his/her patient's data but no other patient information.

This paper considers a WBAN where sensor nodes are sufficiently small and efficient to ensure long battery life. The elec- tronics of a WBAN sensor node are designed to detect and transmit low frequency and low amplitude physiological sig- nals. The sensor node hardware requires a wireless link (AM152100 IC) from an AMI semiconductor used for MICS

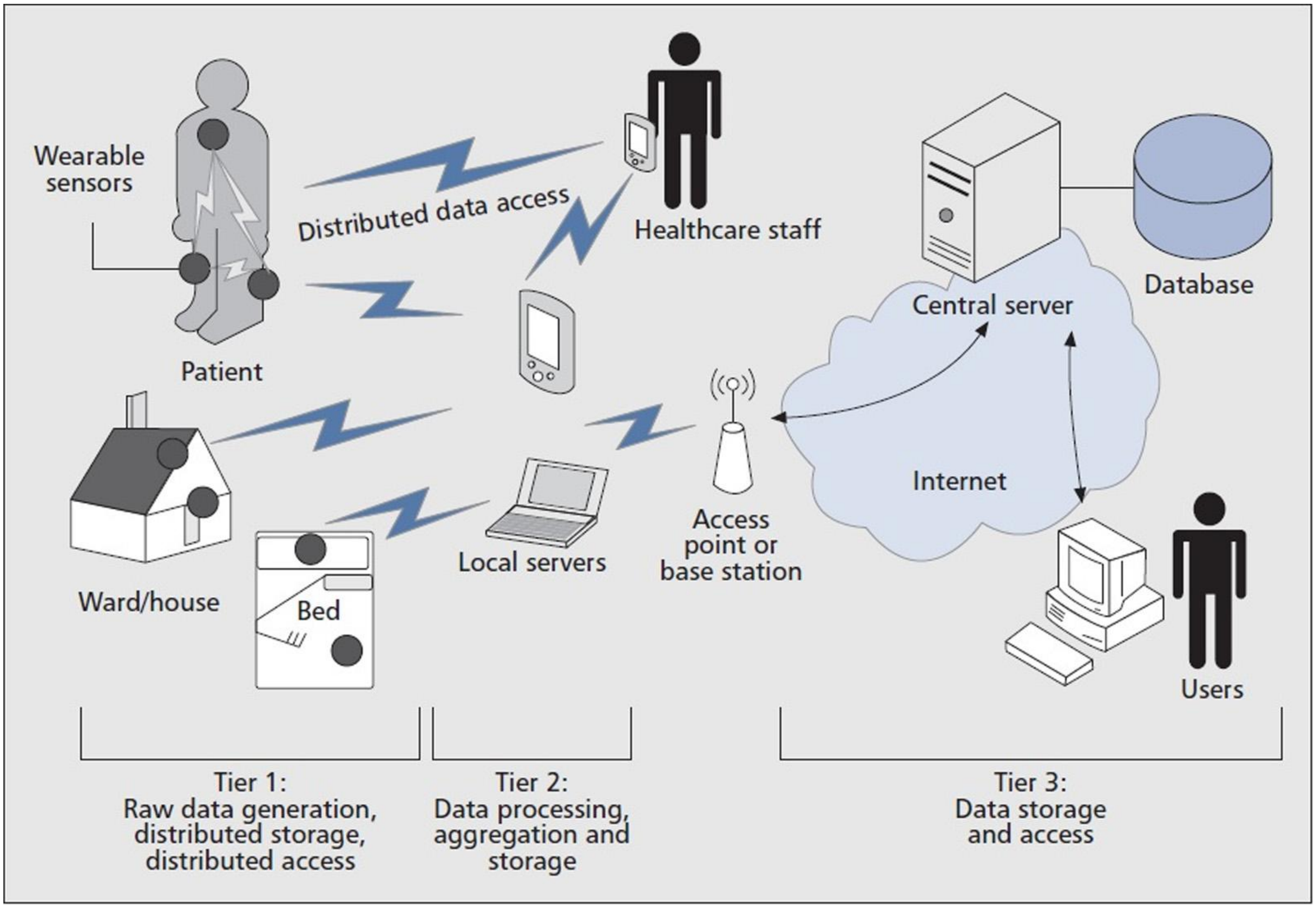

Fig. 1. A general three-tier architecture of WBAN

band generation. Ameen et al. (2012) compared a medical WBAN and a general WSN, clearly mentioning that both general WSNs and medical WBANs have limited resources in battery, computation, and memory while both exhibited dynamic network scale, heterogeneous device ability, and dense distribution. WBAN sensors are single-function, safe, costly and quality devices, and WSN sensors are multi-functional, low cost, redundancy-based reliable devices. In general, a WBAN follows a small-scale star network where there is no device redundancy in the deterministic node distribution; the traffic is periodical and unidirectional, and each channel should be a specific medical channel. However, a general WSN typically has a large scale hierarchical network where redundant and random node distributions are followed. The traffic may be unidirectional or bidirectional, and it generally follows point-to-point communications where obstacles are unknown.

We proposed a new method in which using a single smart card user can join any group of sensor nodes as well as can leave at any time. In our algorithm there are only four message communication and also no storage at BS or SN. Also our algorithm is strongly resistant to password guessing attack.The algorithm will be explained in later sections.

II. Motivation

Our scheme is motivated by the following considerations. In WBAN, external parties (users), those are authorized to access data, should get access as and when they demand. In order to allow authorized access of the real-time data from the sensor nodes inside WBAN to the authorized users on demand, there is a great need for user access control before allowing them to access the real-time data inside WBAN for which they are per- mitted. In healthcare applications, monitoring patient's condi- tions by the expert doctors is very essential. Thus, real-time data sensed by the sensors in a patient's body can be moni- tored directly by an authorized external user (doctor in that hospital) as and when demand is made. Based on critical and emergency situation of the patient, the doctor can take neces- sary action by instructing the nurses/medical staffs in the hos- pital for the patient. Hence, before allowing access to the sensitive and private real-time data of the patients, the external user (doctor) must be authenticated for a particular access privilege by the base station (medical server) as well as sensor node in the network. Considering these points, the user access control in WBAN for healthcare applications becomes a prom- inent research field.

III. Mathematical Background

Following mathematical preliminaries for better understanding of our schemes:

A. One-way hash function

A one-way hash function $H : \{0,1\}^* \rightarrow \{0,1\}^n$ takes an arbitrary-length input $x \in \{0,1\}^*$, and outputs a fixed-length (say, n-bits) message digest or hash value $H(x) \in \{0,1\}^n$. In addition, it has the following important properties :

- H can be applied to a data block of all sizes. For any given input $x \in \{0,1\}^*, H(x) \in \{0,1\}^n$ is relatively easy to compute which enables easy implementation in software and hardware.
- Output length of $H(x) \in \{0,1\}^n$ is fixed.
- From a given hash value $y = H(x)$ and the given hash function $H()$, it is computationally infeasible to derive the input x. This property is known the one-way property.
- For any given input $x \in \{0,1\}^*$, finding any other input $y \in \{0,1\}^*$, with $y \neq x$, such that $H(y){=}H(x)$ is computationally infeasible. This property is known the weak-collision resistance property.
- Finding a pair of inputs $(x,y) \in \{0,1\}^*\{0,1\}^*$, with $x \neq y$, such that $H(y){=}H(x)$ is also computationally infeasible. This property is known the strong-collision resistance property. Example : SHA-1.

IV. The proposed scheme

This section discusses our proposed user access control scheme. Our scheme consists of the following phases which are described in the following subsections.

TABLE I
NOTATIONS USED FOR OUR PROPOSED SCHEME.

