Journal of King Saud University – Computer and Information Sciences (2014) 26, 181–201

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

# A novel and efficient user access control scheme for wireless body area sensor networks

Santanu Chatterjee [a], Ashok Kumar Das [b,*], Jamuna Kanta Sing [c]

[a] Research Center Imarat, Defence Research and Development Organization, Hyderabad 500 069, India
[b] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad 500 032, India
[c] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700 032, India

Received 20 February 2013; revised 13 August 2013; accepted 12 October 2013
Available online 26 October 2013

**KEYWORDS**

Wireless body area sensor networks;
User access control;
Authentication;
Security;
ECC;
AVISPA

**Abstract** Wireless body area networks (WBANs) can be applied to provide healthcare 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, and a password. An elliptic curve cryptography-based public key cryptosystem is used to ensure that a particular legitimate user can only access the information for which he/she is authorized. We show that our scheme performs better than previously existing user access control schemes. Through a security analysis, we show that our scheme is secure against possible known attacks. Furthermore, through a formal security verification using the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool, we show that our scheme is also secure against passive and active attacks.



## 1. 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

* Corresponding author. Tel.: +91 40 6653 1506; fax: +91 40 6653 1413.
E-mail addresses: santanu@rcilab.in (S. Chatterjee), iitkgp.akdas@gmail.com, ashok.das@iiit.ac.in (A.K. Das), jksing@ieee.org (J.K. Sing).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

1319-1578 © 2013 King Saud University. Production and hosting by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jksuci.2013.10.007

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 network 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 network. This control device can also be used as a gateway between 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 networks 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 providing 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 concerned 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 electronics of a WBAN sensor node are designed to detect and transmit low frequency and low amplitude physiological signals. The sensor node hardware requires a wireless link (AM152100 IC) from an AMI semiconductor used for MICS 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-

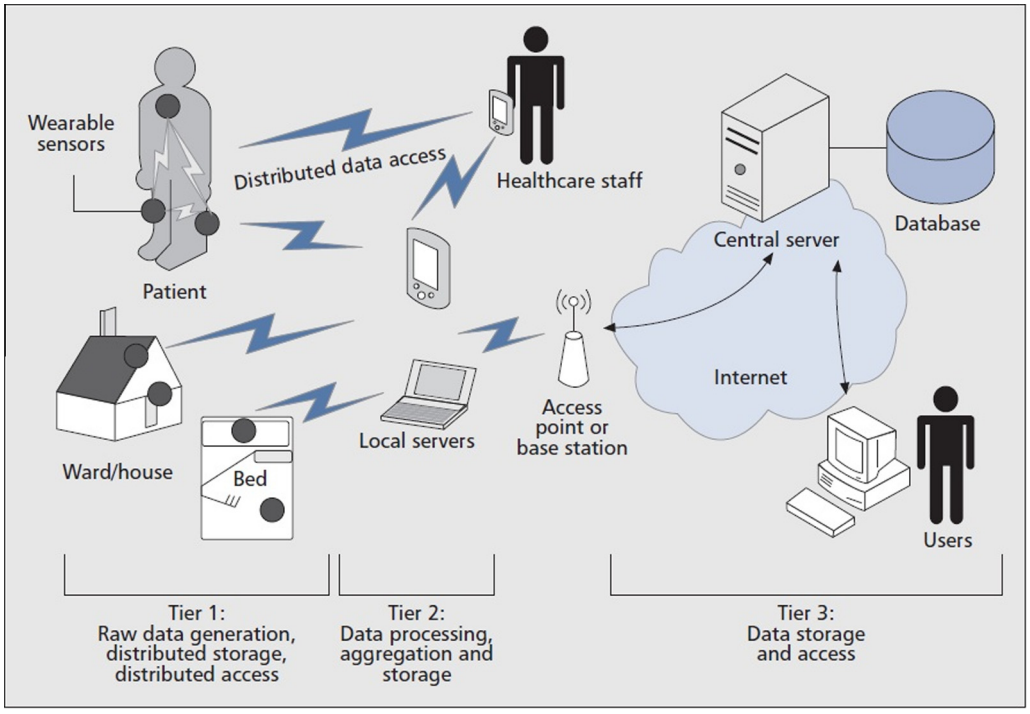



**Figure 1** A general three-tier architecture of WBAN (Li et al., 2010).

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.

### 1.1. 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 permitted. In healthcare applications, monitoring patient's conditions by the expert doctors is very essential. Thus, real-time data sensed by the sensors in a patient's body can be monitored 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 necessary action by instructing the nurses/medical staffs in the hospital 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 prominent research field.

### 1.2. Threat model

Based on Das ML (2009), we apply the Dolev–Yao threat model (Dolev and Yao, 1983) for our scheme, in which two parties (nodes) communicate over an insecure channel. We adopt a similar threat model where the channel is insecure and the end points (sensor nodes) cannot generally be trustworthy. Finally, we assume that an attacker can eavesdrop on all traffic, inject packets and reply to previously delivered messages. The base station (medical server) in our scheme is assumed to be trustworthy and impervious to attack. Due to cost constraints, the sensors are not equipped with tamper resistant hardware; if an attacker compromises any sensor from a patient's body, he/she can exact all cryptographic information, including the key materials, data and code stored on that node. Similar to Das et al. (2012b), we assume that the compromised (captured) nodes can be detected and the base station (medical server), users, and sensor nodes know the IDs of the compromised nodes. As a result, the base station (medical server) alerts the users to the compromised sensor nodes in the network.

### 1.3. Our contributions

This paper proposes a new password and group-based user access control scheme in wireless body area networks for health care applications. Our scheme has the following important properties:

- It provides password and group-based user authentication depending on the access rights provided for the genuine users in WBANs.
- It provides better security compared with the other related user access control schemes because it supports mutual authentication between the user and the base station and sensor node, resists denial-of-service, privileged-insider, smart card breach and man-in-the-middle attacks.
- It supports dynamic node additions after the initial deployment of nodes in the network. It also supports new node deployment for new patients and does not require updated information from the user's smart card.
- It supports a local change to the user's password without help from the base station (medical server).
- It establishes a secret session key between the user and a sensor node so that the same key can be used for future secret communication of real-time data inside the WBAN.
- Through formal security verification using the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool, we show that our scheme is also secure against passive and active attacks such as replay and man-in-the-middle schemes.

### 1.4. Organization of the paper

The rest of this paper is organized as follows. In Section 2, we review the existing related works on user access control in WSN as well as works on security in wireless body area networks. In Section 3, we propose a novel ECC-based user access control scheme in WBANs for healthcare and patient monitoring applications. In Section 4, we analyze the functionality and security properties of our proposed scheme through the informal and formal security analysis. In Section 5, we simulate our proposed scheme using the widely-accepted AVISPA tool. We show that our scheme is secure against passive and active attacks. In Section 6, we compare the performance of our scheme with other related schemes. We conclude the paper in Section 7.

## 2. Related work

This section briefly discusses the existing related user access control schemes that are currently proposed in resource-constrained wireless sensor networks.

We use elliptic curve cryptography (ECC) for our proposed user access control scheme for a wireless body area network. RSA (Rivest et al., 1978) may also be used to authenticate external users and Diffie and Hellman (1976) over DLP (discrete logarithm problem) used to establish shared keys between external users and sensor nodes in the network. However, the evaluation of a 1024-bit modular exponentiation for the DLP of the form $2^x$ (where $x$ is at least 160 bits) requires more than 50 s (Malan et al., 2004; Watro et al., 2004) on both MICA1 and MICA2 motes (Atmel Corporation, 2010). In Gura et al. (2004), Gura et al. implemented the assembly language for ECC and RSA on the Atmel ATmega 128 processor (Atmel Corporation, 2010), and they showed in their implementation that a 160 bit-point multiplication of ECC required 0.81 s, whereas 1024-bit RSA public and private key operations required 0.43 s and 10.99 s, respectively. Compared with RSA, ECC can achieve the same level of security with a

smaller size key. For example, a 160-bit ECC provides comparable security to a 1024-bit RSA and a 224-bit ECC provides the comparable security of a 2048-bit RSA (Rivest et al., 1992). It was noted in Carman et al. (2000) that the transmission energy consumption rates in wireless sensor networks are over three orders of magnitude greater than the energy consumption rates for computing. Therefore, the packet size and the number of packets in the transmission play a crucial performance role in designing an access control protocol in sensor networks. If a node is preloaded with the certificate by the base station, then the verifying RSA signature in the certificate takes less time than the ECC signature verification in the certificate because the signature will be generated offline by the base station prior to the deployment of sensor nodes in the target field. However, compared with a 1024-bit RSA signature (Rivest et al., 1978), an ECC-based signature (Johnson and Menezes, 1999; Liao and Shen, 2006) in the certificate, will only require a 320-bit signature when a 160-bit ECC is used in the proposed scheme. This motivates us to use ECC instead of RSA in our proposed access control scheme so that we can achieve greater energy and bandwidth savings. Our scheme uses symmetric key cryptographic techniques along with ECC to achieve communication and computational efficiency.

Wireless body area networks (WBANs) are envisioned to provide health care and patient monitoring applications in the near future. This paper addresses the importance of secure patient data acquisition for different types of users. The proposed authentication scheme consists of multiple phases that involve the users, the medical server (base station) and the sensors. The users' access is controlled through the use of binary mask value assigned to each user during the registration phase. Exchanged messages among parties are encrypted and signed using elliptic curve cryptography (ECC). The simulation of the proposed solution has been conducted through the use of the widely accepted AVISPA tool to evaluate the method against various known attack scenarios. The formal and informal security analyses show the protocol's resilience to known security attacks.

Wang et al. (2008) split the access control process into a local authentication conducted by a group of sensors physically close to a user and a remote authentication based on the endorsement of the local sensors. They implemented the access control protocol on a test bed of TelosB motes (Atmel Corporation, 2010). Based on ECC, they provided the local authentication. By using certificate-based authentication, the user access was verified by the sensor nodes.

He et al. (2011) proposed a distributed privacy preserving access control scheme for WSNs. They identified the characteristics of a single-owner multi-user sensor network and the requirements of a distributed privacy preserving access control. Their scheme was based on a ring signature technique. The user initially registers with a network owner. The network owner then divides all users into groups. The same group has the same access privilege. The network owner maintains a group access list pool that contains the identity and other information of each group, and access control is provided based on the group.

Wen et al. (2011) proposed a user access control scheme for a wireless multimedia sensor network. In this scheme, an authorized user can access the real time multimedia data. Their proposed scheme used Chinese Remainder Theorem-based group rekeying.

Li et al. (2010) discussed various practical issues required to fulfill the security and privacy requirements in WBANs. They explored the relevant security solutions in sensor networks and WBANs while analyzing various applications. They proposed an attribute-based encryption for achieving fine-grained access control. This is a one-to-many encryption method where the cipher text is only readable by a group of users that satisfy a certain access policy.

Mahmud and Morogan (2012) proposed an identity-based user authentication and access control protocol based on an identity-based signature (IBS) scheme. They used an ECC-based digital signature algorithm (DSA) for signing and verifying a message. At initialization, sensor nodes and users were registered to a base station and group identity and user access rights were also provided by the base station. User revocation was implemented through the expiration of user access time as assigned by the base station at the time of registration. The authenticated user was not allowed to gain access without the proper access rights. Though their scheme was secure against node capture and denial-of-service (DoS) attacks, the password change process was not supported. For new user additions, the base station needed to rebroadcast user parameters such as user ID, group ID and system timestamp, thus incurring more communication overhead in the network.

Wang et al. (2006) proposed an ECC-based user access control scheme. In this scheme, the user must register with the key distribution center (KDC) for access permission prior to authentication. The KDC maintains a user access list pool with the respective user's access privilege. This access privilege consists of user ID, group ID and a user access privilege mask; multiple users within the same group should have the same access privilege. Based on elliptic curve cryptography, the KDC generates the public key, the private key of the user and the access list certificate, based on the user's request. The user requests the sensor node by sending the certificate; the sensor node then selects one random number as a session key. In this scheme, the user authenticates a sensor node and a sensor node also authenticates the user; mutual authentication is thus provided between the user and the sensor node.

