Computer Networks 129 (2017) 429–443

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks

Xiong Li[a,*], Maged Hamada Ibrahim[b], Saru Kumari[c], Arun Kumar Sangaiah[d], Vidushi Gupta[e], Kim-Kwang Raymond Choo[f]

[a] School of Computer science and Engineering, Hunan University of Science and Technology, Xiangtan, 411201, China
[b] Department of Electronics, Communication & Computers, Faculty of Engineering, Helwan University, Helwan, P.O.11792, Cairo, Egypt
[c] Department of Mathematics, Ch. Charan Singh University, Meerut 250 005, Uttar Pradesh, India
[d] School of Computer science and Engineering, VIT University, Vellore-632014, Tamilnadu, India
[e] Department of Computer Science, NSIT, University of Delhi, New Delhi, India
[f] Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, Texas, USA

ARTICLE INFO

Article history:
Received 26 November 2016
Revised 16 February 2017
Accepted 15 March 2017
Available online 16 March 2017

Keywords:
WBAN
Healthcare
Wearable sensors
Mutual authentication
Anonymity
AVISPA

ABSTRACT

Wireless body area networks (WBANs) are used to collect and exchange vital and sensitive information about the physical conditions of patients. Due to the openness and mobility of such networks, even without knowing the context of the exchanged data or linking traffic to the identities of involved sensors, criminals are able to gain useful information about the severe conditions of patients and carry effective undetectable physical attacks. Therefore, confidentiality and mutual authentication services are essential for WBANs, and the transmission must be anonymous and unlinkable as well. Given the limitations of the resources available for these sensors, a lightweight anonymous mutual authentication and key agreement scheme for centralized two-hop WBANs is proposed in this paper, which allows sensor nodes attached to the patient's body to authenticate with the local server/hub node and establish a session key in an anonymous and unlinkable manner. The security of our scheme is proved by rigorous formal proof using BAN logic and also through informal analysis. Besides, the security of our scheme is evaluated by using the Automated Validation of Internet Security Protocols and Applications (AVISPA) as well. Finally, we compare our proposed scheme with other related schemes and the comparison results show that our scheme outperforms previously related schemes.

1. Introduction

The advances in microelectronics and embedded technologies resulted in the advent of miniature and ultra-low power sensors or wearable devices with the ability to sense, process and transmit. A Wireless Body Area Network (WBAN) [1] is formed by various wearable sensors that are situated in the body of patient, where the nodes are connected via wireless communication technologies [2,3]. WBAN can be used to monitor and track physical conditions of patients without reducing the user's comfort by using an Intranet or Internet. WBANs can provide many applications combine with cloud computing technology [4–7], such as vital sign monitoring, home care monitoring, clinical monitoring, and sportsperson health condition monitoring [8]. For example, a WBAN on a diabetic patient could auto inject insulin through a pump as soon as the insulin level declines. WBANs have broad prospects in the medical field since they can reduce the tasks of healthcare providers, eliminate medical errors, increase efficiency of hospital staff, reduce long-term cost of healthcare services, and improve the comfort of the patients.

In 2012, IEEE published a standard IEEE 802.15.6 for WBANs. This standard has adopted a multi-hop centralized architecture, where a special central node, called local server or hub node is involved in this architecture, and all the monitored information of sensor nodes would be sent to it. The main merit of this architecture is the simplification of the control and management of nodes. However, nodes that are far away from the hub require higher energy for communication, and it could be harmful to the patient, specially when the nodes are attached or implanted inside the patient's body. An extended architecture for WBANs with a two-hop centralized architecture is given in Fig. 1. There are three types of nodes, i.e. hub node, first level nodes and the second level

* Corresponding author.
E-mail addresses: lixiongzhq@163.com, lixiong84@gmail.com (X. Li), saryusiirohi@gmail.com (S. Kumari), arunku-marsangaiah@gmail.com (A.K. Sangaiah), raymond.choo@fulbrightmail.org (K.-K.R. Choo).

http://dx.doi.org/10.1016/j.comnet.2017.03.013
1389-1286/

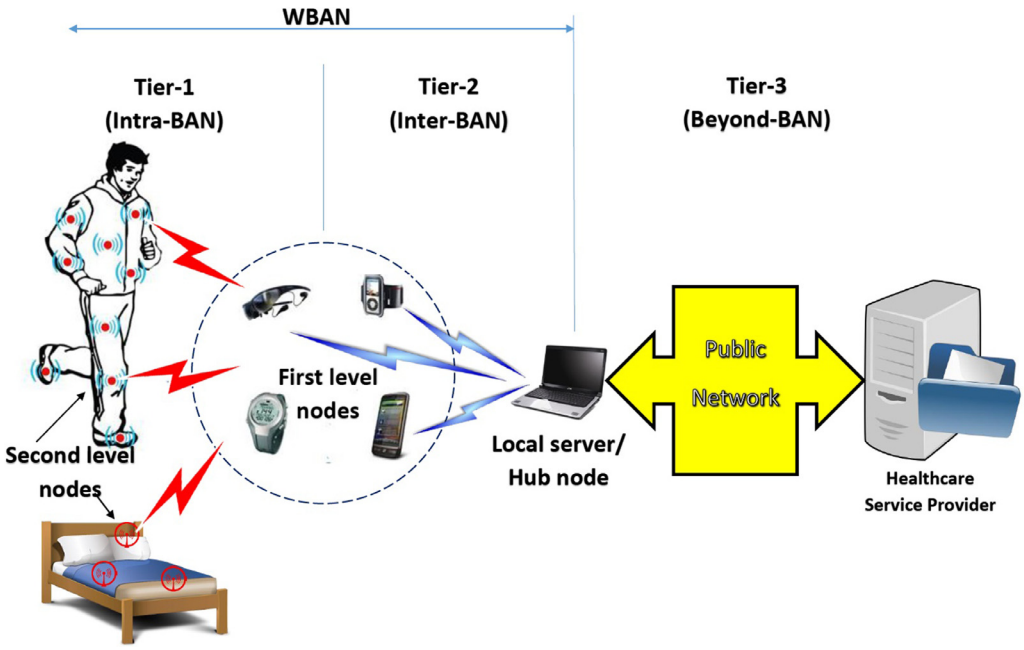

Fig. 1. Architecture for centralized two-hop WBAN.

nodes. The sensor nodes or wearable devices attached to the patient's body represent the second level of this network, and the first level nodes are super nodes with more storage, processing, computing and communication capabilities than the second level nodes. These first level nodes are used as intermediate nodes, and the data collected by the second level nodes is forwarded to the hub node via the first level nodes. Besides, the architecture can be regarded as three tiers as shown in Fig. 1. Tier 1 represents the connection between second level node (wearable sensor) and first level node (intermediate node), and the communication of this tier is also called Intra-BAN communication. Similarly, tier 2 connects first level node (intermediate node) and hub node, and the communication of this tier can be called as Inter-BAN communication. Tier 3 is beyond the WBAN, which connects hub node and healthcare service provider, and provides medical service for users.