Symbol	Description
U_i	i^{th} User
ID_i	i^{th} User Identity
PW_i	i^{th} User Password
B_i	i^{th} User Biometric
APM_i	i^{th} User Access Prieviledge Mask
ID_{sj}	j^{th} Sensor Node ID
SN_j	j^{th} Sensor Node
G_{id_x}	Group of SN's with group ID 'x'
BS	Base Station
SN	Sensor Node
MK_{sj}	Master key of sensor node SN j
K_j	secret key of sensor node SN j
T_j	Bootstrapping time of sensor node SN j
RN_{ui}	Random number for user U_i
S	Secret key of base station of 1024 bits
$H(\cdot)$	Secure collision-resistant one-way hash function
$A\|\|B$	Data A concatenates with data B
$E_K(M)$	Symmetric encryption using the key K
$D_K(M)$	Symmetric decryption using the key K
$\Rightarrow$	Secure channel
$\rightarrow$	Insecure channel

1) Pre deployment phase: **Pre deployment phase:** This phase is used to preload the keying materials to all sensor nodes prior to their deployment. It is performed offline by the (key) setup server. The setup server in our scheme is the base station(BS). This phase is implemented offline by the base station prior to the deployment of sensor nodes(SN) in a target field. The pre-deployment phase consists of the following steps:

Step 1: For each deployed sensor node SN_j, the base station assigns a unique identifier ID_{sj}.

Step 2: The base station also assigns a unique randomly generated master key MK_{sj} for each deployed sensor node SN_j, which is only shared with the base station.

Step 3: BS chooses own secret parameter 'S' of 1024 bit.

Step 4: Depending on probable user query BS prepare group based user access privilege mask (APM) and prepare an access list consisting of APM and the respective access group identity $G_i d$.

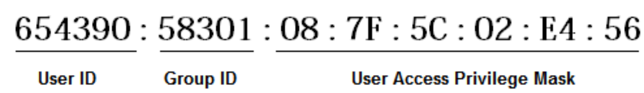

Fig. 2. Access List of user

Step 5: Store ID_{sj}, MK_{sj} on BS

Step 6: Once the set of network parameters are selected, the base station (BS) loads the following information into the memory of each sensor node SN_j prior to its deployment in offline: (i) a unique node identifier ID_{sj}; (ii) its own master key MK_{sj}.

Pre deployment phase
BS assigns ID_{sj},MK_{sj} for each SN_j.
BS chooses own secret parameter 'S'.
BS prepares list consisting ID_i, G_{ID_x}, APM_i.
BS stores all above information in its memory.

Fig. 3. PreDeployment Phase between BS and SN

2) Post deployment phase: **Post deployment phase:** This phase helps the sensor nodes and the base station to establish secure connections between them. As soon as sensor nodes are deployed, their first task is to locate physical neighbors within their communication ranges. For secure communication between sensor nodes, the nodes must establish pairwise secret keys between them. Because the major focus in this paper is addressing the user access control problem, we assume that nodes in a WBAN can establish secret keys by using existing key establishment schemes. For example, we can use an unconditionally secure key establishment scheme (Das AK, 2009) for pairwise key establishment between nodes in each cluster. Because our primary focus is on how authorized users belonging to different groups can access the real-time data for monitoring the sensors node, we require secure communication between the sensor nodes and the authorized users. Once deployed, each sensor node chooses its own secret key K_j and then sends a message with its node identity ID_{sj}, bootstrapping time T_j, and encrypted information containing K_j, ID_{sj} and T_j to the base station:

$SN_j \Rightarrow BS :< ID_{sj}, T_j, E_{MK_{sj}}(K_j, ID_{sj}, T_j)>$

After receiving the message from the sensor node SN_i, the BS decrypts $E_{MK_{sj}}(K_j, ID_{sj}, T_j)$ with the master key MK_{sj} of SN_j, and then checks the validity of the received information K_j, ID_{sj} and T_j. Note that T_j is the bootstrapping time of the sensor node ID_{sj}. The BS further checks if $|T_j - T_j^*| < \triangle T_j$, where T_j^* is the current system timestamp of the BS and $\triangle T_j$ is the expected time interval for the transmission delay. If the check holds, then the BS stores K_j for the sensor node SN_j.

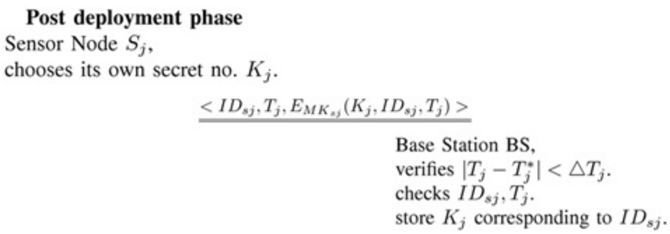

Fig. 4. PostDeployment Phase between SN and BS

3) **Registration Phase:** In the registration phase, a user U_i must register with the base station to access the real time data from a specific sensor node. This phase consists of the following steps:

Step R1:

U_i selects his identity ID_i and password PW_i. He also imprints biometric information B_i to the fuzzy extractor and achieves Gen(B_i) =(R_u, P_u). Then, he computes w=$H(ID_i||PW_i||R_u)$ Then user U_i send $<ID_i, w>$ information through a secure channel.

$U_i \Rightarrow BS :< ID_i, w>$

Step R2:

Base station computes r_i=$w \oplus H(ID_i||S)$. Then, BS issues a smart card to U_i including the security parameters $r_i, H(\cdot)$.

$BS \Rightarrow U_i : S < r_i, H(\cdot)>$

Step R3:

User U_i generates secret Random No. K. User U_i computes L_i=$K \oplus R_u$, V_i=$H(ID_i||PW_i||k)$, J_i=$r_i \oplus w \oplus R_u$ = $H(ID_i||S) \oplus R_u$.

U_i also define HG_i and GL_i where HG_i is the Hashed Group ID, GL_i is set of Group ID, initially HG_i=NULL, GL_i=ϕ.

Step R4:

User store the calculated information on the smart card. S($r_i, H(\cdot), P_u, L_i, V_i, J_i, HG_i, GL_i$).

4) **Login Phase:** In this phase we steps followed are :

Step L1:

U_i inputs ID_i' and PW_i', and imprints B_i' at fuzzy extractor and calculates R_u' = Rep(B_i', P_u), computes K'=$L_i \oplus R_u$' then the smart card verifies V_i?=$H(ID_i'||PW_i'||K')$. If condition holds, goto next step.

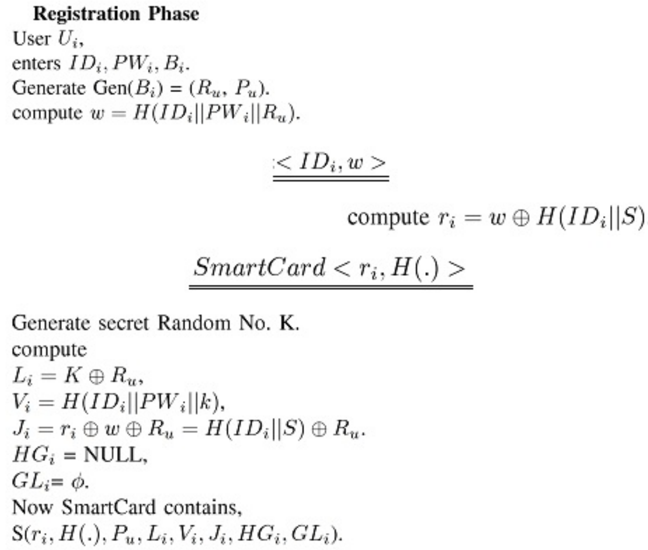

Fig. 5. Registration Phase between User and BS

Step L2:

User enters the G_{id_x}, where G_{id_x} is the group which user intents to access. smart card verifies is user belongs G_{id_x} by searching it in list GL_i. Smart card generates a random number RN_{ui}, then computes w'=$H(ID_i'||PW_i'||R_u')$, M_1=$w' \oplus RN_{ui}$ and M_2=$H(r_i||RN_{ui}||G_{id_x}||T_1)$, where T_1 is current timestamp. Generate K_{ui}=$J_i \oplus R_u' \oplus T_1$=$H(ID_i||S)$ $\oplus T_1$. Calculate M_3=$E_{k_{ui}}(G_{id_x}, M_1, M_2, RN_{ui})$. User sends $< ID_i, M_3, T_1>$ to the Base station.