Le et al. (2009) proposed an energy-efficient access control scheme based on ECC that improved on Wang et al. (2006). Their scheme was a public key cryptography based access control scheme where the user must accept access permissions from a key distribution center (KDC). The KDC maintains an access control list (ACL) pool and associated user identifications. The user's access privileges are defined in the ACL based on the user's access privilege mask. The public keys between the KDC and the sensor nodes are mutually exchanged during the pre-deployment phase. After registration, the user gains a public and private key. One signed certificate of the access control list is also issued by the KDC and sent to the user. The user must then be authenticated by the sensor node for future communications.

## 3. The proposed scheme

In a wireless sensor network that controls user access, a genuine user gains permission to access the network. However, in real life WBAN scenarios, all users should not have the same network access privileges. A particular user should only be able to access required information. To provide controlled user

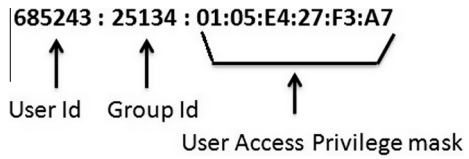



**Figure 2** An example of a user access list.

access for WBANs, we propose a new access control scheme utilizing an access list composed of a user identity, a user access privilege mask and an access group ID $G_{id}$ for each user. $G_{id}$ represents a unique number to identify a particular access group. Each access group can access data according to the privileges given to that particular group. A user access privilege mask is a binary number where each bit represents specific information or services that can be accessed by an authenticated user. A sensor node stores and processes information, sending partially processed information to the next level. An authenticated user with a lower level of privilege is not allowed to access higher privilege information (Wang et al., 2008). An example of a user access list is shown in Fig. 2.

### 3.1. Notations

We use the notations in this paper to describe our proposed scheme given in Table 1. The public key of the base station is $K_{BS} = xG$, where $xG = G + G + \ldots + G(x\ times)$ is called the elliptic curve scalar multiplication in an elliptic curve $E_p(a, b)$, which is the set of all points of $y^2 = x^3 + ax + b(mod\ p)$ such that a, $b \in Z_p = \{0, 1, 2, \ldots, p-1\}$ are constants with $4a^3 + 27b^2 \neq 0(mod\ p)$. If $nG = O$, where $O$ the point at infinity or zero point. Then $O$ is called the order of the base point $G$ in $E_p(a, b)$ (Koblitz, 1987). Here $x$ is the private key of the base station. An example of a one-way hash function is SHA-1 (Secure Hash Standard, 1995), which has the above desired properties (i) to (vi). However, National Institute of Standards and Technology (NIST) does not recommend SHA-1 for top secret documents. Further, in 2011, Manuel showed collision attacks on SHA-1 (Manuel, 2011). As in Das (2012, 2013) one can also use the recently proposed one-way hash function, Quark (Aumasson et al., 2010). Quark is a family of cryptographic hash functions which is designed for extremely resource-constrained environments like sensor networks and radio-frequency identification (RFID) tags. Like most one-way hash functions, Quark can be used as a pseudo-random function (PRF), a message authentication code (MAC), a pseudo-random number generator (PRNG), a key derivation function, etc. Quark is shown to be a much efficient hash function than SHA-1. However, in this paper, as in Das et al. (2013) we use SHA-2 as the secure one-way hash function in order to achieve top security. We may use only 160-bits from the hash digest output of SHA-2.

**Table 1** Notations used in the proposed scheme.

| Symbol | Description |
|---|---|
| $SN_i$ | Identifier of sensor node $i$ |
| $U_j$ | $j$th user |
| $BS$ | Base station |
| $PW_j$ | Password of user $U_j$ |
| $G_{id_j}$ | Group id of user $U_j$ |
| $APM_j$ | Access privilege mask of user $U_j$ |
| $x$ | Private key of base station |
| $K_{BS}$ | Public key of base station |
| $MK_{S_i}$ | Master key of sensor node $SN_i$ |
| $RM_{u_j}$ | Random number for user $U_j$ |
| $K_i$ | Secret key of node $SN_i$ shared with BS |
| $H(\cdot)$ | Secure one-way collision-resistant hash function |
| $T_i$ | Bootstrapping time for node $SN_i$ |
| $A\|\|B$ | Data $A$ concatenates with data $B$ |
| $E_K(M)$ | Symmetric encryption using the key $K$ |
| $D_K(M)$ | Symmetric decryption using the key $K$ |
| $X \rightarrow Y{:}M$ | Entity $X$ sends message M to entity $Y$ |

### 3.2. Different phases

This section discusses our proposed user access control scheme. Our scheme consists of the following phases: pre-deployment, post-deployment, registration, login, authentication, password change and dynamic node addition. These phases are described in the following subsections.

#### 3.2.1. 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 (the medical server). This phase is implemented offline by the base station prior to the deployment of sensor nodes on a patient's body (target field). The pre-deployment phase consists of the following steps:

Step P1: The base station selects a set of network parameters from the following: a finite field GF($p$) where $p$ is a large odd prime of at least 160 bits; an elliptic curve $E_p(a, b)$ that is the set of all points of $y^2 = x^3 + ax + b(mod\ p)$ such that $a, b \in Z_p = \{0, 1, 2, \ldots, p-1\}$ are constants with $4a^3 + 27b^2 \neq 0\ (mod\ p)$; and a base point $G$ in $E_p(a, b)$ whose order is $n$, where $n$ is at least 160 bits such that $n > 4\sqrt{p}$. The base station first selects a random number as its own private key $x \in Z_n^*$ where $Z_n^* = \{1, 2, \ldots, n-1\}$. The base station then computes its public key $K_{BS} = xG$. Depending on the probable user query, the base station prepares the group-based user access privilege mask (APM) and prepares an access list consisting of the access privilege mask and the respective access group identity $G_{id}$. For each deployed sensor node $SN_i$, the base station assigns a unique identifier $SN_i$. The base station also assigns a unique randomly generated master key $MK_{S_i}$ for each deployed sensor node $SN_i$, which is only shared with the base station. The base station computes $x_iG = (x_1, y_1)$ for each sensor node $SN_i$ where $x_i$ is the private key for sensor node $SN_i$, which is known to the BS. The base station then computes the secret key $K_i = x_1\ (mod\ p)$ for each sensor node $SN_i$. For security, $p$ is considered as a 160-bit number for ECC. Note that $K_i$ is also a 160-bit number. However, to use $K_i$ as the secret key for symmetric key encryption (for example, Advanced Encryption Standard (AES) (Advanced Encryption Standard, 2001)), we can only use 128 bits from the 160 bits of $K_i$.

Step P2: Once the set of network parameters are selected, the base station (BS) loads the following information into the memory of each sensor node $SN_i$ prior to its deployment in offline: (i) a unique node identifier $SN_i$; (ii) the elliptic curve $E_p(a, b)$; (iii) the base point $G$; (iv) the secret key $K_i$ with $x_i$; (v)

the base station's public key $K_{BS}$; (vi) a secure one-way hash function H(·); and (vii) its own master key $MK_{S_i}$.

### 3.2.2. 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 (doctors, nurses, medical insurance team, patient parties, etc.) can access the real-time data for monitoring a patient's condition from the sensors inside the WBAN, we require secure communication between the sensor nodes and the authorized users.

Once deployed, each sensor node sends a message with its node identity $SN_i$, bootstrapping time $T_i$, and encrypted information containing $K_i$, $SN_i$ and $T_i$ to the base station:

$$SN_i \rightarrow BS : \langle SN_i, T_i, E_{MK_{S_i}}(K_i, SN_i, T_i)\rangle$$

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

### 3.2.3. Registration phase

In the registration phase, a user $U_j$ must register with the base station to access the real-time data from a specific sensor node in a WBAN. This phase consists of the following steps:

Step R1: The user selects his/her identity $U_j$, a password $PW_j$, his/her access group ID $G_{idj}$ (depending on his/her access privilege), and a random number $RM_{uj}$. $U_j$ generates another secret random secret value $N_j$ that is kept secret to $U_j$ only. $U_j$ then computes the masked password $RPW_j = H(N_j\|PW_j)$ and sends the message $\langle U_j, RPW_j, G_{id_j}, RM_{uj}\rangle$ to the BS through a secure channel.

Step R2: After receiving the information, the BS calculates the secret shared hash value $R_{Uj} = H(RPW_j\|G_{idj}\|RM_{uj})$ for user $U_j$.

Step R3: The BS finally generates a tamper-proof smart card for user $U_j$ with the following parameters and sends the smart card to $U_j$ through a secure channel:

$BS \rightarrow U_j$: $\langle SmartCard(U_j, RM_{uj}, H(\cdot), RPW_j, G_{idj}, R_{U_j})\rangle$

The BS stores $R_{U_j}$, $G_{idj}$ and $APM_j$ for user $U_j$. This registration phase is summarized in Table 2.

### 3.2.4. Login phase

This phase allows users to login to the system to access real-time data from a specified sensor node in a WBAN. The user $U_j$ must perform the following steps:

Step L1: At login, the user $U_j$ inserts his/her smart card into the card reader of a specific terminal and inputs his/her user ID $U_j$, secret value $N_j$ and password $PW_j$, as well as his/her access group ID $G_{idj}$. The smart card then computes the masked password $RPW'_j = H(N_j\|PW_j)$ and the hash value $R'_{Uj} = H\left(RPW'_j\|G_{idj}\|RM_{uj}\right)$ for user $U_j$, using the stored values of $G_{idj}$, $RM_{uj}$ in the smart card. The smart card checks whether $R'_{Uj} = R_{Uj}$. If this verification does not hold, $U_j$ has entered his/her password incorrectly and the phase terminates immediately. Otherwise, the smart card computes the hash value H $(R_{U_j}\|T_1)$ by using the timestamp $T_1$ of the system and then sends the following message to the BS:

$$U_j \rightarrow BS :< U_j, H(R_{Uj}\|T_1), T_1 >$$

Step L2: After receiving the message in Step $L1$, the BS checks whether the condition $|T_1 - T_1^*| < \Delta T_1$ is valid, where $T_1$ is the timestamp of the user's system, $T_1^*$ is the current timestamp of the BS and $\Delta\ T_1$ is the expected time interval for the transmission delay. If it is valid, the BS computes the hash value $H(R_{U_j}\|T_1)$ using the received timestamp $T_1$ and the previously computed value of $R_{U_j}$ by the BS. The BS then compares this computed hash value with the received hash value $H(R_{U_j}\|T_1)$ in the message. If they match, the BS computes the secret parameter $S_j = x + x_i R_{U_j}\ (mod\ p)$ and the hash value $K_{Uj} = H(SN_i\|U_j\|K_{BS}\|K_i)$ for all sensor nodes $SN_i$, $i = 1$,

**Table 2** The registration phase of our Proposed Scheme.

| User $(U_j)$ | BS |
|---|---|
| Selects $U_j$ and $PW_j, N_j$, $G_{id_j}$ and $RM_{u_j}$. Computes $RPW_j = H(N_j \| PW_j)$. | |
| $\xrightarrow{\langle U_j, RPW_j, G_{id_j}, RM_{u_j}\rangle}$ | |
| | Computes $R_{U_j} = H(RPW_j \| G_{id_j} \| RM_{u_j})$. |
| | $\xleftarrow{SmartCard(U_j, RM_{u_j}, H(\cdot), RPW_j, G_{id_j}, R_{U_j})}$ |

2,...,$n$, and user $U_j$. The BS then computes $Z_j = K_{BS} + K_i R_{U_j} \pmod p$. The BS further computes the shared secret symmetric key $U\ K_j = H(R_{U_j} \| U_j \| T_1 \| T_2)$ with the user $U_j$ and sends the following message to the user $U_j$:

$$BS \rightarrow U_j : \langle E_{UKj}(SN_i, S_j, Z_j, K_{Uj}), T_1, T_2 \rangle$$

Step L3: After receiving the message in Step L2 from the BS, user $U_j$ verifies whether $|T_2 - T_2^*| < \Delta T_2$ is valid, where $T_2$ is the timestamp of the BS, $T_2^*$ the timestamp of the user's system and $\Delta T_2$ the expected time interval for the transmission delay. $U_j$ also checks the received value of $T_1$ with its previous $T_1$. If they match, it computes the same symmetric key $UK_j$ shared with the BS with the received value of $T_1$, $T_2$ as $UK_j = H(R_{U_j} \| U_j \| T_1 \| T_2)$ and decrypts $E_{UK_j}(SN_i,\ S_j,\ Z_j,\ K_{Uj})$ to retrieve $S_j$, $Z_j$, and $K_{U_j}$. $U_i$ then stores the retrieved values of $S_j$, $Z_j$, and $K_{U_j}$ for authorization purposes with the sensor node $SN_j$.