The messages exchanged in the WBAN contain vital and sensitive information about the physical conditions of patients, and this information is important for patient's privacy. Therefore, security and privacy protection is an important issue which need to be solved in WBAN, such as the patient's data should only derived from each patient's dedicated WBAN system, and the use of WBAN data should be guaranteed by a robust authorization control strategy. Moreover, the openness and mobility of WBAN lead to the security of this kind network in a more dangerous position. For example, an adversary may track a particular patient by linking traffic to the same sensor node of the patient, even without recognizing the context of this traffic. Besides, an adversary may gain some critical information about the physical condition of a patient may by using traffic analysis, and these information allows the adversary to launch physical attacks against this patient. Meanwhile, due to the wearable sensors in WBAN are resource-constrained in terms of power, memory, communication rate and computational capability, security mechanisms proposed for other networks may not be applicable to WBAN. Therefore, in the design of security and privacy mechanisms for WBANs, the conflicts among security, efficiency, and practicality must be considered carefully. Stringent resource constraints on wearable devices in a WBAN requires the security mechanisms to be as lightweight and low cost as possible.

1.1. Related work

Authentication and key agreement is an essential and vital mechanism for network and information security, which allows the server to confirm user's identity when the user accesses the system. Many user authentication schemes have been proposed by researchers for different environments, such as single server environments [9–12], multi-server environments [13,14] and wireless sensor networks [15,16]. The authentication and key agreement mechanism for a WBAN is relatively a fresh thing, and few articles have discussed this research topic in recent years. To guarantee the secure communication in WBAN, some non-cryptographic authentication and key agreement schemes have been proposed, such as physiological signal based schemes [17,18], channel-based schemes [19,20], and the proximity-based schemes [21,22]. However, physiological signal based schemes [17,18] are based on the assumption that the wearable sensors can measure the same type of electrocardiogram parameters, and therefore the application scope of this approach is limited. Besides, this type of scheme may be vulnerable to denial-of-service attack because there may be a difference in the physiological signals of the same person measured by different devices. Channel-based schemes [19,20] either cannot provide anonymity or requires special hardware or software devices, and are impractical for WBANs. Proximity-based authentication methods [21,22] require that the devices must be within half of the wavelength distance of each other, which limits its application in WBAN. Compared with these schemes, cryptography based schemes [23–30] have fewer hardware and software restrictions on wearable sensors in WBAN, and receiving researches' attention. In 2013, Li et al. [23] proposed an authenticated key agreement, and the wearable devices can authenticate each other with the help of the user. However, the requirement for modular exponentiation operation makes their scheme is unsuitable for resource limited wearable devices. In 2014, Liu et al. [24] proposed two certificateless remote anonymous authentication schemes for WBAN. However, Zhao [25] pointed out that the schemes in [24] are vulnerable to stolen verifier attacks, and proposed an enhanced scheme. At the same time, Xiong [26] found that Liu's schemes [24] are inefficiency of certificate managements, and lack of scalability and

forward secrecy. They proposed an scalable and anonymous certificateless remote authentication protocol for WBAN [26], which not only improved security features but also reduced the communication and computation overload. In 2015, He et al. found that Zhao's scheme [25] cannot provide the unlinkability of a user, and they proposed an authentication protocol for an ambient assisted living system. The scheme in [27] focused on the authentication beyond the WBAN, where the AAL server authenticates the user to the local server. However, their work did not consider authentication in the second tier between local server and body sensors. In 2015, Xiong and Qin [31] proposed a scalable certificateless remote authentication protocol for WBAN to resolve the revocation of a certificate. In 2016, He et al. [29] found the scheme in [24] is vulnerable to an impersonation attack, and they proposed an anonymous authentication scheme for WBAN. At the same time, Liu et al. [30] proposed a 1-round anonymous authentication protocol for WBAN. However, the schemes of Liu et al. [30] and He et al. [29] are also just for the third tier and did not consider any authentication services in the second tier between local server and body sensors. In the above reviewed authentication schemes for WBAN either no revocation procedure was presented for revoking the user's privilege of getting the services, or the schemes were lacking anonymity. Besides, they are all bearing heavy computational costs such as bilinear pairing, and not lightweight enough for wearable sensors. Furthermore, among these schemes, few of them have considered the anonymity authentication of the body sensors in WBAN.

1.2. Motivations and contributions

Although some authentication schemes for WBAN have been proposed by researchers, they are not lightweight enough to be suitable for WBAN sensor nodes due to the high resources constraints of such sensors. Besides, few mutual authentication and key agreement schemes have been proposed for two-hop centralized WBAN. Moreover, almost none of these contributions considered anonymity of the sensor nodes. Motivated by the importance of anonymous authentication service for WBAN with the feature of unlinkability of data transmission, this paper devoted to design a secure scheme not only provides mutual authentication, but also achieves anonymity and unlinkability of transmitted information. The contributions of this paper are as follows:

- We propose a lightweight and secure anonymous mutual authentication and key agreement scheme for two-hop wireless body area networks with a centralized architecture.
- The security of our protocol is not only proved by using the widely accepted BAN logic, but also assessed by using the AVISPA simulator tool. Besides, informal security analysis of the proposed scheme is discussed.
- We compare the proposed scheme with other related schemes, and the comparison results show that our scheme is superior to previously related schemes since the storage requirements, computation cost and energy consumption of the proposed scheme are much less than that required by previously proposed schemes.

1.3. Organization of the paper

The remaining parts of this paper are arranged as follows. Section 2 gives the used system model for WBAN in this paper, which contains the network model and adversary model. A lightweight and anonymous mutual authentication scheme for two-hop WBAN is proposed in Section 3. Section 4 proves the security of the proposed scheme using BAN logic, AVISPA simulator tool, and informal security analysis, respectively. Section 5 evaluates the performance of the proposed scheme and compares it with other related schemes. Finally, the conclusion is given in Section 6.