$U_i \rightarrow BS :< ID_i, M_3, T_1>$

Step L3:

Base Station checks $|T_1 - T_1^*| < \triangle T_1$, where T_1^* is the present Timestamp. Generate h_{si}=$H(ID_i||S)$. Generate K_{ui}=$h_{si} \oplus T_1$. So we get K_{ui} which is used to decrypt M_3. Decrypting M_3 we get $G_{id_x}, M_1, M_2, RN_{ui}$.

Step L4:

Compute r_i'=$M_1 \oplus RN_{ui} \oplus h_{is}$, M_2'=$H(r_i'||RN_{ui}||G_{id_x}||$ $T_1)$. If M_2?=M_2' then BS accept user login request and store the information ID_i, RN_{ui}, G_{id_x} in the respective table of user. Check G_{id_x} is authorized for user U_i.

Step L5:

Compute Key_{ij}=$H(ID_i||ID_{sj}||MK_{sj}||K_j)$, $Token_{ij}$= $H(ID_i||ID_{sj}||T_1||T_2||APM_i||MK_{sj}||M_1)$, where MK_{sj} is the master key shared between BS and Sensor node S_j. We compute token and key for each sensor node S_j, where j=1 to n. for which G_{id_x} is accessible.

Step L6:

Compute M_4=$E_{K_{ui}}(ID_i, ID_{sj}, Key_{ij}, Token_{ij}, APM_i, T_1,$ $T_2)$. Base station sends $< ID_i, M_4, T_2>$ to the user

$BS \rightarrow U_i :< ID_i, M_4, T_2>$.

Step L7:

User verifies $|T_2 - T_2^*| < \triangle T_2$.

Step L8:

Decrypt M_4 using K_{ui}, Checks ID_i ?= ID_i', T_1 ?= T_1',

T_2 ?= T_2', where ID_i', T_1' and T_2' are information retrieved from M_4. If true accept and Store $ID_{sj}, Key_{ij}, Token_{ij}$ for a given S_j. Store T_1, T_2 for G_{id_x}.

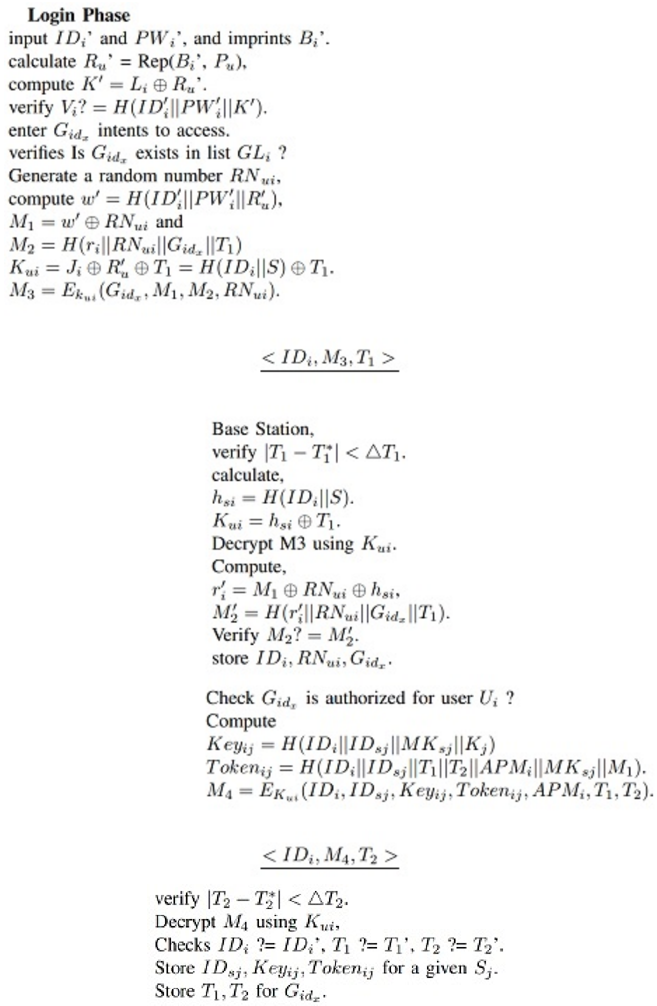

Fig. 6. Login Phase between User and BS

5) **Group joining Phase:** Group Joining Phase is the scenario when a user wants to access different set of Sensor nodes. So a user having a single smart card can access multiple groups of Sensor nodes according to the proper authorization of the user, by the base station. So a user can get authorized to a specific group from Base station but can store only selective group ID in GL and can verify the GL by using HG which is hash of the Group ID.

step J1:

Complete step L1 of Login phase.

step J2:

User U_i enters G_{ID_x}, where G_{ID_x} is the group ID of group the user wants to join, or get access rights. Compute K_{UG_i}=J_i⊕R_u'⊕Tg_1.

Generate M_{x1}=$H(K_{UG_i}||ID_i||G_{ID_x}||Tg_1||HG_i)$. User sends a message to base station containing ID_i, M_{x1}, Tg_1, G_{ID_x}, HG_i.

$U_i \rightarrow BS :< ID_i, M_{x1}, Tg_1, G_{ID_x}, HG_i>$.

step J3:

Base station receive the message and verify $|Tg_1 - Tg_1^*|< \triangle Tg_1$, if its true go to next step BS generate K_{UG_i}=$H(ID_i||S)$⊕Tg_1. Then compute M_{x1}'=$H(K_{UG_i}||ID_i||G_{ID_x}||Tgi||HG_i)$.

step J4:

If M_{x1}=M_{x1}' then accept. If G_{ID_x} is present in BS access-list. HG_i'=HG_i⊕$H(G_{ID_x}||H(ID_i||S))$. Compute M_{x2}= $H(K_{UG_i}||HG_i'||Tg_1||Tg_2)$. Base station replies to User with message having ID_i, M_{x2}, Tg_2.

$BS \rightarrow U_i :< ID_i, M_{x2}, Tg_2>$

step J5:

User U_i verify $|Tg_2 - Tg_2^*|<\triangle Tg_2$, if true go to next step. HG_i'=HG_i⊕$H(G_{ID_x}||(r_i$⊕$w'))$, M_{x2}'=$H(K_{UG_i}||HG'||Tg_1||Tg_2)$. If M_{x2}=M_{x2}', then update GL_i, add G_{ID_x} to GL_i. Update HG_i to HG_i'.