Step L4: The BS computes two encrypted messages $E_{MK_{S_i}}(SN_i,\ U_j,\ (APM_j \oplus G_{id_j}),\ R_{Uj},\ T_1,\ T_2)$ by using the master key $MK_{S_i}$ of the sensor node $SN_i$ and $E_{K_i}(SN_i,\ U_j,\ G_{id_j},\ T_1)$ using the key $K_i$. The BS sends the following message to the sensor node $SN_i$:

$$BS \rightarrow SN_i : \langle SN_i,\ U_j,\ E_{MKS_i}(SN_i,\ U_j, (APM_j \oplus G_{idj}),\ R_{Uj},\ T_1,\ T_2),\ E_{Ki}(SN_i,\ U_j, G_{idj},\ T_1) \rangle$$

In this case, $APM_j$ is the access privilege mask for the access group ID $G_{idj}$ for user $U_j$.

Step L5: When the sensor node $SN_i$ receives the message in Step L4, it decrypts $E_{MK_{S_i}}(SN_i,\ U_j,(APM_j \oplus G_{idj}),\ R_{U_j},\ T_1,\ T_2)$ by using its own master key $MK_{S_i}$ to retrieve the information $SN_i$, $U_j$,$(APM_j \oplus G_{idj})$, $R_{U_j}$, $T_1$, $T_2$. $SN_i$ then checks the received $SN_i$, $U_i$, and $T_2$ values by checking the condition $|T_2 - T^*| < \Delta T_2$, where $T_2$ is the timestamp of the base station, $T^*$ the timestamp of the sensor node $SN_i$ and $\Delta T_2$ is the expected time interval for the transmission delay. If all of these conditions are satisfied, $SN_i$ further decrypts $E_{Ki}(SN_i,\ U_j,\ G_{idj},\ T_1)$ by using the stored key $K_i$ to retrieve the information $SN_i$, $U_j$, $G_{idj}$, $T_1$. By using $G_{idj}$, $SN_i$ computes $APM_j = (APM_j \oplus G_{idj}) \oplus G_{idj}$. $SN_i$ finally saves $R_{U_j}$, $T_1$, $T_2$, $G_{idj}$, and $APM_j$ for authentication purposes. The login phase of our scheme is summarized in Table 3.

### 3.2.5. Authentication phase

The authentication phase is required to authenticate the user when he/she wants to access real-time data inside a WBAN. During the login phase, when the user $U_j$ receives the message in Step L2, $U_j$ saves the values $S_j$, $Z_j$ and $K_{Uj}$ for authorization purposes with the sensor node $SN_i$ in Step L3.

Step A1: For authentication, the user $U_j$ computes the encrypted value $EK_{U_j}(SN_i,\ U_j,\ R_{U_j},\ G_{id_j},\ T_1,\ S_j,\ Z_j)$ and the hash value $H(T_1 \| S_j \| Z_j)$, sending the following authentication request message to the sensor node $SN_i$:

$$U_j \rightarrow SN_i : < SN_i, U_j, E_{KUj}(SN_i, U_j, R_{Uj}, G_{idj}, T_1, S_j, Z_j), H(T_1 \| S_j \| Z_j) >$$

Step A2: After receiving the authentication request message from user $U_j$in Step A1, the sensor node $SN_i$ performs the following to verify whether user $U_j$ is legitimate: $SN_i$ first computes the key $K'_{Uj} = H(SN_i \| U_j \| K_{BS} \| K_i)$, using the stored parameters and the received user ID of $U_j$. Using the computed key $K'_{Uj}, SN_i$ then decrypts $E_{KUj}(SN_i,\ U_j,\ R_{U_j},\ G_{idj},\ T_1,\ S_j,\ Z_j)$ to retrieve the information $SN_i$, $U_j$, $R_{U_j}$, $G_{idj}$, $T_1$, $S_j$, $Z_jSN_i$ further checks whether the retrieved value of $T_1$ matches with the previously received value of $T_1$. If they match, $SN_i$ computes the hash value $H(T_1 \| S_j \| Z_j)$ and verifies whether this value matches with the received hash value. If there is a match, $SN_i$ then proceeds in executing Step A3. Otherwise, the authentication phase immediately terminates.

Step A3: $SN_i$ checks the following signature verification equation $Z_j = S_j G \pmod p$. Note that

$$Z_j = (K_{BS} + x_i G R_{Uj}) = (xG + x_i G R_{Uj}) = (x + x_i R_{Uj}) G = S_j G.$$

If the signature verification fails, $SN_i$ considers user $U_j$ as illegal and the phase terminates immediately. Otherwise, the sensor node $SN_i$ checks the received $G_{idj}$ with the value received from the BS during the login phase. If it is satisfied, $SN_i$ computes a secret session key $SK_{ij}$ to be shared with the user $U_j$ as $SK_{ij} = H(SN_i \| U_j \| APM_j \| G_{id_j} \| S_j \| R_{U_j} \| T_1 \| T_2)$. Finally, $SN_i$ sends an acknowledgment to user $Uj$ and the BS, and responds to the query of the user $Uj$, depending upon the access privilege mask $APM_j$ stored for user $U_j$ using the secret session key $SK_{ij}$.

Step A4: After receiving the acknowledgment from $SN_i$, user $U_j$ computes the same secret session key $SK_{ij}$ shared with the sensor node $SN_i$ using its previous system timestamp $T_1$, storing $T_2$, $S_j$, $R_{U_j}$ as $SK_{ij} = H(SN_i \| U_j \| APM_j \| G_{id_j} \| S_j \| R_{U_j} \| T_1 \| T_2)$. Therefore, both user $U_j$ and the sensor node $SN_i$ will securely communicate in future using the derived secret session key $SK_{ij}$.

At the end of this phase, $SN_i$ deletes $R_{U_j}$, $T_1$, $T_2$, $G_{idj}$, and $AP\ M_j$ from its memory for security reasons. User $U_j$ also deletes $S_j$ and $Z_j$. The authentication phase of our scheme is summarized in Table 4.

### 3.2.6. Password change phase

In this phase, a user $U_j$ may change his/her password freely and completely locally for security reasons without contacting the BS. This phase consists of the following steps:

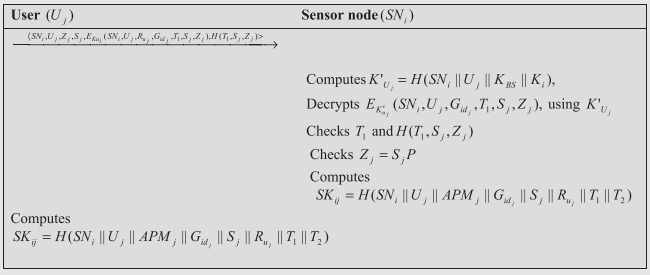
Step PC1: $U_j$ inputs his/her smart card into the card reader of a specific terminal and provides his/her old password $PW_j^{old}$ and secret number $N_j^{old}$, as well as new changed password $PW_j^{new}$ and new secret number $N_j^{new}$.

Step PC2: The smart card computes the masked old password of the user $U_j$ as $RPW_j^{old} = H\left(N_j^{old} \| PW_j^{old}\right)$ and compares this value with the stored value of $RPW_j$ in the smart card. If they do not match, this means that the user $U_j$ has entered his/her old password incorrectly and the password change phase terminates immediately. Otherwise, the smart card computes the hash value $R_{Uj}^{old} = H(RPW_j^{old} \| G_{idj} \| RM_{Uj})$ with the old masked password $RPW_j^{old}$, group identity $G_{idj}$ and random number $RM_{Uj}$. The smart card then further compares this computed hash value $R_{Uj}^{old}$ with the stored value of $R_{U_j}$. If they match, the smart card executes Step PC3.

Step PC3: The smart card computes the new masked password $RPW_j^{new} = \mathrm{H}\left(N_j^{new} \| PW_j^{new}\right)$ and $R_{Uj}^{new} = \mathrm{H}(RPW_j^{new} \| G_{idj} \| RM_{Uj})$.

**Table 3** The login phase of our proposed scheme


**Table 4** Authentication phase of our proposed scheme.




Step PC4: The smart card replaces the old masked password $RPW_j$ with the new masked password $RPW_j^{new}$ and the old hash value $R_{U_j}$ with the new hash value $R_{Uj}^{new}$ into the memory of the smart card.

### 3.2.7. 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 compromised 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), the BS will preload a set of node parameters offline. This set contains (i) a unique node identifier $SN_u$ of the node $u$; (ii) the elliptic curve $E_p(a, b)$; (iii) the base point $G$ in $E_p(a, b)$; (iv) the secret key $K_u$ with $x_u$ for node $SN_u$, where $x_u$ is the private-key of $SN_u$ and $x_uG = (x_1, y_1)$ with $K_u = x_1(mod\ p)$; (v) the base station's public key $K_{BS}$; (vi) a hash function H(·); and (vii) its own master key $MK_{S_u}$.

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

$$SN_u \rightarrow BS :< SN_u,\ T_u,\ E_{MK_{Su}}(K_u,\ SN_u,\ T_u) >$$

After deployment, $SN_u$ establishes pairwise keys between them in the WBAN by using (Das AK, 2009). Then, $SN_u$ authenticates and establishes pairwise symmetric secret keys with user $U_j$ as described in Sections 3.2.4 and 3.2.5. 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.

## 4. Analysis of the proposed scheme

In this section, we perform functionality and security analyses of our proposed access control scheme.

### 4.1. Computational overhead

Let $t_{ecm}$, $t_h$, $t_{enc}$, and $t_{dec}$ denote the time required to perform an elliptic curve scalar multiplication, a one-way hash function H(·), a symmetric key encryption, and a symmetric key decryption, respectively. During the registration phase, the user $U_j$ and the BS require the computational overhead $t_h$ and $t_h$, respectively. In our proposed scheme, during the login and authentication phases, the user $U_j$, the BS and the sensor node $SN_i$ require the computational overhead $6t_h + t_{dec} + t_{enc}$, $3t_h + 2t_{ecm} + 2t_{enc} + t_{eca}$ and $3t_h + 3t_{dec} + t_{ecm}$, respectively. The total computational cost becomes $14t_h + 8t_{enc}/t_{dec} + 3t_{ecm} + t_{eca}$.

### 4.2. Communication overhead

We consider the communication overhead of our scheme for both the login and authentication phase. Based on the login and authentication phases of our scheme, it is clear that the sensor node $SN_i$, the BS and the user $U_j$ must exchange four messages. We have calculated the bitwise and packetwise communication overhead for our proposed scheme during the login and authentication phases. For computing the number of packets required for transmission, we considered a CC2420 transmitter (CC2420:2.4 GHz IEEE 802.15.4, 2011). A CC2420 transmitter supports a packet size of 128 bytes, i.e., 1024 bits. To calculate the communication overhead, we used the bitwise size of different parameters as shown in Table 5.

In Table 6, we calculated the number of bits and packets required for each message in our scheme during the login and authentication phases. It should be noted that we required a communication overhead of 1008 bits and the transmission of only 4 packets during the login and authentication phases.

### 4.3. Storage overhead

During the pre-deployment phase described in Section 3.2.1, a sensor node $SN_i$ primarily requires storage space to meet the following node parameters: a unique node identifier $SN_i$, which needs 16 bits; the elliptic curve $E_p(a, b)$, which needs (160 + 160 + 160) = 480 bits for storing $p$, $a$ and $b$ values of 160 bits each (for security reasons, we have considered 160 bits prime $p$ in ECC); the base point G, which needs (160 + 160) = 320 bits; the secret key $K_i$ with private key $x_i$ for $SN_i$, which needs (160 + 160) = 320 bits; the base station's public key $K_{BS}$, which needs (160 + 160) = 320 bits; and its own master key $MK_{S_i}$, which needs 128 bits. The total storage space of the sensor node $SN_i$ prior to its deployment becomes 1584 bits.