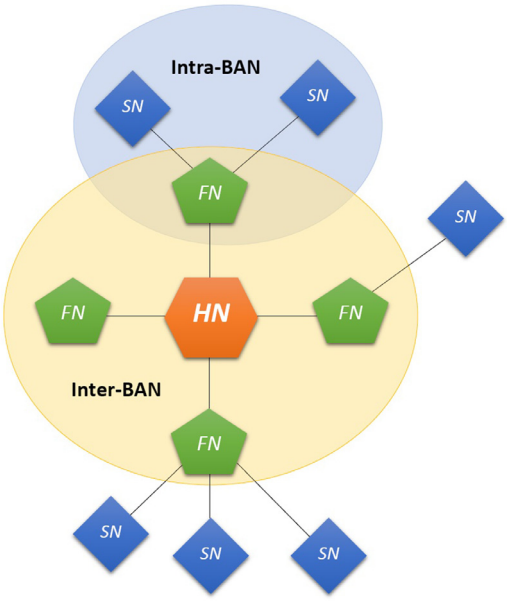

Fig. 2. Network model for WBAN. *HN*: Hub node, *FN*: First level node, *SN*: Second level node.

2. System model

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

2.1. Network model

The network mode for WBAN is a two-hop (two tiers) centralized network as shown in Fig. 2. It contains three types of nodes, i.e. the hub node (*HN*), first level node (*FN*) and second level node (*SN*). The HN is sometimes called a local server, which is rich in resources and can be a PC or a server station. The second level nodes (*SN*) are resource-constrained wearable sensors attached to the human body. The first level node (*FN*) can be seen as the intermediate node between *HN* and *FN*, which has more communication, computing, and storage capabilities than *SN*. The hub node and the first level nodes represent the Inter-BAN as tier-2 of the network, while the connection of the first level and the second level nodes represent the Intra-BAN as tier-1 of the network. The second level nodes (*SN*) in Intra-BAN monitor the patient's vital signals and forward real time information to *HN* via a first level node (*FN*). *HN* holds patient's authentication information, and when collecting physiological vital signals from Intra-BAN, *HN* processes them, and prioritizes the transmission of critical data over public communication networks to the healthcare service provider medical servers.

It is assumed that, the hub node and the first level nodes are always in-range (i.e. any first level node is able to communicate directly with the hub node), but the hub node is out of range of the second level nodes, due to the limitations on the transmitted power. Therefore, for a second level node to communicate with the hub node, we assume that the second level nodes are always in-range with at least one first level node and hence, the first level node is able to relay traffic between this second level node and the hub node. In this case, we call the first level node an intermediate node.

Table 1
Notations used in our protocol .

Symbol	Description
SA	System administrator.
N	Sensor node requesting authentication
HN	Hub node
IN	Intermediate first level sensor node
id_N	Real/permanent identity/secret key of sensor node N
id'_N	Real/permanent identity of first level sensor node.
tid_N	Temporary identity of sensor node N
k_{HN}	Master secret key of HN
k_N, f_N	Temporary secret parameters picked by HN for N.
r_N	Temporary secret parameter picked by N.
a_N, b_N	Authentication parameters stored in N's memory.
x_N, y_N	Auxiliary parameters required for authentication.
α, β, γ, η, μ	Authentication parameters computed by HN to authenticate with N.
k_s	Session key to be agreed on.
t_N	A timestamp generated by node N.
$h(\cdot)$	Collision-resistant one-way cryptographic hash function.
(a, b)	Concatenation of data a and data b.
$\oplus$	Bitwise XOR operation.
$X \rightarrow Y$: Z	Entity X sends the message Z to entity Y via a public channel

2.2. Adversary model

In order to evaluate the security features of the proposed scheme, we define the adversarial model as follows.

- The hub node HN is assumed trustworthy. However, an adversary may be able to infiltrate HN's database. She may steal or manipulate database information. The only parameter that is assumed completely beyond the reach of the adversary is the HN's master secret key.
- The adversary is able to eavesdrop on all communication links in the network. She can also corrupt or replace transmitted messages or replay previously transmitted old messages.
- An adversary is able to capture any sensor node N. Which means, she is able to extract all secret information stored in N's memory. The consequences of capturing any fraction of the sensor nodes must not threaten the security of any other uncaptured node in the network.
- We use the well-known Dolev-Yao threat model [32] in which the model assumes that two communicating parties communicate over an insecure channel. We provide the security analysis and simulation of our scheme assuming this model.

3. Our proposed scheme

The important notations used in the description of our scheme are given in Table 1. Our scheme consists of three phases: Initialization phase, registration phase and authentication phase. The initialization and registration phases are performed by the system administrator (SA). In the authentication phase, a sensor node N engages with the Hub node HN for secure anonymous mutual authentication and session key exchange/agreement. We assume N is a second level node and hence, it communicates with HN through an intermediate first level node IN. In case N is a first level node, the scheme can be easily adapted to allow direct communication with HN by removing the intermediate node IN. Let h: $\{0, 1\}^* \rightarrow \{0, 1\}^\ell$ be a strong hash function. The phases of our scheme are described in details in the following subsections.

3.1. Initialization phase

The system administrator (SA) initializes the hub node (HN) as follows.

Step I1 Picks a master secret key k_{HN} for HN.

Step I2 Stores k_{HN} in HN's memory.

3.2. Registration phase

SA registers a sensor node (N) as follows.

Step R1 Picks a unique secret identity id_N for N.
Step R2 Picks k_N for N.
Step R3 Computes $a_N = id_N \oplus h(k_{HN}, k_N)$ and $b_N = k_{HN} \oplus a_N \oplus k_N$.
Step R4 Picks another short unique identity id'_N for the first level node FN.
Step R5 Stores the tuple $\langle id'_N, id_N, a_N, b_N \rangle$ in first level node FN's memory and the tuple $\langle id_N, a_N, b_N \rangle$ in the second level node SN's memory.
Step R6 Stores the real identity id'_N in HN's memory for first level nodes FN.

Remark 1. Notice that, k_N is not required to be stored at all, neither at the sensor node nor at the hub node. It is only used to create a_N and b_N.

Remark 2. The identity id_N represents the permanent real identity and the secret key for node N.

Remark 3. The other identity id'_N is used by a first level node FN only when it is operating as an intermediate node.

3.3. Authentication phase

The authentication phase is shown in Fig. 3. A second level sensor node N anonymously authenticates with the hub node HN through an intermediate node IN as follows.

Step A1 $N \rightarrow IN$: $\langle tid_N, y_N, a_N, b_N, t_N \rangle$. N performs as follows.