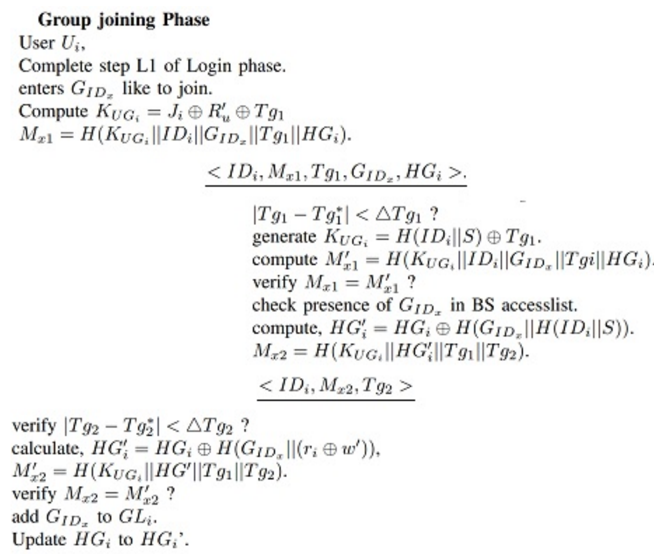

Fig. 7. Group Joining Phase between User and BS

6) **Authentication Phase:** Steps for Authentication are :

Step A1:

User selects ID_{sj} for a sensor node Sj. Selects Key_{ij} and $Token_{ij}$ of S_j from the Key-ID pair sent by the base station. User selects Random Nonce $RN_{U_{si}}$ and compute M_5=M_1⊕ $RN_{U_{si}}$, M_6=$H(M_5||ID_{sj}||RN_{U_{si}})$ and M_7=$E_{key_{ij}}(M_5$, $M_6, T_1, T_2, APM_i, RN_{U_{si}}, T_3)$, where T_3 is time stamp of sending request to sensor node by user. Then, User U_i sends the login message $< ID_i, ID_{sj}, M_7, Token_{ij}, T_3>$ to sensor node SN.

$U_i \rightarrow S_j :< ID_i, ID_{sj}, M_7, Token_{ij}, T_3>$

Step A2:

Sensor Node verifies $|T_3 - T_3^*|<\triangle T_3$, if verification holds go to next step.

Step A3:

Sensor node compute Key_{ij}'=$H(ID_i||ID_{sj}||MK_{sj}||K_j)$ and decrypt M_7 and get $M_5, M_6, T_1, T_2, APM_i, RN_{U_{si}}, T_3$.

Check $T_3?{=}T_3$ for validity of time. Calculate $M_6'{=}H(M_5||ID_{sj}||RN_{U_{si}})$. If $M_6{=}M_6'$ verifies then continue further. Calculate $M_1'{=}M_5{\oplus}RN_{U_{si}}$. Compute $C'{=}H(ID_i||ID_{sj}||T_1||T_2||APM_i||MK_{sj}||M_1')$.
step A4:

If $C'{=}Token_{ij}$, accept. Compute secret symmetric session key $SK_{u_i,s_j}{=}H(ID_i||ID_{sj}||RN_{U_{si}}||M_1)$.
Compute $M_8{=}E_{SK_{u_i,s_j}}(RN_{U_{si}})$, Where $RN_{U_{si}}$ is user random Nonce.
$S_j \to U_i : Ack(ID_i, ID_{sj}, M_8)$.
step A5:

User U_i Computes, $SK_{u_i,s_j}{=}H(ID_i||ID_{sj}||RN_{U_{si}}||M_1)$, Then decrypt message $RN_{U_{si}}'{=}D_{sk_{u_i,s_j}}(M_8)$. User also checks $RN_{U_{si}}{=}RN_{U_{si}}'$. If check is verified then Symmetric Session Key established as SK_{u_i,s_j}.

Authentication Phase
User U_i,
selects ID_{sj} for a sensor node Sj.
Select Key_{ij} and $Token_{ij}$ of S_j.
Select Random Nonce $RN_{U_{si}}$ and
compute $M_5 = M_1 \oplus RN_{U_{si}}$,
$M_6 = H(M_5||ID_{sj}||RN_{U_{si}})$ and
$M_7 = E_{key_{ij}}(M_5, M_6, T_1, T_2, APM_i, RN_{U_{si}}, T_3)$,

$< ID_i, ID_{sj}, M_7, Token_{ij}, T_3 >$

Sensor Node SN_j, verify $|T_3 - T_3^*| < \triangle T_3$?
compute $Key_{ij}' = H(ID_i||ID_{sj}||MK_{sj}||K_j)$ and
Decrypt M_7 using Key_{ij}'.
verify $T_3? = T_3$ from M_7.
Calculate $M_6' = H(M_5||ID_{sj}||RN_{U_{si}})$.
verify $M_6 = M_6'$?
Calculate $M_1' = M_5 \oplus RN_{U_{si}}$.
Compute $C' = H(ID_i||ID_{sj}||T_1||T_2||APM_i||MK_{sj}||M_1')$.
If $C' = Token_{ij}$, accept.
Compute,
$SK_{u_i,s_j} = H(ID_i||ID_{sj}||RN_{U_{si}}||M_1)$.
$M_8 = E_{SK_{u_i,s_j}}(RN_{U_{si}})$.

$Ack(ID_i, ID_{sj}, M_8)$

User U_i Computes,
$SK_{u_i,s_j} = H(ID_i||ID_{sj}||RN_{U_{si}}||M_1)$,
Decrypt M_8 using SK_{u_i,s_j}.
check $RN_{U_{si}} = RN_{U_{si}}'$?
Symmetric Session Key established as SK_{u_i,s_j}.

Fig. 8. Authentication Phase between User and SN

7) **Password change Phase:** In this phase user can change his password, by authorizing himself by providing ID, old password and biometric information.
Step C1:

User insert Smart Card in Card Reader, input ID_i', PW_i^{old}, imprint B_i' at fuzzy extractor and calculate $R_u'{=}Rep(B_i', Pu)$ and $K'{=}L_i{\oplus}R_u'$ then smart card verifies $V_i{=}?H(ID_i'||PW_i^{old}||K')$ If condition holds, go to next step.
Step C2:

Smart card calculate $w_i^{old}{=}H(ID_i'||PW_i^{old}||R_u')$ and $h_{r_i}{=} w_i^{old}{\oplus}r_i{=}H(ID_i||S)$. User inputs new password PW_i^{new}, then smart card computes $w_i^{new}{=}H(ID_i||PW_i^{new}||R_u')$, $r_i^{new}{=}w_i^{new}{\oplus}h_{r_i}$, $V_i^{new}{=}H(ID_i||PW_i^{new}||K')$ and $J_i^{new}{=} r_i^{new}{\oplus}w_i^{new}{\oplus}R_u'$ and replaces r_i, V_i, J_i with r_i^{new}, V_i^{new}, J_i^{new} respectively.

Password change Phase
insert Smart Card in Card Reader.
enter ID_i', PW_i^{old}, imprint B_i'.
calculate $R_u' = Rep(B_i', Pu)$ and
$K' = L_i \oplus R_u'$
verify $V_i = ?H(ID_i'||PW_i^{old}||K')$?
calculate $w_i^{old} = H(ID_i'||PW_i^{old}||R_u')$ and
$h_{r_i} = w_i^{old} \oplus r_i = H(ID_i||S)$.
User inputs new password PW_i^{new},
compute $w_i^{new} = H(ID_i||PW_i^{new}||R_u')$,
$r_i^{new} = w_i^{new} \oplus h_{r_i}$,
$V_i^{new} = H(ID_i||PW_i^{new}||K')$ and
$J_i^{new} = r_i^{new} \oplus w_i^{new} \oplus R_u'$ and
replace r_i, V_i, J_i with $r_i^{new}, V_i^{new}, J_i^{new}$ respectively.