**Table 5** Size (in bits) of different parameters used for our scheme.

| Type | Bitwise size |
|---|---|
| User identifier, $U_j$ | 16 |
| Bootstrapping time, $T_i$ | 32 |
| Node identifier, $SN_i$ | 16 |
| Group identifier, $G_{id_j}$ | 8 |
| Access privilege mask, $APM_j$ | 64 |
| Random number, $RM_{u_j}$ | 32 |
| Hash value | 160 |
| Symmetric encryption, $E_K(M)$ | 128 |

### 4.4. Energy consumption

Based on Zhang et al. (2012), we also use the Chipcon CC2420 (CC2420:2.4 GHz IEEE 802.15.4, 2011) configuration that is widely used in low-rate wireless personal area networks. Table 7 shows that the CC2420 supports a total of eight transmission power levels and a typical supply current (Zhang et al., 2012). As noted in Zhang et al. (2012), $I_r$ = 19.7 mA is required to receive the signal, and the transmission rate is 250 kb/s.

Based on Zhang et al. (2012), we evaluated the energy consumption for communication through the following three-case model: Case I: Success: both data packets and acknowledgments are successfully transmitted.

Case II: PF: Unsuccessful data packet transmission.

Case III: AF: Successful data packet transmission followed by an unsuccessful acknowledgment transmission.

According to Zhang et al. (2012), the total energy consumption for communication can be calculated as

$$\mathrm{E}(\cdot) = \mathrm{E}(\cdot|\text{Success}) + \mathrm{E}(\cdot|\text{PF}) \times \mathrm{N}_{\mathrm{PF}}(\cdot) + \mathrm{E}(\cdot|\text{AF}) \times \mathrm{N}_{\mathrm{AF}}(\cdot),$$

where E(·| Success), E(·| PF ), and E(·| AF) represent the energy required for Case I: successful transmission, Case II: packet failure, and Case III: acknowledgment failure. $\mathrm{N}_{PF}(\cdot)$ denotes the expected number of packet transmission failures, and $\mathrm{N}_{\mathrm{AF}}(\cdot)$ is the expected number of acknowledgment transmission failures. For a detailed analysis, refer to Zhang et al. (2012).

### 4.5. Network scalability

Assume that there will be $m$ cluster heads and $m'$ controller nodes in a hierarchical WBAN (HWBAN) as shown in Fig. 3, representing a hospital ward with multiple patients. In this figure, a set of sensor nodes are deployed on a patient's body that constitute a WBAN. The WBAN is then associated with a cluster head, and a set of cluster heads are attached to a controller node. For example, if a patient's body is deployed with 10 regular sensor nodes and there are 1000 patients in various wards to be monitored in the hospital,

the total number of regular sensor nodes is $10 \times 1000 = 10{,}000$. If 5 cluster heads are attached to a controller node in a ward, we require 1000/5 = 200 controller nodes in the hospital. As a result, the total nodes to be deployed in the HWBAN is 11,200, and these nodes constitute a large-scale network.

In a case where a patient will be monitored at home, the total number of regular sensor nodes is 10 in the WBAN and only one cluster head is required in that WBAN. In both scenarios, the access control mechanism remains the same.

### 4.6. 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.

#### 4.6.1. 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 user's password because the BS and sensor nodes do not maintain any password/verifier table to validate a user's login request. During the registration phase of our scheme, a user securely $U_j$ submits his/her identity $U_j$ and masked password $H(N_j\|PW_j)$ to the BS. According to our threat model, the BS is considered to be a trustworthy entity in the network and cannot be compromised by any attacker. Because the secret value $N_j$ is only known to user $U_j$, it is computationally infeasible for the BS to retrieve $PW_j$ from $H(N_j\|PW_j)$ due to one-way property of the hash function H(·). Therefore, our scheme has the ability to prevent such an attack.

#### 4.6.2. 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 user's smart card. At the time of login, a user $U_j$ must have a valid smart card with the valid input tuple $\langle U_j, PW_j, N_j\rangle$. 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 due to random secret numbers $N_i$ and $N_j$ used in computation of their masked passwords, they will have different masked passwords. 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.

**Table 7** Transmission power levels of CC2420.

| Index $i$ | Transmission power $P_t(i)[dBm]$ | Transmission current $I_t(i)[mA]$ |
|---|---|---|
| 1 | −25 | 8.5 |
| 2 | −15 | 9.9 |
| 3 | −10 | 11.2 |
| 4 | −7 | 12.5 |
| 5 | −5 | 13.9 |
| 6 | −3 | 15.2 |
| 7 | −1 | 16.5 |
| 8 | 0 | 17.4 |

#### 4.6.3. 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% 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.

**Table 6** Message size and number of packets to be transmitted per message for our scheme during the login and authentication phases.

| Message | Exchange between | Size | No of packets |
|---|---|---|---|
| $\langle U_j, H(R_{u_j} \| T_1), T_1\rangle$ | $U_j$ and $BS$ | 208 | 1 |
| $< E_{UK_j}(SN_i, S_j, Z_j, K_{u_j}), T_2, T_1 >$ | $BS$ and $U_j$ | 192 | 1 |
| $\langle SN_i, U_j, E_{MKs_j}(SN_i, U_i, (APM_j \oplus G_{id_j}), R_{u_j}, T_1, T_2), E_{K_i}(SN_i, U_j, G_{id_j}, T_1)>$ | $BS$ and $SN_i$ | 288 | 1 |
| $\langle SN_i, U_j, Z_j, S_j, E_{Ku_i}(SN_i, U_j, R_{u_j}, G_{id_j}, T_1, S_j, Z_j), H(T_1, S_j, Z_j)>$ | $U_j$ and $SN_i$ | 320 | 1 |

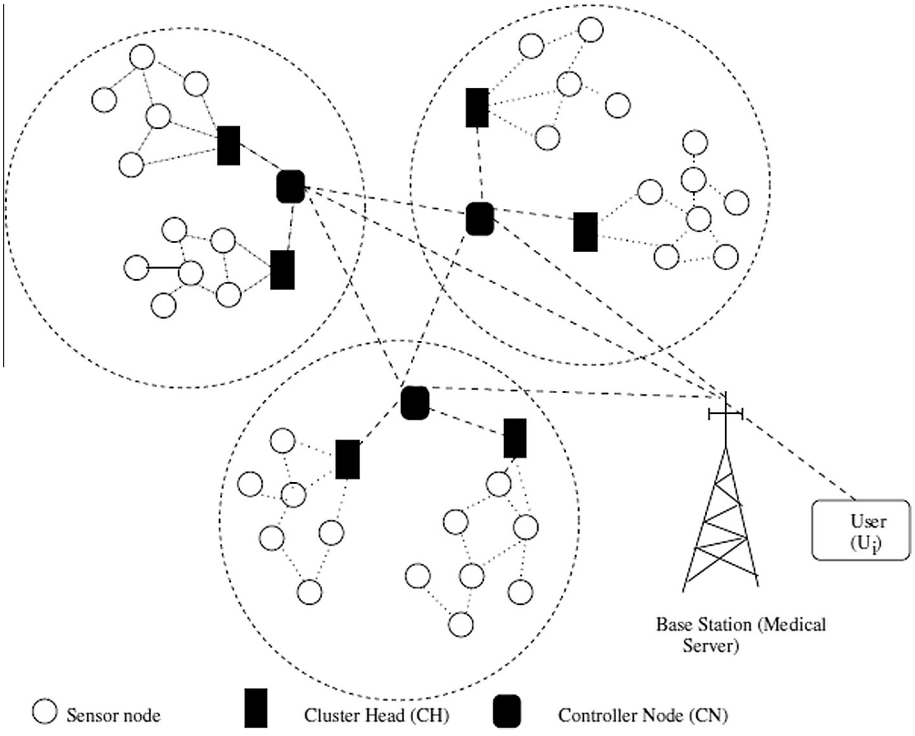



**Figure 3** An example of a hierarchical body area sensor network.

**Remark 1.** Note that in our scheme, the session key between the user and the sensor node in the BAN is secured after the successful authentication process. This key is used between the sensor and the user to secure the communication channel for the real-time data transmission. However, when a sensor node is physically captured by an attacker from a patient's body (WBAN), the attacker is able to discern the master key along with other information from its memory, including the established session key. As in our threat model discussed in Section 1.2, the compromised (captured) nodes can be detected and as a result, the base station (medical server), users and sensor nodes know the IDs of the compromised nodes. Consequently, the base station (medical server) alerts the users with the compromised sensor nodes in the network. Thus, another new sensor must be deployed in place of the captured sensor. In this case, with the help of the dynamic node addition phase described in Section 3.2.7, the newly deployed sensor will be able to establish a new session key and be shared with the user after a successful authentication process.

#### 4.6.4. 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 $U_j$, secret value $N_j$, password $PW_j$ and access group ID $G_{id_j}$. The smart card then computes the masked password $RPW'_j = H(N_j \| PW_j)$ and the hash value $R'_{U_j} = H(RPW'_j \| G_{id_j} \| RM_{u_j})$ for user $U_j$ by using the stored values of $G_{id_j}, RM_{U_j}$ in the smart card. The smart card checks whether $R'_{U_j} = R_{U_j}$. If this verification passes, user $U_j$ sends the login request message $\langle U_j , H(R_{U_j} \| T_1), T_1 \rangle$ to the BS. To convince the BS that this is a legal remote login request, the illegal user must know the value of $Nj$ as well as $PW_j$, $N_j$, $G_{id_j}$, and $RM_{uj}$. As a result, the attacker does not have the ability to create a fake login request message on behalf of the original user $U_j$. Thus, our scheme resists this type of attack.

#### 4.6.5. 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 $\langle U_j, H(R_{U_j} \| T_1), T_1 \rangle$ to the BS. Because the attacker cannot change the hash value $H(R_{U_j} \| T_1)$, 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.

#### 4.6.6. Privileged-insider attack

Note that during the registration phase of our proposed scheme, the user $U_j$ does not send his/her password $PW_j$ in plaintext. The user $U_j$ sends the masked password $RPW_j = H(N_j \| PW_j)$ to the BS. Without knowing the secret value $N_i$ (which is only known to the user $U_j$), it is computationally infeasible to retrieve $PW_j$ from $RPW_j$ due to the one-way property of the hash function H(·). A privileged insider at the BS does not have the ability to know the password $PW_j$ of user

$U_j$, and he/she is then unable to impersonate $U_j$ by accessing other servers where $U_j$ could also be a registered user and use the same password $PW_j$ for his/her convenience. Thus, our scheme protects against such an attack.

#### 4.6.7. Smart card breach attack

As in Fan et al. (2010), the smart card is assumed to be safe and unable to be cracked; however, there is a risk of smart card crack. If an attacker/intruder obtains a smart card and cracks it, we must assume that he/she can obtain its stored information, such as $U_j$, $RM_{uj}$, $H(\cdot)$, $RPW_j$, $G_{id_j}$, and $R_{U_j}$. However, the attacker has no feasible way to know user $U_j$'s password $PW_j$ from $RPW_j$ due to the one-way property of the hash function H(·). Moreover, based on the hash value $R_{U_j} = H(RPW_j \| G_{id_j} \| RM_{uj})$, it is also difficult to know $PW_j$ for $U_j$ due to one-way property of the hash function H(·). Therefore, the attacker must guess user $U_j$'s correct password $PW_j$ and secret number $N_j$ to pass clear the password verification during the login phase. In addition, the computation of $N_j$ at the login phase becomes infeasible due to the one-way property of the hash function H(·). As a result, our scheme prevents a smart card breach attack.

#### 4.6.8. 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. At the time of authentication, the BS sends an authentication request message to a specific sensor node $SN_i$ from which user $U_j$ wants to access real-time data inside the WBAN. After receiving the request message from user $U_j$, the sensor node $SN_i$ 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 $U_j$ at the end of user authentication.