- Picks r_N.
- Generates a timestamp t_N.
- Computes $x_N = a_N \oplus id_N$.
- Computes $y_N = x_N \oplus r_N$.
- Computes the temporary identity $tid_N = h(id_N \oplus t_N, r_N)$.

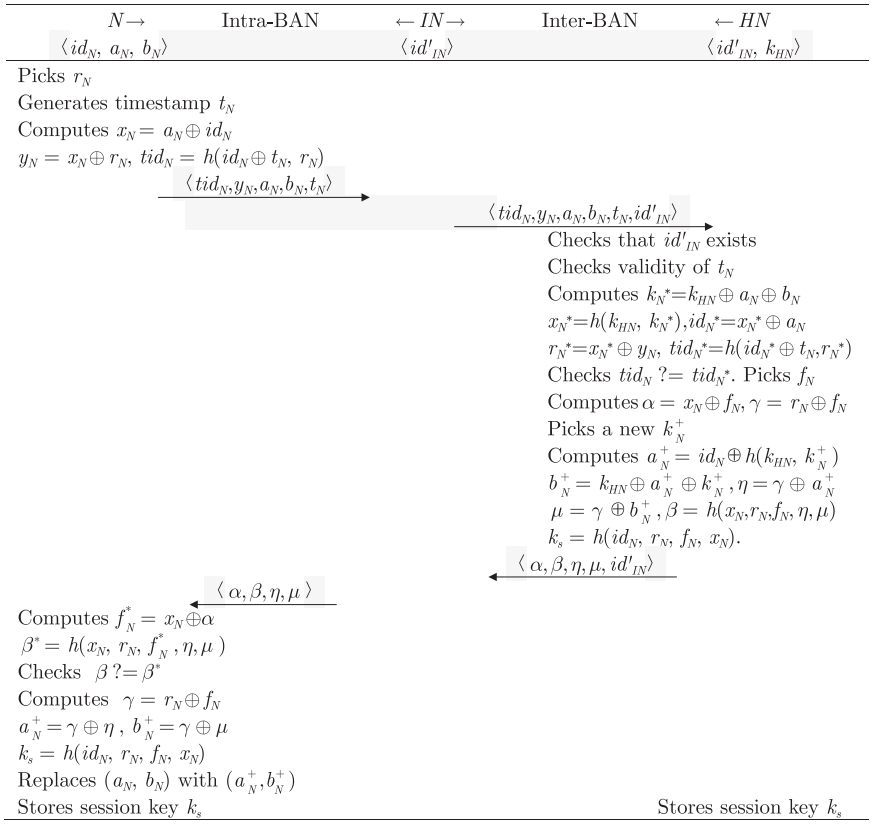

Fig. 3. The authentication and key agreement phase of our scheme for a two-hop centralized WBAN.

Step A2 $IN \rightarrow HN$: $\langle tid_N, y_N, a_N, b_N, t_N, id'_{IN}\rangle$.
The intermediate node *IN* simply forwards what it receives from the node *N* to *HN*. It only places her identity id'_{IN} (we emphasize that *IN* uses id'_{IN} not id_{IN}) to be identified by *HN*. *IN* uses its real identity since it is not incorporated in the session.

Step A3 $HN \rightarrow IN$: $\langle \alpha, \beta, \eta, \mu, id'_{IN}\rangle$. On the reception of the tuple $\langle tid_N, y_N, a_N, b_N, id'_{IN}\rangle$, *HN* performs as follows.

- Checks that id'_{IN} is in its database. Aborts if the check fails.
- Checks the validity of the timestamp t_N, by checking the validity of the predicate $(t^* - t_N \stackrel{?}{<} \Delta t)$, where t^* is the time when the message is received and Δt is the maximum transmission delay. Aborts if the predicate is not justified.
- Computes $k_N^* = k_{HN} \oplus a_N \oplus b_N$, $x_N^* = h(k_{HN}, k_N^*)$, $id_N^* = x_N^* \oplus a_N$ and $r_N^* = x_N^* \oplus y_N$.
- Computes $tid_N^* = h(id_N^* \oplus t_N, r_N^*)$.
- Checks $tid_N \stackrel{?}{=} tid_N^*$. Aborts if the check fails.
- Picks f_N and computes $\alpha = x_N \oplus f_N$ and $\gamma = r_N \oplus f_N$.
- Picks a new k_N^+.
- Computes a new $a_N^+ = id_N \oplus h(k_{HN}, k_N^+)$.
- Computes a new $b_N^+ = k_{HN} \oplus a_N^+ \oplus k_N^+$.
- Computes $\eta = \gamma \oplus a_N^+$ and $\mu = \gamma \oplus b_N^+$.
- Computes $\beta = h(x_N, r_N, f_N, \eta, \mu)$.
- Computes and stores the session key $k_s = h(id_N, r_N, f_N, x_N)$.

Step A4 $IN \rightarrow N$: $\langle \alpha, \beta, \eta, \mu\rangle$.
IN simply forwards to *N* what she received from *HN*. It just drops her identity id'_{IN}. Its identity is useless to *N* since *N* does not know *IN*.

Step A5 *N*: On the reception of the tuple $\langle tid_N, \alpha, \beta, \eta, \mu\rangle$, *N* performs as follows.

- Computes $f_N^* = x_N \oplus \alpha$.
- Computes $\beta^* = h(x_N, r_N, f_N^*, \eta, \mu)$.
- Checks $\beta \stackrel{?}{=} \beta^*$. Aborts if the check fails.
- Computes $\gamma = r_N \oplus f_N$.
- Computes $a_N^+ = \gamma \oplus \eta$.
- Computes $b_N^+ = \gamma \oplus \mu$.
- Computes and stores the session key $k_s^* (= k_s) = h(id_N, r_N, f_N, x_N)$.
- Replaces the parameters (a_N, b_N) with the parameters (a_N^+, b_N^+) in its memory.

4. Security analysis and evaluation of our scheme

In this section, the security of our scheme is analyzed by using Burrows–Abadi–Needham (BAN) logic [33] and also evaluated by the AVISPA tool [34]. Besides, the informal discussion with well-known attacks of our scheme is also given in this section.

4.1. Formal proof based on BAN logic

In the following, using the formal security analysis with the help of the widely-accepted BAN logic, we prove that our scheme provides secure mutual authentication between a sensor node *N* and the hub node *HN*.

We first give a brief introduction about the important symbols and rules of BAN logic, then we proceed the formal proof. Let *P* and *S* be participators and let *X* and *Y* denote a parameter, a formula or an expression.