Fig. 9. Password Change Phase at User-End

8) **Dynamic Node Addition Phase:** New node deployment in sensor networks is inevitable due to the loss of sensor nodes resulting from power exhaustion after weeks or months of operation. Some nodes may become com- promised and require replacement. We assume that one or more nodes must be deployed in a dynamic node addition phase. Let a new sensor node u be deployed during the dynamic node addition phase. Prior to its deployment, (during the pre-deployment phase),
Step D1:

the BS will preload a set of node parameters offline. This set contains (i) a unique node identifier ID_{su} of the node u, (ii) a hash function $H(\cdot)$ and (iii) its own master key MK_{su}.
Step D2:

After deployment, SN_u sends a message containing its own identity ID_{s_u}, the bootstrapping time T_u, and the encrypted information $E_{MK_{su}}(K_u, ID_{su}, T_u)$ using the master key MK_{su} to the BS:
$SN_u \Rightarrow BS :< ID_{su}, T_u, E_{MK_{su}}(K_u, ID_{su}, T_u)>$

Therefore, the dynamic node addition phase in our scheme is simple and efficient, and it does not require any involvement of the base station after deployment.

V. Security Analysis

In this section, we show that our scheme has the ability to tolerate various known attacks, which are discussed in the following subsections.

- **Stolen-verifier attack**
  It should be noted that our scheme does not require any verifier/ password table storage for password verifications. A network insider cannot obtain a users password because the BS and sensor nodes do not maintain any password/verifier table to validate a users login request. During the registration phase of our scheme, a user securely U_i calculates

$w{=}H(ID_i||PW_i||R_u)$. Because the extracted string value R_u is unique and only accessible to user U, it is computationally infeasible for the BS to retrieve PW_i from h( ) due to one-way property of the hash function w. Therefore, our scheme has the ability to prevent such an attack.

- **Many logged-in users with the same login-ID attack**
In general, if the systems that maintain the password table verify the user login, they can be vulnerable to attack. However, in our scheme, the BS and sensor nodes do not maintain any verifier table containing passwords for verification. In addition, no passwords are stored in the users smart card. At the time of login, a user U_j must have a valid smart card with the valid input tuple ID_i, PW_i, B_i.
Note that our scheme requires on-card computation for both password verification and login to the WSN, once the smart card is removed from the system, the login process is aborted. If two users U_i and U_j have the same password, still they will definitely have a different Bi, as Biometric impression are unique. As a result, even if two users have the same password, the problem of many logged-in users with the same login ID does not arise in our scheme. Thus, our scheme resists the many logged-in users with the same login-ID attack.

- **Resilience against node capture attack**
We evaluate the ability of our scheme to tolerate compromised nodes in the network. Let $P_e(c)$ denote the probability that an adversary compromises a fraction of total secure communications by capturing c number of sensor nodes in the network. If $P_e(c)$ = 0, we classify our user access control scheme as unconditionally secure against node capture attack. If an attacker captures a sensor node, he/she is able to discern the master key along with other information from its memory because the sensor nodes are not equipped with tamper-resistant hardware. However, each node is given a unique randomly generated master key prior to its deployment and each sensor node establishes a distinct secret session key with a user. Thus, the attacker can only respond with false data to a legitimate user by capturing a sensor node from which the user wants to access data. However, other non-captured sensor nodes can still communicate real-time data to legitimate users with 100 percent, secrecy. As a result, the compromise of a sensor node does not lead to a compromise in any other secure communication between the user and the non-captured sensor node in the network; therefore, our scheme provides unconditional security against node capture attack.

- **Masquerade attack**
In our scheme, an illegal user cannot fabricate the fake login request message to convince the BS that it is a legal login request in the login phase. At the time of login, the user must insert his/her smart card into a card reader and then to provide his/her user ID ID_i, password PW_i and access group ID G_{id_j}. U_i inputs ID_i' and PW_i', G_{id_x} and imprints B_i' at fuzzy extractor and calculates $R_u'{=}Rep(B_i', P_u)$, computes $K'{=}L{\oplus}R_u'$ by the stored values in the smart card. Then the smart card verifies $V{=}?H(ID_i||PW_i||K)$. If this verification passes , then smart card calculates M3 and user U_j sends the login request message $U_i \rightarrow BS :< ID_i; M3; T1>$. As a result, the attacker does not have the ability to create a fake login request message on behalf of the original user U_i. Thus, our scheme resists this type of attack.

- **Replay attack**
In this scenario, an attacker may try to pose as a valid user logging into the BS by sending messages that were previously transmitted by a legal user. However, our scheme utilizes a current system timestamp during the login and authentication phases. A comparison of the previous timestamp with the current timestamp of the receiver system withstands these replay attacks because the expected time interval for the transmission delay is very short. Moreover, in the login phase, the user sends the message $< ID_i, M_3, T_1>$ to the BS. Because the attacker cannot change the T_1 the attacker also cannot change the value of T_1. Thus, an attacker does not have the ability to successfully replay previously used messages during the login and authentication phases. As a result, our scheme resists the replay attack.

- **Privileged-insider attack**
Note that during the registration phase of our proposed scheme, the user U_i does not send his/her password PW_i in plaintext. The user U_i send a masked password w, where $w{=}h(ID_i||PW_i||R_u)$ through a secure channel to the BS. Without knowing the secret value R_u (which is only known to the user U_i and can only be produced by Fuzzy extractor when Biometric impression is given to it), it is computationally infeasible to retrieve PW_i from 'w' due to the one-way property of the hash function $H(\cdot)$. A privileged insider at the BS does not have the ability to know the password PW_i of user.

- **Denial-of-service attack**
After deployment, the sensor node in our scheme initially sends a message to the BS to inform its own bootstrapping time. But after that there is no communication between the BS and SN. So we safeguard the sensor node from draining away the energy from fake requests and we dont make the sensor node to store any information about each request. But authenticate a request from a user on the go, which reduces the memory requirement. At the time of authentication, after receiving the request message from

user U_i, the sensor node SN_j sends an acknowledgment to the user after successful authentication. If an attacker blocks the messages from reaching the BS and sensor nodes, the BS and sensor node will know about the malicious dropping of these control messages. Therefore, the denial-of-service attack is not possible in our scheme because an acknowledgment is sent to user Ui at the end of user authentication.

- **Offline Password Guessing Attack**
It is special case of stolen smart card Attack. In our scheme, it is hard to derive $Key_{ij}{=}H(ID_i||ID_{sj}||MK_{sj}||K_j)$ and $K_{ui}{=}J_i \oplus R'_{ui} \oplus T_1$, the relation for helping guess the password is not available to the adversary. Therefore, it is impossible for an adversary to get the right password because of the uncertainty of these different $MK_{sj}, K_j and R_{ui}$. As R_{ui} is extracted string from the Fuzzy extractor.

VI. Formal security verification of our scheme using AVISPA back-ends

In this section, we only simulate our scheme for the formal security analysis. We do not simulate communication, computation and energy cost of our scheme, since these are evaluated extensively theoretically in this paper. Through the simulation results using the widely-accepted AVISPA tool we show that our scheme is secure against passive and active attacks including the replay and man-in-the-middle attacks. For this purpose, we first describe in brief the AVISPA tool, implement our scheme in the high level language, called HLPSL and simulate the implemented protocol to show that our scheme is secure.

The AVISPA is an acronym for Automated Validation of Internet Security Protocol and Applications, is a Push-button security protocol analyzer, and supports the specification of security protocols and properties by means of a modular and expressive specification language. It integrates different back-ends implementing a variety of automatic analysis techniques for protocol falsification by finding an attack on the input protocol, and abstraction-based verification methods both for finite and infinite numbers of sessions. The user interaction is facilitated by a web interface that is an easy way to use AVISPA tool without installing any other software to support it. The next section describes the architecture of AVISPA tool.