#### 4.6.9. Formal security proof of the proposed scheme

This section shows through a formal security analysis that our scheme is secure against an attacker deriving the user's password and the base station's private key. For this purpose, we define the following formal definitions:

**Definition 1.** (One-way hash function). There exists a secure one-way hash function $H{:}X \to Y$, where $X = \{0,1\}^*$ and $Y = Z_p^* = \{a | 0 < a < p$ and $gcd(a, p) = 1\}$ satisfying the following requirements (Stallings, 2003):

(i) For a given $y \in Y$, it is hard to find an $x$ in $X$ such that $H(x) = y$.
(ii) For a given $x \in X$, it is hard to find another $x'$ in $X$, with $x' \neq x$, such that $H(x') = H(x)$.
(iii) It is hard to find a pair $x, x') \in X \times X$, with $x' \neq x$, such that $H(x') = H(x)$.

As defined in Sarkar (2010), Stinson (2006), a collision-resistant one-way hash function $H{:}X \to Y$, where $X = \{0,1\}^*$ and $Y = \{0, 1\}^n$, is considered as a deterministic algorithm that takes an input as an arbitrary length binary string $X = \{0,1\}'$ and produces an output $y \in \{0,1\}^n$ as a binary string of fixed-length, $n$. If $Adv_A^{HASH}(t)$ denotes an adversary (attacker) A's advantage in finding collision, we then have

$$Adv_A^{HASH}(t) = Pr[(x, x') \Leftarrow_R A : x \neq x', H(x) = H(x')],$$

where $Pr[E]$ denotes the probability of a random event E, and $(x, x') \Leftarrow_R A$ denotes the pair $(x, x')$ is selected randomly by $A$. In this case, the adversary A is allowed to be probabilistic and the probability in the advantage is computed over the random choices made by the adversary A with the execution time $t$. We call the hash function H(·) collision-resistant if $Adv^{H\ ASH}{}_A(t) \leqslant \in$, for any sufficiently small $\in > 0$.

**Definition 2.** (Indistinguishability of encryption and chosen plaintext attack (IND-CPA)). As in Wu and Chen (2012), we define the indistinguishability of encryption (IND) and chosen-plaintext attack (CPA) as follows. Let $SE/ME$ be the single/multiple eavesdropper, respectively, and $O_{k1}, O_{k2}, \ldots, O_{kN}$ be $N$ different independent encryption oracles associated with encryption keys $k_1, k_2, \ldots, k_N$, respectively. Define the advantage functions of $SE$ and $ME$, respectively as: $Adv_{\Omega,SE}^{ind\text{-}cpa}(l) = 2Pr[SE \leftarrow O_{k1}; (m_0, m_{1\leftarrow R}SE); \theta_{\leftarrow R}\{0,1\}; \gamma \leftarrow_R O_{k1}(m_\theta){:}SE(\gamma) = \theta] - 1$, and $Adv_{\Omega,ME}^{ind\text{-}cpa}(l) = 2Pr[ME \leftarrow O_{k1}, \ldots, O_{kN}; (m_0, m_{1\leftarrow R}ME); \theta_{\leftarrow R}\{0,1\}; \gamma_{1\leftarrow R}O_{k1}(m_\theta), \ldots, \gamma_{N\leftarrow R}O_{kN}(m_\theta){:}ME(\gamma_1, \ldots, \gamma_N) = \theta] - 1$ where $\Omega$ is the encryption scheme. Then, we say that the encryption scheme $\Omega$ is IND-CPA secure in the single (multiple) eavesdropper setting if $Adv_{\Omega,SE}^{ind\text{-}cpa}(l)$ (respectively, $Adv_{\Omega,ME}^{ind\text{-}cpa}(l)$) is negligible (in the security parameter l) for any probabilistic, polynomial time (PPT) adversary $SE$ $(ME)$.

**Definition 3.** (Elliptic curve discrete logarithm problem (ECDLP)). We define the elliptic curve discrete logarithm problem (ECDLP) formally given in Das et al. (2012a). Let $E_p(a,b)$ be an elliptic curve modulo a prime $p$. Let $P \in E_p(a,b)$ and $Q = kP \in E_p(a,b)$ be two points, where $k \in_R Z_p$ (We use the notation $a \in_R B$ to denote that $a$ is chosen randomly from the set $B$).

$$Instance : (P, Q, r) \text{ for some } k, r \in_R Z_p.$$
$$Output : Yes, \text{if } Q = rP, \text{i.e.}, k = r, \text{and output } No, \text{otherwise}.$$

$$\Delta real = \{k \in_R Z_p, A = P, B = Q(= kP), C = k : (A, B, C)\},$$
$$\Delta rand = \{k, r \in_R Z_p, A = P, B = Q(= kP), C = r : (A, B, C)\}.$$

The advantage of any probabilistic, polynomial-time, 0/1-valued (false/true-valued) distinguisher $D$ in solving ECDLP on $E_p(a, b)$ is defined as

$$Adv_{D,E_p(a,b)}^{ECDLP} = |P\ r[(A, B, C) \leftarrow \Delta real : D(A, B, C) = 1] - P\ r[(A, B, C) \leftarrow \Delta rand : D(A, B, C) = 1]|,$$

where the probability Pr[·] is taken over the random choices of $k$ and $r$. $D$ is said to be a $(t, \in)$-ECDLP distinguisher for $E_p(a, b)$ if $D$ runs at most in time $t$ such that $Adv_{\Delta,Ep(a,b)}^{ECDLP}(t) \geqslant \in$.

*ECDLP assumption*: There exists no $(t, \in)$-ECDLP distinguisher for $E_p(a, b)$. In other words, for every probabilistic,

polynomial-time 0/1-valued distinguisher $D$, we have $Adv^{ECDLP}_{\Delta,E_p(a,b)}(t) \leqslant \in$ for any sufficiently small $\in > 0$.

We define the following three random oracles for the attacker (adversary) $A$:

*Reveal1:* This unconditionally outputs $k$ from given points $P$ and $Q = kP$ in an elliptic curve $E_p(a,b)$.

*Reveal2:* This unconditionally outputs the plaintext message $M$ using symmetric-key cryptosystem $\Omega$ with the help of the relevant public parameters and cipher text message $E_{key}(M)$, without knowing the symmetric key, *key*.

*Reveal3:* This unconditionally outputs the input $x$ from the corresponding hash value $y = H(x)$.

in solving the one-way hash function and the indistinguishability of the encryption and chosen plaintext attack (IND-CPA). As a result, $Adv1^{HASH,IND\text{-}CPA}_{UACS,A}(t_1, q_{R_2}, q_{R_3}) \leqslant \varepsilon$, for any sufficiently small $\in > 0$, as it is dependent on $Adv^{ind\text{-}cpa}_{\Omega,ME}(l)$ and the difficulty of inverting the one-way hash function, i.e., $Adv^{HASH}_{A}(t)$. Therefore, our scheme is probably secure against an attacker deriving a user's password. □

**Algorithm 1.**

1: Eavesdrop the message $\langle U_j, H(R_{U_j}||T_1), T_1\rangle$ during the login phase, which is sent from the user $U_j$ to the BS.
2: Call $Reveal3$ oracle on the input $H(R_{U_j}||T_1)$ to retrieve the information $R_{U_j}$ and $T_1$. Let $(R'_{U_j}, T'_1) \leftarrow Reveal3(H(R_{U_j}||T_1))$.
3: Check if $T'_1$ matches with $T_1$ in the eavesdropped message. If so, call $Reveal3$ oracle on the input $R'_{U_j} = H(RPW_j||G_{id_j}||RM_{u_j})$, where $RPW_j = H(N_j||PW_j)$, in order to retrieve the information $RPW_j$, $G_{id_j}$ and $RM_{u_j}$. Let $(RPW'_j, G'_{id_j}, RM'_{u_j}) \leftarrow Reveal3(R'_{U_j})$.
4: Call $Reveal3$ oracle on the input $RPW'_j$ to derive $N_j$ and $PW_j$ of the user $U_j$. Let $(N'_j, PW'_j) \leftarrow Reveal3(RPW'_j)$.
5: Eavesdrop the message $\langle SN_i, U_j, E_{MK_{S_i}}(SN_i, U_i, (APM_j \oplus G_{id_j}), R_{U_j}, T_1, T_2), E_{K_i}(SN_i, U_j, G_{id_j}, T_1)\rangle$ during the login message, which is sent from the BS to a sensor node $SN_i$.
6: Call $Reveal2$ oracle on the input $E_{K_i}(SN_i, U_j, G_{id_j}, T_1)$.
Let $(SN''_i, U''_j, G''_{id_j}, T''_1) \leftarrow Reveal2(E_{K_i}(SN_i, U_j, G_{id_j}, T_1))$.
7: **if** $(G''_{id_j} = G_{id_j})$ and $(T''_1 = T_1)$ **then**
8: Accept the derived password $PW'_j$ as the correct password $PW_j$ of the user $U_j$.
9: **return** 1 (Success)
10: **else**
11: **return** 0 (Failure)
12: **end if**

**Theorem 1.** *Let the used symmetric encryption scheme $\Omega$ be IND-CPA. Our scheme is then secure against deriving a user's password by an attacker under the assumption that the one-way hash function $H(\cdot)$ closely behaves like a random oracle.*

**Proof.** We follow the similar proof as in Das et al. (2012a) and Odelu et al. (2013). We must construct an adversary $A$ that can correctly derive the user $U_j$'s password $PW_j$. For this purpose, the adversary $A$ runs the experiment given in Algorithm 1 for our proposed user access control scheme *UACS*.

We define the success probability for $EXP1^{HASH,IND\text{-}CPA}_{UACS,A}$ provided in Algorithm 1 as

$$Succ1^{HASH,IND\text{-}CPA}_{UACS,A} = |2\Pr[EXP1^{HASH,IND\text{-}CPA}_{UACS,A} = 1] - 1|.$$

The advantage function for this experiment is given by

$$Adv1^{HASH,IND\text{-}CPA}_{UACS,A}(t_1, q_{R_2}, q_{R_3}) = \max_A\{Succ1^{HASH,IND\text{-}CPA}_{UACS,A}\},$$

where the maximum is taken over all $A$ with the execution time $t_1$, and the number of queries $q_{R_2}$ made to the *Reveal2* oracle and the number of queries $q_{R_3}$ made to the *Reveal3* oracle. Our scheme is probably secure against an adversary $A$ for deriving a user's password by an attacker, if $Adv1^{HASH,IND\text{-}CPA}_{UACS,A}(t_1, q_{R_2}, q_{R_3}) \leqslant \varepsilon$, *for any sufficiently small* $\varepsilon > 0$.

Finally, consider the experiment $EXP1^{HASH,IND\text{-}CPA}_{UACS,A}$. According to this experiment, if the adversary $A$ can correctly derive the private key of the BS, he/she can win the game. However, it is computationally infeasible due to the difficulty

**Theorem 2.** *Let the used symmetric encryption scheme $\Omega$ be IND-CPA. Under the ECDLP assumption, our scheme is secure against an attacker deriving the base station's private key if the hash function $H(\,)$ closely behaves like a random oracle.*

**Proof.** We must construct an adversary $A$ that can correctly derive the base station BS's private key $x$. For this purpose, the adversary A runs the experiment $Exp2^{HASH,IND\text{-}CPA,ECDLP}_{UACS,A}$ given in Algorithm 2 for our proposed user access control scheme *UACS*.

We define the success probability for the experiment in Algorithm 2 as

$$Succ2^{HASH,IND\text{-}CPA,ECDLP}_{UACS,A} = |3\Pr[Exp2^{HASH,IND\text{-}CPA,ECDLP}_{UACS,A} = 1] - 1|.$$

The advantage function for this experiment is given by

$$Adv2^{HASH,IND\text{-}CPA,ECDLP}_{UACS,A}(t_2, q_{R_1}, q_{R_2}, q_{R_3}) = \max_A\{Succ2^{HASH,IND\text{-}CPA,ECDLP}_{UACS,A}\},$$

where the maximum is taken over all $A$ with the execution time $t_2$, and the number of queries $q_{R_1}, q_{R_2}, q_{R_3}$ made to the *Reveal1*, *Reveal2* and *Reveal3* oracles, respectively. Our scheme is called probably secure against an adversary $A$ deriving the base station's private key if

$$Adv2^{HASH,IND\text{-}CPA,ECDLP}_{UACS,A}(t_2, q_{R_1}, q_{R_2}, q_{R_3}) \leqslant \varepsilon,$$

for any sufficiently small $\varepsilon > 0$.