- $P| \equiv X$: P believes the statement X.
- $\#(X)$: X is fresh.
- $P| \Rightarrow X$: P has jurisdiction over the statement X.
- $P \triangleleft X$: P sees the statement X.
- $P| \sim X$: P once said the statement X.
- (X, Y): X or Y is one part of the formula (X, Y).
- $\langle X \rangle_Y$: X combined with Y.
- $P \overset{K}{\longleftrightarrow} S$: K is a secret parameter shared (or to be shared) between P and S.
- $P \overset{X}{=} S$: X is secret known only to P and S, and possibly to parities trusted by them.

In addition, we use the following commonly used BAN logic rules to prove that our protocol provides secure mutual authentication and key agreement:

- Message-meaning rule: $\dfrac{P \mid\equiv P \overset{Y}{\longleftrightarrow} S, P \triangleleft \langle X \rangle_Y}{P \mid\equiv S \mid\sim X}$.
- Nonce-verification rule : $\dfrac{P \mid\equiv \#(X), P \mid\equiv S \mid\sim X}{P \mid\equiv S \mid\equiv X}$.
- Jurisdiction rule: $\dfrac{P \mid\equiv S \mid\Rightarrow X, P \mid\equiv S \mid\equiv X}{P \mid\equiv X}$.
- Freshness-conjuncatenation rule: $\dfrac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)}$.
- Belief rule: $\dfrac{P \mid\equiv (X, Y)}{P \mid\equiv X}$

In the following, we use the above BAN logic rules to prove mutual authentication and key agreement of our scheme.

Goals. The main goals of the analysis are as follows.

G1 $HN \mid\equiv N \mid\equiv (N \overset{x_N}{\longleftrightarrow} HN)$.
G2 $HN \mid\equiv (N \overset{x_N}{\longleftrightarrow} HN)$.
G3 $N \mid\equiv HN \mid\equiv (N \overset{k_s}{\longleftrightarrow} HN)$.
G4 $N \mid\equiv (N \overset{k_s}{\longleftrightarrow} HN)$.

Idealization. The idealized form of the messages transferred in the authentication phase between a node N and the hub node HN are given as follows.

M1. $N \rightarrow HN : \langle N \overset{x_N}{\longleftrightarrow} HN, r_N, t_N \rangle_{N \overset{id_N}{\longleftrightarrow} HN}$.
M2. $HN \rightarrow N : \langle N \overset{x_N}{\longleftrightarrow} HN, f_N, r_N, N \overset{k_s}{\longleftrightarrow} HN \rangle_{N \overset{id_N}{\longleftrightarrow} HN}$.

Assumptions: The initial assumptions of our protocol are listed below:

A1 $HN \mid\equiv (N \overset{id_N}{\longleftrightarrow} HN)$.
A2 $HN \mid\equiv \#(t_N)$.
A3 $HN \mid\equiv N \mid\Rightarrow (N \overset{x_N}{\longleftrightarrow} HN)$.
A4 $N \mid\equiv (N \overset{id_N}{\longleftrightarrow} HN)$.
A5 $N \mid\equiv \#(r_N)$.
A6 $N \mid\equiv HN \mid\Rightarrow (N \overset{k_s}{\longleftrightarrow} HN)$.

Analysis: We are ready to prove mutual authentication of our scheme as follows.

D1 From message **M1**, assumption **A1** and by applying the message meaning rule, we deduce,
$$\frac{HN \mid\equiv (N \overset{id_N}{\longleftrightarrow} HN), HN \triangleleft \langle N \overset{x_N}{\longleftrightarrow} HN, r_N, t_N \rangle_{N \overset{id_N}{\longleftrightarrow} HN}}{HN \mid\equiv N \mid\sim (N \overset{x_N}{\longleftrightarrow} HN, r_N, t_N)}$$
D2 From assumption **A2** and by applying the freshness rule, we deduce,
$$\frac{HN \mid\equiv \#(t_N)}{HN \mid\equiv \#(N \overset{x_N}{\longleftrightarrow} HN, r_N, t_N)}.$$
D3 From deductions **D1** and **D2**, by applying the nonce verification rule, we deduce,
$$\frac{HN \mid\equiv \#(N \overset{x_N}{\longleftrightarrow} HN, r_N, t_N), HN \mid\equiv N \mid\sim (N \overset{x_N}{\longleftrightarrow} HN, r_N, t_N)}{HN \mid\equiv N \mid\equiv (N \overset{x_N}{\longleftrightarrow} HN, r_N, t_N)}$$
D4 From deduction **D3** and by applying the belief rule, we deduce,
$$\frac{HN \mid\equiv N \mid\equiv (N \overset{x_N}{\longleftrightarrow} HN, r_N, t_N)}{HN \mid\equiv N \mid\equiv (N \overset{x_N}{\longleftrightarrow} HN)} \text{ (Goal } \mathbf{G1})$$
D5 From assumption **A3** and deduction **D4**, by applying the jurisdiction rule, we deduce,
$$\frac{HN \mid\equiv N \mid\Rightarrow (N \overset{x_N}{\longleftrightarrow} HN), HN \mid\equiv N \mid\equiv (N \overset{x_N}{\longleftrightarrow} HN)}{HN \mid\equiv (N \overset{x_N}{\longleftrightarrow} HN)} \text{ (Goal } \mathbf{G2})$$
D6 From message **M2**, assumption **A4** and by applying the message meaning rule, we deduce,
$$\frac{N \mid\equiv (N \overset{id_N}{\longleftrightarrow} HN), N \triangleleft \langle x_N, f_N, r_N, N \overset{k_s}{\longleftrightarrow} HN \rangle_{N \overset{id_N}{\longleftrightarrow} HN}}{N \mid\equiv HN \mid\sim (x_N, f_N, r_N, N \overset{k_s}{\longleftrightarrow} HN)}$$
D7 From assumption **A5** and by applying the freshness rule, we deduce,
$$\frac{N \mid\equiv \#(r_N)}{N \mid\equiv \#(x_N, f_N, r_N, N \overset{k_s}{\longleftrightarrow} HN)}.$$
D8 From deductions **D6** and **D7**, by applying the nonce verification rule, we deduce,
$$\frac{N| \equiv \#(x_N, f_N, r_N, N \overset{k_s}{\longleftrightarrow} HN, N \overset{id_N}{\longleftrightarrow} HN), N| \equiv HN| \sim (x_N, f_N, r_N, N \overset{k_s}{\longleftrightarrow} HN)}{N| \equiv HN| \equiv (x_N, f_N, r_N, N \overset{k_s}{\longleftrightarrow} HN)}$$
D9 From deduction **D8** and by applying the belief rule, we deduce,
$$\frac{N \mid\equiv HN \mid\equiv (x_N, f_N, r_N, N \overset{k_s}{\longleftrightarrow} HN)}{N \mid\equiv HN \mid\equiv (N \overset{k_s}{\longleftrightarrow} HN)} \text{ (Goal } \mathbf{G3})$$
D10 From assumption **A6** and deduction **D9**, by applying the jurisdiction rule, we deduce,
$$\frac{N \mid\equiv HN \mid\Rightarrow (N \overset{k_s}{\longleftrightarrow} HN), N \mid\equiv HN \mid\equiv (N \overset{k_s}{\longleftrightarrow} HN)}{N \mid\equiv (N \overset{k_s}{\longleftrightarrow} HN)} \text{ (Goal } \mathbf{G4})$$

Therefore, our scheme achieves mutual authentication and key agreement between HN and N.