A. AVISPA tool

The architecture of the AVISPA tool is shown in Figure 3. HLPSL-High Level Protocol Specification Language provides a high level of abstraction and has many features that are common to most protocol specifications such as intruder models and encryption primitives. The Intermediate Format (IF), the language into which HLPSL specifications are translated, is a lower level language at an accordingly lower abstraction level. HLPSL specifications are translated into the IF by the HLPSL2IF translator. These translations in turn, serve as input to the various backends. These are analysis tools of the AVISPA tool-set.

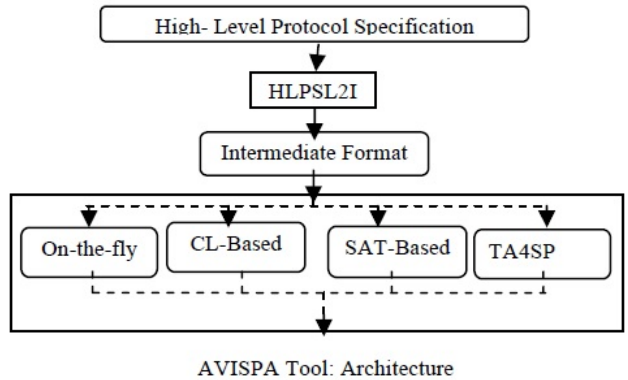

Fig. 10. Avispa

The High Level Protocol Specification Language (HLPSL): is an expressive language for modelling communication and security protocols. HLPSL draws its semantic roots from Lamports Temporal Logic of Actions (TLA). TLA is an elegant and powerful language which lends itself well to specifying concurrent systems. The development of HLPSL was thus undertaken with the following design objectives:

It must provide a convenient, human readable and easy to use language yet be powerful enough to support the specification of modern Internet security protocols. To achieve, HLPSL has been defined in such a way as to closely resemble a language for defining guarded transitions within a state-transition system and is equipped with constructs which allow the modular specification of protocols.

It must be consists of a formal semantics. For this,HLPSL has been based on Lamports TLA and its semantics is given by a translation to a subset of TLA

It must amenable to automated formal analysis.This is achieved by a translation of HLPSL into the Intermediate Format.

Supports symmetric and asymmetric keys, nonatomic keys, key-tables, Diffie-Hellman keyagreement, hash functions, algebraic functions, typed and untyped data, etc.

Supports security properties, different forms of authentication and secrecy.

The intruder model is made by the channel(s) over which the communication takes places.

Role based language, a role for each (honest) agent and Parallel and sequential composition glue roles together.

The HLPSL2IF translator automatically translates a HLPSL protocol specification provided by the user into an IF specification, which is then given as input to the different back-ends of the AVISPA tool. Hence, the main goal in the design of the IF was to provide a low-level description of the protocol that is suitable for automatic analysis and yet this format should be independent from the analysis methods employed by the various back-ends. The Back-Ends are used to provide protocol falsification, bounded and unbounded verification. OFMC (The On-the-fly Model-Checker) employs several symbolic techniques to explore the state space in a demand-driven way. CL-AtSe (Constraint-Logic-based Attack Searcher) applies constraint solving with simplification heuristics and redundancy elimination techniques. It provides a translation from any security protocol specification written as transition relation in the IF, into a set of constraints which can be effectively used to find attacks on, protocols. SATMC (SAT-based Model-Checker), builds a propositional formula which is then fed to a state-of-the-art SAT solver and any model found is translated back into an attack. TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols) approximates the intruder knowledge by using regular tree languages and rewriting to produce under and over approximations. Specifying our scheme : We have implemented our scheme in the HLPSL language. In this implementation, we have three basic roles: the sensor node SN, the BS and the user U. We have also defined the session and environment in our scheme. Figure below illustrates the role specification for user U in HLPSL. During the registration phase, U sends the message IDi.Wi'.RMui' securely to the BS with the Snd() operation. The type declaration channel (dy) indicates the channel for the DolevYao threat model (as described in our threat model in previous sections) Uj then waits for the smart card containing the secure information in the xor(H(IDi.PWi.Ru),H(IDi.S)).H.RMui' from the BS from the Rcv() operation. The intruder will have the ability to intercept, analyze, and/or modify messages transmitted over the insecure channel. During the login phase, U sends the login request message .¡ IDi;M3; T1 ¿ where M3= Ekui (Gidx ;M1;M2;RNui) to the BS. In reply, the BS sends the message :¡ IDi;M4; T2 ¿ to U. During the authentication phase, U finally sends the authentication request message :¡ IDi; IDsj ;M7; Tokenij ; T3 ¿ to the sensor node SNi.

Figure above shows the role specification for the BS in the HLPSL language. During the post-deployment phase, the BS receives the message :¡ IDsj ; Tj ;EMKsj (Kj ; IDsj ; Tj) ¿ from the sensor node SNi. During the registration phase after receiving the message ¡IDi,w,Ru¿ securely from the user U, the BS securely sends the smart card containing the information in the message ¡ ri, H(.) ,Ru¿ to the user U.

In Figure above, we have implemented the role specification for the sensor node SN in the HLPSL language. In the post-deployment phase, the sensor node SN sends the message :¡ IDsj ; Tj ;EMKsj (Kj ; IDsj ; Tj) ¿ to the BS. During the authentication phase, the sensor node receives the authentication request message :¡ IDi; IDsj ;M7; Tokenij ; T3 ¿ from the user Uj. Where M7 = Ekeyij (M5;M6; T1; T2;APMi;RNUsi ; T3).

Witness (A, B, ID, E) declares for a (weak) authentication property of A by B on E, declares that agent A is witness for the information E; this goal will be identified by the constant ID in the goal section. Request (B, A, ID, E) demands a strong authentication property of A by B on E, declares that agent B requests a check of the value E; this goal will be identified

```
role user(U,BS,SN : agent,
       MKsj : symmetric_key,
       MKui : symmetric_key,
       H : hash_func,
       Snd, Rcv : channel(dy))
played_by U
def=
local State : nat,
IDi, Wi, APMi, RMui, Pu, PWi, KUi,
Rui, Keyij, IDsj,  Kj, GIdx, RNuj,
M1,M2,GIDx,Ru,S,T1, T2 :text

const user_basestation, user_sensornode,
subs1, subs2, subs3, subs4, subs5 : protocol_id
init State := 0
transition
1. State = 0 /\ Rcv(start) =|>

%registrationPhase
State' := 1 /\ Wi' := H(IDi.PWi.Ru)
          /\ Snd(U.BS.{IDi.Wi'.RMui'}_MKui)
          /\ RMui' := new()

2. State = 1 /\ Rcv(BS.U.{xor(H(IDi.PWi.Ru),H
(IDi.S)).H.RMui'}_MKui) =|>
              %smart card values

%loginPhase
State' := 2
       /\secret({Kj},subs1,{SN,BS})
       /\secret({MKsj},subs2,{SN,BS})
       /\secret({RMui'},subs3,{U,BS})
       /\secret({APMi,GIdx},subs4,{U,BS})
                   /\secret({PWi,Pu},subs5,U)

                   /\T1' := new()
                   /\ M1' := xor(H(IDi.PWi.Ru),RMui')
                   /\ M2' := H(xor(H(IDi.PWi.Ru),H
(IDi.S)).RMui'.GIDx.T1')
                   /\KUi' := xor(H(IDi.S).T1')
                   /\Snd(U.BS.{GIdx.M1'.M2'.RMui'}
_KUi'.T1')
                   /\witness(U,BS,user_basestation,T1')

       3. State = 2 /\ Rcv(BS.U.{IDi.IDsj.Keyij.H
(IDi.IDsj.T1'.T2'.APMi.MKsj.M1').APMi.T1'.T2'}_KUi.T2')
=|>

%authenticationPhase
              State' := 3 /\ KUi' := xor(H(IDi.S).T1')
                   /\ Snd(U.SN.IDi.IDsj.
{IDi.IDsj.Rui.GIdx.T1'}_Keyij.H
(IDi.IDsj.T1'.T2'.APMi.MKsj.M1'))
                   /\ witness(U,SN,user_sensornode,T1')
end role
```