**Algorithm 2.**

1: Eavesdrop the message $\langle U_j, H(R_{U_j}||T_1), T_1\rangle$ during the login phase, which is sent from the user $U_j$ to the BS.
2: Call $Reveal3$ oracle on the input $H(R_{U_j}||T_1)$ to retrieve the information $R_{U_j}$ and $T_1$. Let $(R'_{U_j}, T'_1) \leftarrow Reveal3(H(R_{U_j}||T_1))$.
3: Check if $T'_1$ matches with $T_1$ in the eavesdropped message. If so, eavesdrop the message $\langle E_{UK_j}(SN_i, S_j, Z_j, K_{U_j}), T2, T1\rangle$ during the login phase, which is sent from the BS to the user $U_j$.
4: Call $Reveal2$ oracle on the input $E_{UK_j}(SN_i, S_j, Z_j, K_{U_j})$. Let $(SN'_i, S'_j, Z'_j, K'_{U_j}) \leftarrow Reveal2(E_{UK_j}(SN_i, S_j, Z_j, K_{U_j}))$.
5: Compute $UK'_j = H(R'_{U_j}||U_j||T_1||T_2)$, and encrypt the information using the key $UK'_j$ as $E_{UK'_j}(SN'_i, S'_j, Z'_j, K'_{U_j})$. If this encrypted value $E_{UK'_j}(SN'_i, S'_j, Z'_j, K'_{U_j})$ matches with received $E_{UK_j}(SN_i, S_j, Z_j, K_{U_j})$, accept $K'_{U_j}$ as the correct $K_{U_j}$.
6: Call $Reveal3$ oracle on the input $K'_{U_j}$ to retrieve $K_BS$. Let $(SN'_i, U'_j, K'_{BS}, K'_i) \leftarrow Reveal3(K'_{U_j})$, where $K_{U_j} = H(SN_i||U_j||K_{BS}||K_i)$.
7: Call $Reveal1$ oracle on the input $K'_{BS}$ to derive the private key $x$ of the BS. Let $x' \leftarrow Reveal1(K'_{BS})$. Compute $Z''_j = K'_{BS} + K'_i R'_{U_j} \pmod p$.
8: **if** $(Z''_j = Z'_j)$ **then**
9: Accept the derived $x'$ as the correct private key $x$ of the BS.
10: **return** 1 (Success)
11: **else**
12: **return** 0 (Failure)
13: **end if**

Consider the experiment $Exp2^{HASH,IND\text{-}CPA,ECDLP}_{UACS,A}$. According to the experiment, if the adversary $A$ can correctly derive the user password, he/she can win the game. However, it is computationally infeasible due to the difficulty of solving the one-way hash function, the indistinguishability of the encryption and chosen plaintext attack (IND-CPA) and the elliptic curve discrete logarithm problem (ECDLP). As a result, $Adv2^{HASH,IND\text{-}CPA,ECDLP}_{UACS,A}(t_2, q_{R_1}, q_{R_2}, q_{R_3}) \leqslant \varepsilon$, for any sufficiently small $\varepsilon > 0$, because it is dependent on $Adv^{ind\text{-}cpa}_{\Omega,\mathsf{ME}}(l), Adv^{ECDLP}_{\Delta,\mathsf{Ep}(a,b)}(t)$ and $Adv^{HASH}_{\mathsf{A}}(t)$. Therefore, our scheme is probably secure against an attacker deriving the private key of the BS.

## 5. 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.

### 5.1. AVISPA tool

AVISPA (Automated Validation of Internet Security Protocols and Applications) (Armando, 2005) is a widely-accepted and powerful tool for the formal security verification of a protocol, which ensures whether the protocol is secure or not. Model checking methods are used to search for states of the system whether some properties are violated or not. Model checking tools have been successfully employed to detect attacks on security protocols (Basin et al., 2005). We have used AVISPA back-ends for our formal security verification. AVISPA implements four different back-ends and abstraction-based methods which are integrated through the high level protocol specific language, known as HLPSL (von Oheimb, 2005). A static analysis is performed to check the executability of the protocol, and then the protocol and the intruder actions are compiled into an intermediate format (IF). This intermediate format is the start point for the four automated protocol analysis techniques. IF is a lower level language than HLPSL and it is read directly by the back-ends to the AVISPA tool. The back-ends are used to provide protocol falsification, bounded and unbounded verification. The first back-end, called the On-the-fly Model-Checker (OFMC), does several symbolic techniques to explore the state space in a demand-driven way. The second back-end, called the CL-AtSe (Constraint-Logic-based Attack Searcher), provides a translation from any security protocol specification written as transition relation in intermediate format into a set of constraints which are effectively used to find whether there are attacks on protocols. Third back-end, called the SAT-based Model-Checker (SATMC), 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. Finally, TA4SP (Tree Automata based on Automatic Approximations for the Analysis of Security Protocols) is the final back-end, which approximates the intruder knowledge by using regular tree languages.

HLPSL is a role-oriented language, in which each principal is implemented in transitional roles where the transitions of a

principal takes place during the protocol run as specified. The protocol session is considered as a parallel composition of these transitional roles. The intruder is modeled using the Dolev–Yao model (Dolev and Yao, 1983) (as in our threat model) with the possibility for the intruder to assume a legitimate role in a protocol run. The role system also defines the number of sessions, the number of principals and the roles.

### 5.2. Specifying our scheme

We have implemented our scheme in the HLPSL language. In this implementation, we have three basic roles, namely *alice, server* and *bob*, which represent the participants: the sensor node $SN_i$, the BS and the user $U_j$, respectively. We have also defined the session and environment in our scheme.

Fig. 4 illustrates the role specification for user $U_j$ in HLPSL. During the registration phase, $U_j$ sends the message $\langle U_j, RPW_j, G_{idj}, RM_{uj}\rangle$ securely to the BS with the Snd() operation. The type declaration channel (dy) indicates the channel for the Dolev–Yao threat model (as described in our threat model in Section 1.2). $U_j$ then waits for the smart card containing the secure information in the message $\langle U_j, RM_{uj}, H(\cdot), RPW_j, G_{idj}, R_{U_j}\rangle$ 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, $U_j$ sends the login request message $\langle U_j, H(R_{U_j}\|T_1), T_1)\rangle$ to the BS. In reply, the BS sends the message $\langle E_{U\ Kj}(SN_i, S_j, Z_j, K_{Uj}), T_2, T_1\rangle$ to $U_j$. During the authentication phase, $U_j$ finally sends the authentication request message $\langle SN_i, U_j, Z_j, S_j, E_{KU_j}(SN_i, U_j, R_{U_j}, G_{id_j}, T_1, S_j, Z_j), H(T_1\|S_j\|Z_j)\rangle$ to the sensor node $SNi$.

Fig. 5 shows the role specification for the BS in the HLPSL language. During the post-deployment phase, the BS receives the message $\langle \mathrm{SN}_i, T_i, E_{MK_{S_i}}(K_i, SN_i, T_i)\rangle$ from the sensor node $SN_i$. During the registration phase after receiving the message $\langle U_j, RPW_j, G_{idj}, RM_{uj}\rangle$ securely from the user $U_j$, the BS securely sends the smart card containing the information in the message $\langle U_j, RM_{uj}, H(\cdot), RPW_j, G_{idj}, R_{U_j}\rangle$ to the user $U_j$. In the login phase, when the BS receives the message $\langle U_j, H(R_{U_j}\|T_1), T_1)\rangle$ from the user $U_j$, the BS sends the messages $\langle E_{UK_j}(SN_i, S_j, Z_j, K_{U_j}), T2, T1\rangle$ to $U_j$ and $\langle SN_i, U_j, E_{MK_{S_i}}(SN_i, U_i, (APM_j \oplus G_{id_j}), R_{U_j}, T_1, T_2), E_{K_i}(SN_i, U_j, G_{id_j}, T_1)\rangle$ to the sensor node $SN_i$.

In Fig. 6, we have implemented the role specification for the sensor node $SN_i$ in the HLPSL language. In the post-deployment phase, the sensor node $\mathrm{SN}_i$ sends the message $\langle \mathrm{SN}_i, T_i, E_{MK_{S_i}}(K_i, SN_i, T_i)\rangle$ to the BS. In the login phase, the sensor

```
role bob (U, BS, SN   : agent,
        MKsi : symmetric_key,
        MKuj:  symmetric_key,
        H : hash_func,
        Snd, Rcv: channel(dy))
 played_by U
 def=
  local State  : nat,
  Uj, RPWj, APMj, RMuj, Nj, PWj, UKj : text,
  Ruj, Kuj, SNi, Sj, Zj, Ki, Kbs, GIdj, RNui :  text,
  T1, T2 : text
  const  alice_server, server_bob, bob_server,bob_alice,
  subs1, subs2, subs3, subs4, subs5, subs6 : protocol_id
  init  State := 0
  transition
   1. State   = 0 ∧ Rcv(start) =|>
     State' := 1 ∧ RPWj' := H(PWj.Nj)
              ∧ RMuj' := new( )
              ∧ Snd( U.BS.{Uj.RPWj'.GIdj.RMuj'}_MKuj)
   2. State = 1 ∧ Rcv(BS.U.{Uj.GIdj.H(H(PWj.Nj).GIdj.RMuj').
                H.H(PWj.Nj)}_MKuj) =|>
            %smart card values
     State' := 2 ∧ secret ({Ki}, subs1, {SN,BS})
              ∧ secret ({MKsi}, subs2, {SN,BS})
              ∧ secret({RMuj'},subs3,{U,BS})
              ∧ secret({Kbs},  subs4, {SN,BS})
              ∧ secret({APMj,GIdj}, subs5, {U,BS})
              ∧ secret({PWj,Nj}, subs6, U)
              ∧ T1' := new( )
             /\ Snd(U.BS.Uj.H(H(H(PWj.Nj).GIdj.RMuj ).T1' ).T1' )
             /\ witness(U, BS, bob_server, T1' )
   3. State = 2 ∧ Rcv(BS.U.{Sj.Zj.SNi.H(SNi.Uj.Kbs.Ki)}_H(H(H(PWj.Nj).
                GIdj.RMuj') .Uj.T1'.T2').T2'.T1' ) =|>
     State' := 3 ∧ UKj' := H(Ruj.Uj.T1'.T2')
              ∧ Kuj' := H(SNi.Uj.Kbs.Ki)
              ∧ Snd(U.SN.SNi.Uj.Zj.Sj.{ SNi.Uj.Ruj.GIdj.T1'.Sj.Zj}_Kuj'.
                   H(T1'.Sj.Zj))
              ∧ witness(U, SN, bob_alice, T1' )
end role
```