4.2. Simulation based on AVISPA tool

In this section we give the details and results of the full simulation of our scheme using the AVISPA tool.

4.2.1. Preliminaries

AVISPA (Automated Validation of Internet Security Protocols and Applications) [34] is a security-assessment tool for the analysis of Internet security-sensitive protocols and applications. This tool is believed to speed up the development of the next generation of network security protocols and therefore increases the public acceptance of advanced, distributed IT applications based on them. We use AVISPA to assess the security goals of our proposed scheme since it is a widely used security-assessment tool of researchers. The tool measures whether the security protocol is SAFE or UNSAFE according to specified goals and is supported by a High Level Protocol Specification Language abbreviated as HLPSL. AVISPA is also supported by four integrated back-ends and abstraction-based methods through the HLPSL. These four back-ends are described as follows; (i) The On-the-Fly Model-Checker (OFMC) which is responsible for several symbolic techniques to explore the state space in a demand-driven way, this method is relatively a fast check. (ii) The Constraint-Logic-based Attack Searcher (CL-AtSe), which provides a translation from any security protocol specification written as a transition relation in intermediate format (IF) into a set of constraints which are effectively used to find

Table 2
Mapping of scheme symbols to HLPSL syntax variables/functions .

Scheme Symbols/functions	HLPSL syntax variables/functions
id_N	IDN
tid_N	TIDN
a_N	AN
b_N	BN
a_N^+	ANnew
b_N^+	BNnew
y_N	YN
x_N	XN
r_N	RN
k_N	KN
k_N^+	KNnew
t_N	TN
k_{HN}	KHN
f_N	FN
α	Alpha
β	Beta
η	Eta
μ	MU
$h(\cdot)$	H(·)
$\oplus$	xor

whether there are attacks on the protocol. (iii) SAT based Model checker which outputs a propositional formulas and then feeds these outputs to a state-of-the-art SAT solver and any model found is translated back into an attack. (iv) The Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) which is responsible for approximates the intruder knowledge by using regular tree languages. The HLPSL specification is interpreted into a lower level language than HLPSL. The intermediate form uses the so called HLPSL2IF interpreter which is transparent to the user. Intermediate form (IF) is read directly by the back-ends to the AVISPA tool. AVISPA is a role-oriented language in which each participants play a role during the protocol simulation. Each role is independent of the other, and gets some initial information by parameters and communicates with the other roles via channels. The intruder is modeled after the Dolev-Yao [32] model with the possibility for the intruder to assume a legitimate role in a protocol run. The role system also describes the number of sessions, the number of principals and the roles. Based on the four back-ends, the OUTPUT FORMATs (OFs), which are the simulation reports, are generated and after successful execution, the reports describe the result whether the protocol is SAFE or UNSAFE or under what condition the output is obtained. Again, these are achieved according to the specified goals.

There is another module, so called CAS+, which one may think of it as an editor for describing the security protocol in the form of Alice-Bob exchanged messages. This tool reduces the burden of the complicated syntaxes of the HLPSL language. In this tool, the designer specifies the protocol users/agents, the protocol identifiers, the transmitted messages, the knowledge available to each agent, the protocol instances, the intruder knowledge and finally, the security goals. Then this tool transforms the CAS+ code into the HLSPL code. However, The CAS+ does not give full control over the simulator functionalities, therefore, we prefer to write our script for the simulation using HLPSL language.

4.2.2. Simulation details

Table 2 shows the symbols used in our scheme and the corresponding AVISPA HLPSL scripting variables/functions. We start by determining our security goals from the simulation. We want to ensure, (i) the secrecy of the parameters KHN, KN and IDN. (ii) HN and N authenticate each other. Next, we write the HLPSL code/script for our scheme.

We define five roles: (i) admin, which is played by the system administrator SA, (ii) node, which is played by the sensor node N, (iii) hub node, which is played by the hub node HN, (iv) session, which defines the session role and all its declarations, (v) environment, which instantiate all variables, functions, agents, sessions and define the simulation security goals. The secure channel between SA and N for the registration/deployment is modeled in the HLPSL by assuming a symmetric key (SK) shared between SA and N. This key is beyond the reach of an attacker.

In the rest of this section we briefly describe each role. The role admin played by SA is shown in Fig. 4. The SA initially knows all other entities/agents in the protocol (SA, HN and N), the secret key KHN, the secret identity IDN of the node N, the secret KN, the symmetric key SK, the hash function in use H(·) and the send/receive channels Snd/Rcv. The notation (dy) means that the channels follow the Dolev-Yao model. The keyword 'Played_by SA' means that the role admin is played by the agent SA. After the keyword 'local', all local variables used in this role are declared. The variable 'State' is declared as 'nat' to represent the local state index of this role and is initialized to zero. AN and BN are the local variables in this role, as these what will be computed by SA for N. After the keyword 'const', we declare secKHN, secIDN and secKN as "protocol_id" for measuring the secrecy of the parameters KHN, IDN and KN, respectively, at different states of the protocol execution. The constant "hubnode_node" is also declared as "protocol_id" for HN-N authentication measurement. At state 0, the SA receives a start message "Rcv(start)" as an initialization for execution. The SA, using KHN, KN and IDN, computes AN': = xor(IDN,H(KHN.KN)) and BN': = xor(xor(KHN,AN'),KN) as described in our scheme and then sends the message (IDN.AN'.BN') encrypted using the symmetric key SK. This message will appear as a received message in the node's role. This ends the role admin played by SA. Notice that the variable is marked as primed (') if it is locally computed by the agent. One may also think of the symbol (/\) as conjunction.