User

Fig. 11. User

```
role basestation(BS,SN,U :agent,
MKsj : symmetric_key,
MKui : symmetric_key,
H : hash_func,
Snd, Rcv :channel(dy))
played_by BS
def=

local State : nat,
Wi, RMui, Rui, Keyij,  T2, APMi, GIdx,
Pu, PWi, KUi,Ri,Ru,S,Tokenij, GIDx :
text,
IDsj, IDi, Kj, Tj, T1, M1 : text
const
sensornode_basestation,user_basestation,
subs1, subs2, subs3, subs4, subs5 :
protocol_id
init State := 0
transition

%postDeploymentPhase
1. State = 0 /\ Rcv(SN.BS.IDsj.Tj.
{Kj.IDsj.Tj}_MKsj) =|>
State' := 1 /\ Keyij' := H
(IDi.IDsj.MKsj.Kj)

%registrationPhase
2. State = 1 /\ Rcv(U.BS.{IDi.H
(IDi.PWi.Ru).RMui'}_MKui) =|>
%user registration through secure
channel
State' := 2
       /\ Ri' := xor(H(IDi.PWi.Ru),H
(IDi.S))
       /\ Snd(BS.U.{Ri'.H.RMui'}_MKui)
       /\secret({Kj},subs1,{SN,BS})
       /\secret({MKsj},subs2,{SN,BS})
       /\secret({RMui'},subs3,{U,BS})
       /\secret({APMi,GIdx},subs4,
{U,BS})
       /\secret({PWi,Pu},subs5,U)
       /\request
(SN,BS,sensornode_basestation,Tj)

%loginPhase
3. State = 2 /\ Rcv(U.BS.{GIdx.M1'.H(xor
(H(IDi.PWi.Ru),H
(IDi.S)).RMui'.GIDx.T1').RMui'}
_KUi'.T1') =|>
State' := 3      /\T2' := new()
/\KUi' := xor(H(IDi.S).T1')
/\Tokenij' := H
(IDi.IDsj.T1'.T2'.APMi.MKsj.M1')
/\Snd(BS.U.
{IDi.IDsj.Keyij.Tokenij'.APMi.T1'.T2'}
_KUi.T2')
/\request(U,BS,user_basestation,T1')
end role
```

Base Station

```
role sensornode(SN,BS,U : agent,
     MKsj : symmetric_key,
     H : hash_func,
     Snd,Rcv : channel(dy))
played_by SN
def=

local State : nat,
IDsj, Tj, Kj :text,
IDi, APMi, GIdx, Wi, RMui,
T1,T2,Rui,Keyij,Pu,PWi,KUi,M1 :
text
const
sensornode_basestation,sensornode_u
ser,user_sensornode,
subs1, subs2, subs3, subs4, subs5 :
protocol_id

init State := 0
 transition
 1. State = 0 /\ Rcv(start)=|>

%postDeploymentPhase
     State' := 1 /\ T1' := new()
/\secret({Kj},subs1,{SN,BS})
/\secret({MKsj},subs2,{SN,BS})
/\secret({RMui},subs3,{U,BS})
/\secret({APMi,GIdx},subs4,{U,BS})
/\secret({PWi,Pu},subs5,U)
/\Snd(SN.BS.IDsj.Tj.{Kj.IDsj.Tj}
_MKsj)
/\witness
(SN,BS,sensornode_basestation,T1')

%authenticationPhase
 2. State = 1 /\ Rcv(U.SN.IDi.IDsj.
{IDi.IDsj.Rui.GIdx.T1'}_Keyij.H
(IDi.IDsj.T1'.T2'.APMi.MKsj.M1'))
=|>
     State' := 2 /\ request
(U,SN,user_sensornode,T1')
```

Sensor Node

Fig. 13. Sensor Node

```
role environment()
def=
const sn, bs, u : agent,
     mksj : symmetric_key,
     mkui : symmetric_key,
     h : hash_func,
rpwi,rui,kui,kj,rnui,tj,t1,t2,apmi,gi
di,kbs,snj,ui : text,
sensornode_basestation,
sensornode_user, user_basestation,
user_sensornode,subs1, subs2, subs3,
subs4, subs5 : protocol_id

intruder_knowledge =
{u,bs,sn,h,ui,snj,ui}
composition
%session(sn, bs, u, mksj, mkui, h)
session(sn, u, bs, mksj, mkui, h)/\

session(u, sn, bs, mksj, mkui, h)/\

session(u, sn, bs, mksj, mkui, h)
end role

goal
secrecy_of subs1
secrecy_of subs2
secrecy_of subs3
secrecy_of subs4
secrecy_of subs5
authentication_on user_basestation
authentication_on sensornode_user
authentication_on user_sensornode
authentication_on
sensornode_basestation
end goal
environment()
```

Environment

Fig. 14. Environment

by the constant ID in the goal section. The intruder is always denoted by i.

Finally, the specifications in the HLPSL language for the role of session, goal and environment are specified in Figs. 7 and 8. In the session segment, all of the basic rolesalice, server and bobare instanced with concrete arguments. The toplevel role (environment) is always defined in the specification of the HLPSL language. This role contains the global constants and a composition of one or more sessions, where the intruder may play some roles as legitimate users. The intruder also participates in the execution of protocol as a concrete session.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-
computation/./tempdir/workfilesIrO
ii.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 5.60s
  visitedNodes: 552 nodes
  depth: 9 plies
```

OFMC Result

Fig. 15. OFMC

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/avispa/web-interface-
computation/./tempdir/workfilesIr
Oii.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed    : 48 states
  Reachable   : 15 states
  Translation: 0.19 seconds
  Computation: 0.00 seconds
```

CL-ATSe

Fig. 16. CL-Atse

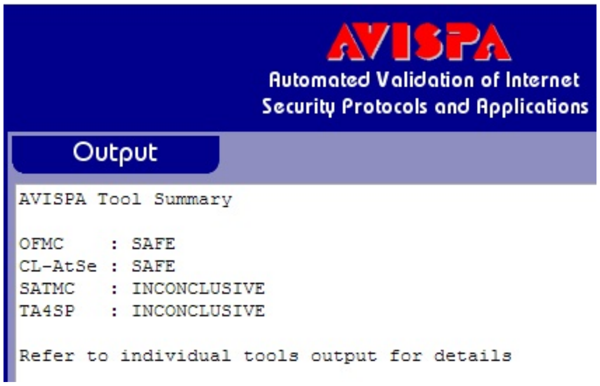

Fig. 17. Avispa output

VII. Performance Analysis

This section compares the performance of our scheme with rel- evant existing access control schemes such as Mahmud et al.s scheme (Mahmud and Morogan, 2012), Wang et al.s scheme (Wang et al., 2006) and Le et al.s scheme (Le et al., 2009).

Table Time complexity of various operations in terms of t_{mul}.