**Figure 4** Role specification in HLPSL for the user $U_j$ of our scheme.

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

 local State  : nat,
 RPWj, RMuj, Ruj, Kbs, Kuj, Sj,
 Zj, T2, APMj, GIdj, Nj, PWj, UKj  : text,
 SNi, Uj, Ki, Ti, T1, M3 : text
 const  alice_server, alice_bob, bob_server,
      bob_alice, subs1, subs2, subs3,
      subs4, subs5, subs6 : protocol_id
 init  State := 0
 transition
 1.  State = 0 ∧ Rcv(SN.BS.SNi.Ti.{Ki.SNi.Ti}_MKsi) =|>
    State' := 1 ∧ Kuj'  := H(SNi.Uj.Kbs.Ki)
 2.  State = 1 ∧ Rcv(U.BS.{Uj.H(PWj.Nj) .GIdj.RMuj'}_MKuj) =|>
           % user registration through secure channel
    State' := 2 ∧ Snd (BS.U.{Uj.GIdj.H(H(PWj.Nj).GIdj.RMuj').
                H.H(PWj.Nj)}_MKuj)
           ∧ secret ({Ki}, subs1, {SN,BS})
           ∧ secret ({MKsi}, subs2, {SN,BS})
           ∧ secret({RMuj'},subs3,{U,BS})
           ∧ secret({Kbs},  subs4, {SN,BS})
           ∧ secret({APMj,GIdj}, subs5, {U,BS})
           ∧ secret({PWj,Nj}, subs6, U)
           ∧ request(SN, BS, alice_server, Ti )
 3.  State =  2 ∧ Rcv(U.BS.Uj.H(H(H(PWj.Nj).
                GIdj.RMuj').T1' ).T1' ) =|>
    State' := 3  ∧ M3' := xor(APMj, GIdj)
           ∧ T2' := new( )
           ∧ UKj' := H(Ruj.Uj.T1'.T2')
           ∧ Snd(BS.U.{Sj.Zj.Kuj.SNi}_UKj.T2'.T1' )
           ∧ Snd(BS.SN.SNi.Uj.{SNi.Uj.M3'. Ruj.
                T1'.T2'}_MKsi.{SNi.Uj.GIdj.T1' }_Ki)
           ∧ witness(BS, SN, alice_server, T2' )
           ∧ request(U, BS, bob_server, T1' )
end role
```

**Figure 5** Role specification in HLPSL for the BS of our scheme.

node $SN_i$receives the message $\langle SN_i,\ U_j,\ E_{MK_{S_i}}(SN_i,U_j,(APM_j \oplus G_{idj}),\ R_{U_j},T_1,\ T_2),E_{Ki}(SN_i,\ U_j,G_{idj},T_1)\rangle$ from the BS. During the authentication phase, the sensor node receives the authentication request message $\langle SN_i,U_j,Z_j,S_j,\ E_{KUj}(SN_i,U_j,R_{U_j},G_{idj},T_1,S_j,Z_j),\ H(T_1\|S_j\ \|Z_j)\rangle$ from the user $U_j$.

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 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 roles—alice, server and bob—are instanced with concrete arguments. The top-level 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.

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

local State  : nat,
SNi, Ti, Ki,Kbs : text,
Uj, APMj, GIdj, RPWj, RMuj, T1, T2, Sj,
Zj, Ruj, Kuj, Nj, PWj, UKj  : text
const  alice_server, bob_server, alice_bob,
     bob_alice, subs1, subs2,
     subs3, subs4, subs5, subs6 : protocol_id

init  State := 0
 transition
  1. State  = 0 ∧ Rcv(start) =|>
    State' := 1 ∧ Ti' := new( )
            ∧ secret ({Ki}, subs1, {SN,BS})
            ∧ secret ({MKsi}, subs2, {SN,BS})
            ∧ secret({RMuj},subs3,{U,BS})
            ∧ secret({Kbs},  subs4, {SN,BS})
            ∧ secret({APMj,GIdj}, subs5, {U,BS})
            ∧ secret({PWj,Nj}, subs6, U)
            ∧ Snd(SN.BS.SNi.Ti.{Ki.SNi.Ti}_MKsi)
            ∧ witness(SN, BS, alice_server, Ti' )
  2.  State =  1 ∧ Rcv(BS.SN.SNi.Uj.{SNi.Uj.xor(APMj,GIdj).
              H(H(PWj.Nj).GIdj.RMuj').T1'.T2'}_MKsi.
              {SNi.Uj.GIdj.T1'}_Ki) =|>
     State' := 2 ∧ request(BS, SN, alice_server, T2' )
  3. State = 2  ∧ Rcv(U.SN.SNi.Uj.Zj.Sj.
              {SNi.Uj.H(H(PWj.Nj).GIdj.RMuj').
              GIdj.T1'.Sj.Zj}_ H(SNi.Uj.Kbs.Ki).
              H(T1'.Sj.Zj)) =|>
    State' := 3 ∧ request(U, SN, bob_alice, T1')
end role
```

**Figure 6** Role specification in HLPSL for the sensor $SN_i$ of our scheme.

The current version of HLPSL supports the standard authentication and secrecy goals. In our scheme, six secrecy goals and four authentications are verified. We simulated our scheme for OFMC and CL-AtSe back-ends using the AVISPA web tool (AVISPA, 2013). The simulation results are shown in Figs. 9 and 10. The summary of the results are as follows:

- OFMC reports the protocol is safe.
- CL-AtSe reports the protocol is safe.

Thus, it is clear that our scheme is secure against passive and active attacks, including the replay and man-in-the-middle attacks.

```
role session(SN, BS, U : agent,
        % H is hash function
        MKsi : symmetric_key,
        MKuj : symmetric_key,
        H : hash_func )
def=
local US, UR, SS, SR, VS, VR: channel (dy)
composition
  alice(SN, BS, U, MKsi, H, US, UR)
∧ server(BS, U, SN, MKsi, MKuj, H, SS, SR)
∧ bob(U, BS, SN, MKsi, MKuj,  H, VS, VR)
end role
```

**Figure 7** Role specification in HLPSL for the session of our scheme.

```
role environment()
def=
const sn, bs, u : agent,
    mksi : symmetric_key,
    mkuj : symmetric_key,
    h : hash_func,
    rpwj, ruj, sj, zj, kuj, ki, rmuj, ti, t1,
    t2, apmj, gidj, kbs, sni, uj : text,
alice_server, alice_bob, bob_server, bob_alice,
subs1, subs2, subs3, subs4, subs5, subs6 : protocol_id

intruder_knowledge = {u, bs, sn , h, uj , sni, uj}

composition
session(sn, u, bs, mksi, mkuj, h) ∧
session(u, sn, bs, mksi, mkuj, h) ∧
session(u, sn, bs, mksi, mkuj, h)
end role

goal
secrecy_of subs1
secrecy_of subs2
secrecy_of subs3
secrecy_of subs4
secrecy_of subs5
secrecy_of subs6

authentication_on alice_server
authentication_on bob_server
authentication_on bob_alice
authentication_on alice_bob
end goal

environment()
```

**Figure 8** Role specification in HLPSL for the goal and environment of our scheme.

## 6. Performance comparison with other related schemes

This section compares the performance of our scheme with relevant 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).

### 6.1. Comparison of computational costs

We have used the notations for computational cost comparisons between our scheme and other schemes provided in Table 8. $t_{ecm}$, $t_{eca}$, $t_i$, $t_{add}$, $t_{mul}$, $t_h$, $t_{enc}$, $t_{dec}$, $t_{ecenc}$, $t_{ecdec}$, $t_{mac}$, $t_{siggen}$, 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 considered the time taken for one MAC operation as that for one hashing operation. The quantitative analysis of Koblitz et al. (2000) shows that the computation of a multiplication point requires approximately 1200 field multiplications; an elliptic curve point addition requires one field inversion and two field multiplications; the computation of a field inversion requires approximately three field multiplications; the computation of elliptic curve encryption and decryption require approximately 2405 and 1205 field multiplications, respectively (DeWin et al., 1996; Schroeppel et al., 1995); and the cost of field addition is negligible. Furthermore, a 1024-bit modular multiplication takes 41 times longer than a field multiplication in a finite field GF ($2^{163}$). The results of Wong et al. (2001) show the speed for AES encryption and decryption, hash function using SHA-1 and 1024-bit modular multiplication. In Table 8, the time complexity of various operations in terms of $t_{mul}$ are listed according to the analysis results reported in Wu and Chen (2012).

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-computation/
  ./tempdir/workfileu1mRmM.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 5.18s
  visitedNodes: 472 nodes
  depth: 9 plies
```

**Figure 9** The result of the analysis using OFMC.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/avispa/web-interface-computation/
  ./tempdir/workfileu1mRmM.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed  : 15 states
  Reachable : 15 states
  Translation: 0.25 seconds
  Computation: 0.00 seconds
```

**Figure 10** The result of the analysis using CL-AtSe.

**Table 8** 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 | $t_h \approx 0.36t_{mul}$ | $t_{enc} \approx 0.15t_{mul}$ |
| $t_{dec} \approx 0.15t_{mul}$ | $t_{ecenc} \approx 2405t_{mul}$ | $t_{ecdec} \approx 1205t_{mul}$ |
| $t_{mac} \approx t_h$ | $t_{siggen} \approx 1204.36t_{mul}$ | $t_{eca} \approx 5t_{mul}$ |

We have compared the computational complexity using both formulated results and a rough quantitative analysis in Table 9 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.

### 6.2. Comparison of communication costs

In Table 10, we compared the communication costs between our scheme and the other related schemes (Le et al., 2009; Wang et al., 2006; Mahmud and Morogan, 2012) in terms of the total number of bits and the total number of packets required for transmissions during all phases. The table shows that our scheme requires six message exchanges; among those where a sensor node is directly involved, only one message transmission is required, compared with the other schemes where a sensor node is directly involved. As a result, our scheme is significantly efficient in terms of communication costs as compared with the other related schemes.

$I_1$: Total number of bits transmission required for messages of all phases for the schemes; $I_2$: Total number of packets transmissions during all phases for the schemes; $I_3$: Total number of message transmissions during all phases for the schemes.

We further calculated the total number of bits required for all of the messages during all phases for the access control

**Table 9** 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 | $U_j$ | - | - | - | $t_h$ |
| | BS | $2t_{ecm} + t_{siggen}$ | $t_h + 3t_{ecm} + t_{mul} + t_{eca}$ | $t_h$ | $t_h$ |
| | $SN_i$ | - | - | - | - |
| Login + Authentication | $U_j$ | $t_h + t_{sigver} + t_{mac}$ | $t_{ecm} + 2t_{mac}$ | $t_h + t_{sigver}$ | $6t_h + t_{dec} + t_{enc}$ |
| | BS | $2t_{sigver} + 2t_{mac} + 2t_h$ | - | - | $3t_h + 2t_{ecm} + 3t_{enc} + t_{ena}$ |
| | $SN_i$ | $3t_{mac} + t_h$ | $t_{eca} + 3t_{ecm} + t_h + 2t_{mac}$ | $2t_h + t_{siggen} + t_{sigver}$ | $3t_h + t_{ecm} + 3t_{dec}$ |
| | Total Cost | $4t_h + 2t_{ecm} + 4t_{hsigver} + 6t_{mac}$ | $2t_h + 7t_{ecm} + t_{mul} + 2t_{eca} + 4t_{mac}$ | $4t_h + 2t_{siggen} + 2t_{sigver}$ | $14t_h + 8t_{enc}/t_{dec} + 3t_{ecm} + t_{eca}$ |
| | Rough Estimation | $12025.04t_{mul}$ | $8413.16t_{mul}$ | $7220.88t_{mul}$ | $3611.24t_{mul}$ |

**Table 10** Comparison of communication costs between the proposed scheme and the other schemes.

| Scheme | $I_1$ | $I_2$ | $I_3$ |
|---|---|---|---|
| Le et al. (2009) | 2208 | 7 | 7 |
| Mahmud and Morogan (2012) | 1132 | 5 | 5 |
| Wang et al. (2006) | 2544 | 6 | 6 |
| Ours | 1400 | 6 | 6 |

schemes. We also calculated the number of packets required for transmission of a message for the CC2420 transceiver (CC2420:2.4 GHz IEEE 802.15.4, 2011) which supports a packet size of 128 bytes, i.e., 1024 bits. The results shown in Table 10 demonstrate that our scheme is also efficient compared with other related schemes.

### 6.3. Comparison of energy costs

Because sensor nodes are resource-constrained, we primarily considered the energy costs of a sensor node during the login and authentication phases. We compared the energy costs of a sensor node during the login and authentication phases between our scheme, Le et al.'s scheme (Le et al., 2009), Mahmud–Morogan's scheme (Mahmud and Morogan, 2012) and Wang et al.'s scheme (Wang et al., 2006) in Table 11. As in Chatterjee et al. (in press), Das et al. (2012b), the energy costs of a sensor node consider both the computational and communication costs involved during the login and authentication phases. In wireless communication, the energy for sensor nodes primarily goes towards the transmission and reception of messages/packets rather than computing. Because our scheme requires no message or packet transmissions during the login and authentication phases (compared with the other schemes), the energy spent by sensor nodes is significantly less compared with those schemes.