Next we write the code for the role node played by N. This code is shown in Fig. 5h and is briefly described as follows. Node N knows all agents SA, N and HN, the symmetric key SK for the secure channel and the hash function H(·). Notice that N still does not know her identity IDN as it will be received in a secure message from SA. All locally used variables in this role are declared. Then, at State 0, N receives the registration message (IDN.AN'.BN') on the Rcv channel from SA encrypted using the symmetric key SK. Since SK is known to N, the simulator assumes that N has know the contents of the message. Now, as N knows IDN, AN and BN, N is ready to start authenticating with HN. Recall that N creates and sends the tuple $\langle tid_N, y_N, a_N, b_N, t_N\rangle$. This is represented by, Snd(TIDN'.YN'.AN'.BN'.TN'). TN':=new() and RN':=new() mean that TN and RN are picked by N as fresh new strings/nonces. The computations for XN,YN and TIDN follow from the description of our scheme. The received message, Rcv(Alpha'.Beta'.Eta'.MU'), in State 1, is received by N from HN which is the tuple $\langle \alpha, \beta, \eta, \mu\rangle$. The computations performed by N in this state follow from the description of our scheme. The statement secret(KHN,secIDN,{SA,HN}) is a request for the simulator to check that KHN is private to SA and HN, i.e. it is not revealed to an attacker or to N. The same applies for secret(IDN,secIDN,{SA,N,HN}) and secret(KN,secIDN,{SA,HN}). The identifiers secIDN, secKHN and secKN are used to specify these goals in the goal section. Finally, the statement, request(N,HN,node_hubnode,Beta), marked by node_hubnode identifier is a request for the simulator to check that N authenticates HN on Beta (i.e. on the random string/nonce RN, since Beta contains RN). The simulator checks that Beta satisfies this requirement. This "request" works in complement with the "witness" keyword discussed shortly in the HN's role. The "request" keyword stipulates, N accepts value RN, relies on exis-

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role admin (SA,HN,N:agent,SK:symmetric_key,KHN,KN,IDN:text,
H:hash_func, Snd,Rcv:channel(dy))
played_by SA
def=
local State:nat,
AN,BN:text
const secKHN,secIDN,secKN,hubnode_node: protocol_id
init State:=0
transition
1. State =0 /\ Rcv(start) =|> State':=1 /\
AN':=xor(IDN,H(KHN.KN)) /\ BN':= xor(xor(KHN,AN'),KN) /\
Snd({IDN.AN'.BN'}_SK)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fig. 4. HLPSL code for role admin played by SA.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role node (SA,HN,N:agent, SK:symmetric_key, H:hash_func,
Snd,Rcv:channel(dy))
played_by N
def=
local State:nat,
IDN,KN,KHN,AN,BN,YN,TN,RN,FN,KNnew,ANnew,BNnew,Alpha,
Eta,Gamma,MU:text,  XN:message,TIDN:hash(text.text),
KS:hash(text.text.text.message),
Beta:hash(message.text.text.text.text)
const secKHN,secIDN,secKN,hubnode_node: protocol_id
init State:=0
transition
1. State =0 /\ Rcv({IDN'.AN'.BN'}_SK) =|> State':=1 /\
RN':=new() /\ TN':=new() /\ XN':=xor(AN',IDN') /\
YN':= xor(XN',RN') /\ TIDN':= H(xor(IDN',TN').RN') /\
secret(KN,secKN,{SA,HN}) /\ secret(KHN,secKHN,{SA,HN})/\
secret(IDN,secIDN,{SA,N,HN}) /\ Snd(TIDN'.YN'.AN'.BN'.TN')
2. State =1 /\ Rcv(Alpha'.Beta'.Eta'.MU') =|>
State' :=2 /\ FN':=xor(XN,Alpha') /\
Beta':=H(XN.RN.FN'.Eta'.MU') /\ request (N,HN,hubnode_node,Beta) /\
Gamma':=xor(RN.FN') /\
ANnew':=xor(Gamma', Eta') /\ BNnew':=xor(Gamma',MU') /\
KS':=H(IDN.RN.FN'.XN) /\  AN':=ANnew' /\ BN':=BNnew' /\
secret(IDN,secIDN,{SA,N,HN})/\ secret(KHN,secKHN,{SA,HN})/\
secret(KN,secKN,{SA,HN})
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fig. 5. HLPSL code for role node played by N.

tence of HN and HN's agreeing to value Beta, Beta should be fresh, not replayed and HN's existence bound to protocol_id hubnode_ node.

The code for role hubnode played by the hub node HN is written in a similar way, taking into consideration the necessary changes specific for this role, which can be seen in Fig. 6. Here, in addition to the other parameters initially know by HN, HN initially knows the master secret key KHN. Notice that HN does not know neither KN nor IDN, as these parameters will be locally computed once HN receives from N. Notice also that HN does not know the symmetric key SK as it does not communicate with SA over a secure channel. This role has only one state, where HN receives, Rcv(TIDN'.YN'.AN'.BN'.TN') which was sent by N in role node played by N. Once this message is received, HN performs the necessary computations and replies by Snd(Alpha'.Beta'.Eta'.MU') which was received by N in the role node. We also create a secrecy check request to check the secrecy of the parameters IDN, KHN and KN. The statement witness(HN,N,hubnode_node,Beta') works in complement with the statement request(N,HN,hubnode_node,Beta). This statement stipulates, agent HN wants to be a peer with agent N in the protocol run with protocol_ id hubnode_node and HN wants to agree with N on the value Beta, for the purpose of authentication.

We are not able to check that HN authenticates N, since in our scheme, this is done using a timestamp TN. The AVISPA tool does not support authentication based on timestamps. In order to simulate our protocol, we have bypassed this problem by allowing N to pick TN as if it is a random nonce. However, the concept of random nonce based authentication is different from timestamp based authentication and hence, checking the authentication of N to HN will fail. Due to this limitation of AVISPA, we only checked one direction, which is the authentication of HN to N, which is based on random nonce in our scheme.