$t_{ecm} \approx 1200t_{mul}$	$t_{sigver} \approx 2405.36t_{mul}$	$t_i \approx 3t_{mul}$
t_{add} is negligible	$t_h \approx 0.36t_{mul}$	$t_{enc} \approx 0.15t_{mul}$
$t_{dec} \approx 0.15t_{mul}$	$t_{ecenc} \approx 2405t_{mul}$	$t_{ecdec} \approx 1205t_{mul}$
$t_{mac} \approx t_h$	$t_{siggen} \approx 1204.36t_{mul}$	$t_{eca} \approx 5t_{mul}$

Fig. 18. Time complexity of various operations

We have used the notations for computational cost comparisons between our scheme and other schemes as t_{ecm}, t_{eca}, t_i, t_{add}, t_{mul}, t_h, t_{enc}, t_{dec}, t_{ecenc}, t_{ecdec}, t_{mac} , t_{siggen} and t_{sigver} denote the time taken for performing one ECC point multiplication over a finite field GF (2^{163}), an ECC point addition over a finite field GF (2^{163}), a modular inverse over a finite field GF (2^{163}), a modular addition over a finite field GF (2^{163}), a modular multiplication over finite field GF (2^{163}), a hashing operation $H(\cdot)$, an AES encryption, an AES decryption, an ECC encryption over a finite field GF (2^{163}), an ECC decryption over a finite field GF (2^{163}), a MAC operation, an ECC signature generation over finite field GF (2^{163}), and an ECC signature verification over a finite field GF (2^{163}), respectively. For the sake of simplicity, we consid- ered the time taken for one MAC operation as that for one hashing operation.

We have compared the computational complexity using both formulated results in above Table for different phases: the registration, login and authentication phases of Le et al. (2009), Wang et al. (2006), Mahmud and Morogan (2012), and our scheme. It is clear that, compared with the other existing schemes, the computational cost of our scheme is significantly lower. Thus, our scheme is more suitable for resource-constrained sensor nodes.

VIII. Conclusion

In this paper we have presented a scheme which is secure, elegant, have low memory requirement, provide direct communication between user and the sensor node without passing through base station. It provide an alternate way of providing user access control and also increase the scalability of the sensor network because of low memory requirement on sensor node to serve multiple group users. According to the security analysis, it is obvious that our scheme is secure enough to withstand all possible attacks.

Acknowledgment

The authors would like to thank Dr. Ashok Kumar Das, Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad.

References

[1] Alemdar, H., Ersoy, C., 2010. Wireless sensor networks for healthcare: a survey. Computer Networks 54 (15), 26882710.

[2] Ameen, M. Al., Liu, J., Kwak, K., 2012. Security and privacy issues in wireless sensor networks for healthcare applications. Journal of Medical Systems 36 (1), 93101.

[3] Ankita Chaturvedi, Dheerendra Mishra, and Sourav Mukhopadhyay, 2013. Improved Biometric-Based Three-factor Remote User Authentication Scheme with Key Agreement Using Smart Card, A. Bagchi and I. Ray (Eds.): ICISS 2013, LNCS 8303, pp. 63-77, 2013

[4] Chatterjee, S. et al., A novel and efficient user access control scheme for wireless body area sensor networks. Journal of King Saud University Computer and Information Sciences (2013),http://dx.doi.org/10.1016/j.jksuci.2013.10.007

[5] D. Dolev, A. Yao, On the security of public key protocols, IEEE Transactions on Information Theory, 29 (2) (1983), pp. 198208

[6] Das, A.K., 2009. An unconditionally secure key management scheme for large-scale heterogeneous wireless sensor networks. In: First International Conference on Communication Systems and Networks (COMSNETS 2009), pp. 110

[7] Das, A.K., 2012. A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks. International Journal of Information Security 11 (3), 189211.

[8] Das, A.K., 2013. A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications. Networking Science 2 (1-2), 1227.

[9] Das, A.K., Sharma, P., Chatterjee, S., Sing, J.K., 2012b. A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. Journal of Network and Computer Applications 35 (5), 16461656.

[10] Das, M.L., 2009. Two-factor user authentication in wireless sensor networks. IEEE Transactions on Wireless Communications 8 (3), 10861090.

[11] E. Klaoudatou, E. Konstantinou, G. Kambourakis, S. Gritzalis, A survey on cluster-based group key agreement protocols for WSNs, IEEE Communications Surveys and Tutorials, 13 (3) (2011), pp. 429442

[12] Ghasemzadeh, H., Jafari, R., 2011. Physical movement monitoring using body sensor networks: a phonological approach to construct spatial decision trees. IEEE Transactions on Industrial Informatics 7 (1), 6677.

[13] He, D., Bu, J., Zhu, S., Chan, S., Chen, C., 2011. Distributed access control with privacy support in wireless sensor networks. IEEE Transactions on Wireless Communications 10, 34733481.

[14] Kwak, K.S., Ameen, M.A., Kwak, D., Lee, C., Lee, H., 2009. A study on proposed IEEE 802.15 WBAN MAC Protocols. In: Proceedings of ICCIT09.

[15] Li, M., Lou, W., Ren, K., 2010. Data security and privacy in wireless body area networks. IEEE Wireless Communications, 5158.

[16] Liang, X., Li, X., Shen, Q., Lu, R., Lin, X., Shen, X., Zhuang, W., 2012. Exploiting prediction to enable Secure and Reliable routing in wireless body area networks. In: INFOCOM 2012, pp. 388396.

Table Comparison of computational costs for different phases in different access control schemes.

Phase	User or Node	Le et al. (2009)	Wang et al. (2006)	Mahmud and Morogan (2012)	Ours
Registration	U_j	-	-	-	$2t_h$
	BS	$2t_{ecm} + t_{siggen}$	$t_h + 3t_{ecm} + t_{mul} + t_{eca}$	t_h	t_h
	SN_i	-	-	-	-
Login + Authentication	U_j	$t_h + t_{sigver} + t_{mac}$	$t_{ecm} + 2t_{mac}$	$t_h + t_{sigver}$	$5t_h + 2t_{dec} + 2t_{enc}$
	BS	$2t_{sigver} + 2t_{mac} + 2t_h$	-	-	$2t_h + t_{dec} + t_{enc}$
	SN_i	$3t_{mac} + t_h$	$t_{eca} + 3t_{ecm} + t_h + 2t_{mac}$	$2t_h + t_{siggen} + t_{sigver}$	$4t_h + t_{dec} + t_{enc}$
	Total Cost	$4t_h + 2t_{ecm} + 4t_{hsigver} + 6t_{mac}$	$2t_h + 7t_{ecm} + t_{mul} + 2t_{eca} + 4t_{mac}$	$4t_h + 2t_{siggen} + 2t_{sigver}$	$14t_h + 8t_{enc}/t_{dec}$

Fig. 19. Comparison of computational costs

[17] Otto, C., Milenkovic, A., Sanders, C., Jovanov, E., 2006. System architecture of a wireless body area sensor network for ubiquitous health monitoring. Journal of Mobile Multimedia 1 (4), 307326.

[18] Singelee, D., Latre, B., Braem, B., Peeters, M., Soete, M.D., Cleyn, P.D., Preneel, B., Moerman, I., Blondia, C., 2008. A secure crosslayer protocol for multi-hop wireless body area networks. In: Proceedings of 7th International Conference on Ad-hoc, Mobile and Wireless Networks (ADHOC-NOW 2008), LNCS 5198.

[19] Sarkar, P., 2010. A simple and generic construction of authenticated encryption with associated data. ACM Transactions on Information and System Security 13 (4), 33.

[20] Stallings, W., 2003. Cryptography and Network Security: Principles and Practices, 3rd ed. Prentice Hall.

[21] Wang, H., Sheng, B., Tan, C.C., Li, Q., 2008. Comparing symmetrickey and public-key based security schemes in sensor networks: a case study of user access control. In: Proceedings of 28th International Conference on Distributed Computing Systems.

[22] Wu, S., Chen, K., 2012. An efficient key-management scheme for hierarchical access control in e-medicine system. Journal of Medical Systems 36 (4), 23252337.

[23] Zois, D.S., Levorato, M., Mitra, U., 2012. A POMDP framework for heterogeneous sensor selection in wireless body area networks. In: INFOCOM 2012, pp. 26112615.