### 6.4. Comparison of functionality

This section compares the functionality of our scheme with schemes (Le et al., 2009; Wang et al. (2006); Mahmud and Morogan (2012)) in Table 12. It is noted that Le et al.'s scheme (Le et al., 2009) is based on ECC; it supports session key establishment between the user and the sensor node and mutual authentication between the user and the sensor node. Their scheme does not support a user's password change or a dynamic sensor node addition phase after initial deployment. In Wang et al.'s scheme (Wang et al., 2006), ECC is used as the cryptographic technique. It supports session key establishment between the user and the sensor node and mutual authentication between the user and the sensor node, but it does not support a user's password change or dynamic sensor node addition phase after initial deployment. In Mahmud–Morogan's scheme (Mahmud and Morogan, 2012), an identity-based signature approach with ECC is the basis for the cryptographic technique. As in other schemes, their scheme supports session key establishment between the user and the sensor node and mutual authentication between the user and the sensor node, but it does not support a user's password change or dy-

**Table 11** Comparison of energy costs of a sensor node during the login and authentication phases between our scheme and other schemes.

| Scheme | Sensor node's energy cost |
|---|---|
| Le et al. (2009) | three MAC operations + one hash operation + three message transmissions |
| Mahmud and Morogan (2012) | one ECC-point addition + three ECC-point multiplication + one hash operation + two MAC operations + three message transmissions |
| Wang et al. (2006) | two hash operations + one ECC-signature generation + two ECC-signature verifications + two message transmissions |
| Ours | three hash operations + one ECC-point multiplication + three symmetric-key decryptions + no message transmissions |

**Table 12** Comparison of functionality analysis between the proposed scheme and the other schemes.

| Scheme | Le et al. (2009) | Wang et al. (2006) | Mahmud and Morogan (2012) | Ours |
|---|---|---|---|---|
| Cryptographic technique | ECC | ECC | IBS with ECC | Hybrid (ECC with symmetric-key cryptosystem) |
| Session key establishment | Supported | Supported | Supported | Supported |
| User password change | Not available | Not available | Not available | Supported |
| Dynamic sensor node addition | Not available | Not available | Not available | Supported |
| Mutual authentication between user and sensor node | Supported | Supported | Supported | Supported |

namic sensor node addition phase after initial deployment. Our scheme uses a hybrid approach of both ECC and symmetric-key cryptosystem (AES) for communication and computational efficiency, compared with other schemes. Our scheme supports session key establishment between the user and the sensor node and mutual authentication between the user and the sensor node. In addition, our scheme supports a user's password change and a dynamic sensor node addition phase after initial deployment, which are important requirements for an ideal user access scheme designed for WSNs. Furthermore, our scheme provides for mutual authentication between the BS and the sensor nodes.

## 7. Conclusion

This paper proposed a new user access control scheme suitable in wireless body area networks for healthcare and patient monitoring applications. The proposed scheme allowed the user to authenticate at the sensor node inside a WBAN under certain access privileges. After successful authentication, both the user and the sensor node from which the user wants to access real-time data can establish a secret session key between them. By using this session key, the user can later contact the sensor node for the real-time data inside the WBANs. Our scheme provides unconditional security against node capture attack and also prevents other known attacks such as denial-of-service, masquerade, stolen-verifier, many logged-in users with the same login-ID, replay, privileged insider, smart card breach, and man-in-the-middle attacks. The proposed scheme supports a dynamic node addition phase; there is no need to update stored information in the user's smart card for accessing real-time data from the added/replaced sensor nodes in the network. Using a AVISPA tool, we showed that our scheme is secure against both passive and active attacks, including the replay and man-in-the-middle attacks. Our scheme also supports other features such as freely and locally changing the password by the user without contacting the BS for any security reasons, and other existing schemes do not support this feature. Our scheme also supports a dynamic sensor node addition after initial deployment, whereas other existing approaches do not have this important feature. Our scheme is also efficient in terms of communication, computation, storage and energy overheads. Overall, the higher security and the lower communication and computational costs make our scheme much more appropriate for practical applications in the emerging healthcare field compared with other existing approaches.

## Acknowledgements

The authors would like to acknowledge the many helpful suggestions of the anonymous reviewers and the Editor-in-Chief, which have improved the content and the presentation of this paper.

## References

Advanced Encryption Standard, 2007. FIPS PUB 197, National Institute of Standards and Technology (NIST), US Department of Commerce (November 2001). <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.

Alemdar, H., Ersoy, C., 2010. Wireless sensor networks for healthcare: a survey. Computer Networks 54 (15), 2688–2710.

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), 93–101.

Armando, A. et al., 2005. The AVISPA tool for the automated validation of internet security protocols and applications. In: Computer Aided Verification (CAV), LNCS, vol. 3576, pp. 281-285.

Atmel Corporation. Available from: <http://www.atmel.com> a(accessed November 2010).

Aumasson, J.P., Henzen, L., Meier, W., Plasencia, M.N., 2010. Quark: a lightweight hash. In: Workshop on Cryptographic Hardware and Embedded Systems (CHES 2010), LNCS, vol. 6225, pp. 1–15.

AVISPA. AVISPA Web Tool. <http://www.avispa-project.org/web-interface/expert.php/> (accessed January 2013).

Basin, D., Modersheim, S., Vigano, L., 2005. OFMC: a symbolic model checker for security protocols. International Journal of Information Security 4 (3), 181–208.

Carman, D.W., Kruus, P.S., Matt, B.J., 2001. Constraints and approaches for distributed sensor network security (Dated September 1, 2000). NAI Labs Technical Report No. 00-010.

CC2420:2.4 GHz IEEE 802.15.4/ ZigBee-Ready RF Transceiver. Available from: <http://www.ti.com/product/cc2420> (accessed September 2011).

Chatterjee, S., Das, A.K., Sing, J.K., in press. An enhanced access control scheme in wireless sensor networks. Ad Hoc & Sensor Wireless Networks.

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. 1–10.

Das, M.L., 2009. Two-factor user authentication in wireless sensor networks. IEEE Transactions on Wireless Communications 8 (3), 1086–1090.

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), 189–211.

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), 12–27.

Das, A.K., Paul, N.R., Tripathy, L., 2012a. Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem. Information Sciences 209, 80–92.

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), 1646–1656.

Das, A.K., Massand, A., Patil, S., 2013. A novel proxy signature scheme based on user hierarchical access control policy. Journal of King Saud University – Computer and Information Sciences 25 (2), 219–228.

DeWin, E., Bosselaers, A., Vandenberghe, S., De Gersem, P., Vandewalle, J., 1996. A fast software implementation for arithmetic operations in GF (2n). In: Proceedings of Advances in Cryptology – ASIACRYPT '96. Lecture Notes in Computer Science, vol. 1163. Springer-Verlag, pp. 65–76.

Diffie, W., Hellman, M.E., 1976. New directions in cryptography. IEEE Transactions on Information Theory 22, 644–654.

Dolev, D., Yao, A., 1983. On the security of public key protocols. IEEE Transactions on Information Theory 29 (2), 198–208.

Fan, R., Ping, L.-D., Fu, J.-Q., Pan, X.-Z., 2010. A secure and efficient user authentication protocol for two-tiered wireless sensor networks. In: Second Pacific–Asia Conference on Circuits, Communications and System (PACCS'10), pp. 425–428.

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), 66–77.

Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C., 2004. Comparing elliptic curve cryptography and RSA on 8-bit CPUs. In: Proceedings of Sixth International Workshop on Cryptographic Hardware and Embedded Systems (CHES'04).

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, 3473–3481.

Johnson, D., Menezes, A., 1999. The Elliptic Curve Digital Signature Algorithm (ECDSA). Technical Report CORR 99-34, Dept. of C & O, University of Waterloo, Canada, August 23, 1999.

Klaoudatou, E., Konstantinou, E., Kambourakis, G., Gritzalis, S., 2011. A survey on cluster-based group key agreement protocols for WSNs. IEEE Communications Surveys and Tutorials 13 (3), 429–442.

Koblitz, N., 1987. Elliptic curves cryptosystems. Mathematics of Computation 48, 203–209.

Koblitz, N., Menezes, A., Vanstone, S.A., 2000. The state of elliptic curve cryptography. Designs, Codes and Cryptography 19 (2-3), 173–193.

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 ICCIT'09.

Latre, B., Braem, B., Moerman, I., Blondia, C., Demeester, P., 2011. A survey on wireless body area networks. Wireless Networks 17 (1), 1–18.

Le, X.H., Lee, S., Butun, I., Khalid, M., Sankar, R., Kim, M., Han, M., Lee, Y.-K., Lee, H., 2009. An energy-efficient access control scheme for wireless sensor networks based on elliptic curve cryptography. Journal of Communications and Networks 11 (6), 599–606.

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. 388–396.

Liao, H.Z., Shen, Y.Y., 2006. On the elliptic curve digital signature algorithm. Tunghai Science 8, 109–126.

Li, M., Lou, W., Ren, K., 2010. Data security and privacy in wireless body area networks. IEEE Wireless Communications, 51–58.

Mahmud, A. Al., Morogan, M.C., 2012. Identity-based authentication and access control in wireless sensor networks. International Journal of Computer Applications 41 (13), 18–24.

Malan, D.J., Welsh, M., Smith, M.D., 2004. A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In: Proceedings of First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON'04), Santa Clara, California, USA.

Manuel, S., 2011. Classification and generation of disturbance vectors for collision attacks against SHA-1. Designs, Codes and Cryptography 59 (1-3), 247–263.

Odelu, V., Das, A.K., Goswami, A., 2013. An effective and secure key-management scheme for hierarchical access control in e-medicine system. Journal of Medical Systems 37 (2), 1–18.

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), 307–326.

Rivest, R.L., Shamir, A., Adleman, L.M., 1978. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21, 120–126.

Rivest, R.L., Hellman, M.E., Anderson, J.C., Lyons, J.W., 1992. Responses to NIST's proposal. Communications of the ACM 35 (7), 41–54.

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

Schroeppel, R., Orman, H., O'Malley, S., Spatscheck, O., 1995. Fast key exchange with elliptic curve systems. In: Proceedings of Advances in Cryptology – CRYPTO '95. Lecture Notes in Computer Science, vol. 963. Springer-Verlag, pp. 43–56.

Secure Hash Standard. FIPS PUB 180-1, National Institute of Standards and Technology (NIST), US Department of Commerce, April 1995.

Seyedi, M., Kibret, B., Lai, D., Faulkner, M., 2013. A survey on intrabody communications for body area network applications. IEEE Transactions on Biomedical Engineering.

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.

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

Stinson, D.R., 2006. Some observations on the theory of cryptographic hash functions. Designs, Codes and Cryptography 38 (2), 259–277.

Venkatasubramanian, K.K., Banerjee, A., Gupta, S.K.S., 2010. PSKA: usable and secure key agreement scheme for body area networks. IEEE Transactions on Information Technology in Biomedicine 14 (1), 60–68.

von Oheimb, D., 2005. The high-level protocol specification language HLPSL developed in the EU project AVISPA. In: Proceedings of APPSEM Workshop.

Wang, H., Sheng, B., Li, Q., 2006. Elliptic curve cryptography-based access control in sensor networks. International Journal of Security and Networks 1 (3/4), 127–137.

Wang, H., Sheng, B., Tan, C.C., Li, Q., 2008. Comparing symmetric-key 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.

Watro, R., Kong, D., Cuti, S., Gardiner, C., Lynn, C., Kruus, P., 2004. TinyPK: securing sensor networks with public key technology. In Proceedings of the 2nd ACM Workshop on Security of ad hoc and Sensor Networks (SASN'04), Washington, DC, USA, October 2004, pp. 59–64.

Wen, M., Lei, J., Li, J., Wang, Y., Chen, K., 2011. Efficient user access control mechanism for wireless multimedia sensor networks. Journal of Computational Information Systems 7 (9), 3325–3332.

Wong, D.S., Fuentes, H.H., Chan, A.H., 2001. The performance measurement of cryptographic primitives on palm devices. In: Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC 2001), pp. 92–101.

Wu, S., Chen, K., 2012. An efficient key-management scheme for hierarchical access control in e-medicine system. Journal of Medical Systems 36 (4), 2325–2337.

Zhang, Z., Wang, H., Vasilakos, A.V., Fang, H., 2012. ECG-Cryptography and Authentication in Body Area Networks. IEEE Transactions on Information Technology in Biomedicine 16 (6), 1070–1078.

Zois, D.S., Levorato, M., Mitra, U., 2012. A POMDP framework for heterogeneous sensor selection in wireless body area networks. In: INFOCOM 2012, pp. 2611–2615.