After we finished the code for the three roles of the agents, we continue to write the session role. In this part of the HLPSL code which is shown in Fig. 7, the agents' roles are called and all the session parameters are declared. The initial (a priori/pre-known) constant parameters and their declarations are given first. Notice that, KHN, KN, IDN, SK and H are included as pre-known constants in the session. After keyword 'local', we declare the channels available to the agents. Each agent is assigned a send and a receive channel. In the composition part, the role is called for each agent with his constant parameters/functions. The admin role is called with SA, HN and N as agents, KHN, KN and IDN as pre-known constants, SK as a pre-known symmetric key, H as the used hash function, SSAch and RSAch as the send and receive channels for SA. In

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role hubnode (SA,HN,N:agent, KHN:text,H:hash_func,
Snd,Rcv:channel(dy))
played_by HN
def=
local State:nat,
IDN,KN,AN,BN,YN,TN,RN,FN,KNnew,ANnew,BNnew,Alpha,Eta,
Gamma,MU:text,TIDN:hash(text.text), XN:message,
Beta:hash(message.text.text.text.text),
KS:hash(text.text.text.message)
const secKHN,secIDN,secKN,hubnode_node: protocol_id
init State:=0
transition
1. State =0 /\ Rcv(TIDN'.YN'.AN'.BN'.TN')=|>
State':=1  /\ XN':=H(KHN.xor(xor(KHN,AN'),BN')) /\
IDN':=xor(XN',AN') /\ RN':=xor(XN',YN') /\
TIDN':=H(xor(IDN',TN').RN') /\ FN':=new()/\ Alpha':=xor(XN',FN') /\
Gamma':=xor(RN',FN') /\ KNnew':=new() /\
ANnew':=xor(IDN',H(KHN.KNnew')) /\
BNnew':=xor(xor(KHN,ANnew'), KNnew') /\
Eta':=xor(Gamma',ANnew') /\ MU':=xor(Gamma',BNnew') /\
Beta':=H(XN'.RN'.FN'.Eta'.MU') /\  KS':=H(IDN'.RN'.FN'.XN') /\
secret(KHN,secKHN,{SA,HN})/\ secret(KN,secKN,{SA,HN}) /\
secret(IDN,secIDN,{SA,N,HN}) /\
Snd(Alpha'.Beta'.Eta'.MU') /\ witness (HN,N,hubnode_node,Beta')
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fig. 6. HLPSL code for role hubnode played by HN.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role session (SA, HN,N:agent, SK:symmetric_key, KHN,KN,IDN:text,
H:hash_func)
def=
local SHNch,RHNch,SNch,RNch,SSAch,RSAch,T1,T2:channel(dy)
composition
admin (SA,HN,N, SK,KHN, KN,IDN,H, SSAch,RSAch) /\
hubnode(SA,HN,N,KHN,H,SHNch,RHNch) /\
node (SA,HN,N,SK,H,SNch,RNch)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fig. 7. HLPSL code for role session.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role environment()
def=
const sa,hn,n:agent, sk:symmetric_key,khn,kn,idn:text,h:hash_func,
seckhn,secidn,seckn,hubnode_node:protocol_id
intruder_knowledge ={sa,hn,n,h}
composition
session(sa,hn,n,sk,khn,kn,idn,h)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
goal
secrecy_of  secKHN
secrecy_of secIDN
secrecy_of secKN
authentication_on hubnode_node
end goal
environment()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Fig. 8. HLPSL code for role environment.

a similar way the roles, hubnode and node, are called. This ends the session role.

Next, the environment role is given in Fig. 8. This is where one or more sessions are instantiated. First, all constants are instantiated and declared. The instants, sa, n and hn are for SA, N and HN respectively, which instantiate the agents, sk instantiates the symmetric key SK, h instantiates the hash function H, and khn, kn and idn instantiate KHN, KN and IDN respectively. The protocol identifiers, secKHN, secIDN, secKN and hubnode_node are also declared. In the intruder knowledge part, all instants that the attacker is assumed to know prior to execution are given. We assume that the attacker knows sa, n and hn, he also knows the hash h. Finally, in the role environment, in the composition part, the session is instantiated with the instances (sa,hn,n,sk,h,khn,kn,idn). Finally, after all the roles are written, the simulation goals are defined under the 'goal' keyword using the protocol identifiers declared as 'protocol_id', as shown in Fig. 8. The simulator is told to check the secrecy of KHN at the different states using 'secrecy_of

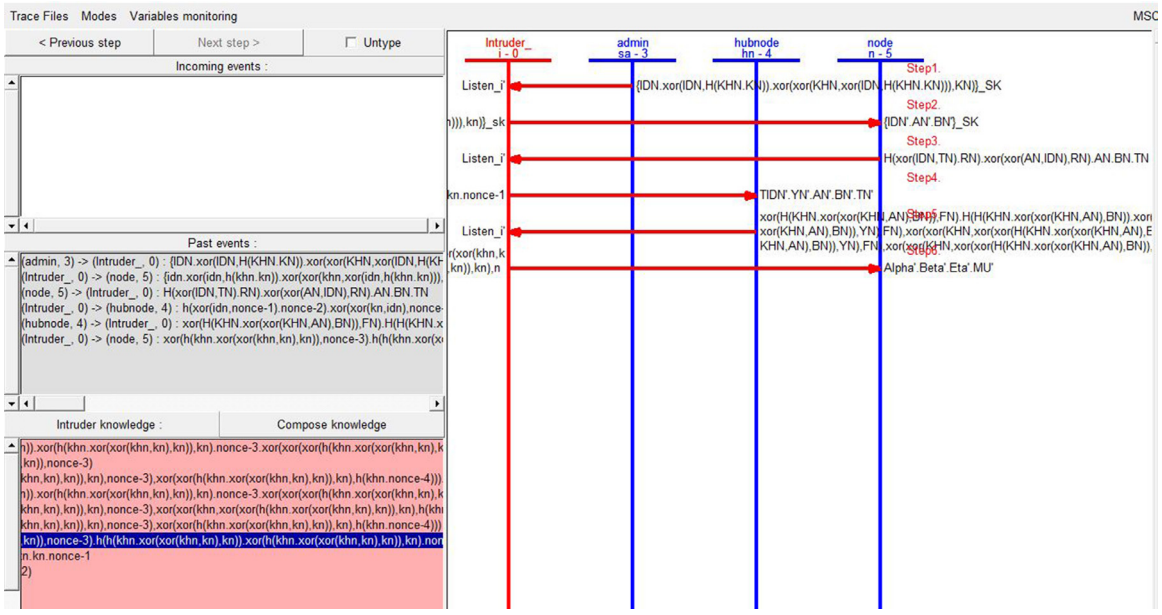

Fig. 9. Snapshot of the protocol full execution using SPAN animator.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\AVISPA\suite\results\wban.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 136.38s
  visitedNodes: 62 nodes
  depth: 2 plies
```

(a) OFMC summary report

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  C:\AVISPA\suite\results\wbanauth.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 1 states
  Reachable  : 1 states
  Translation: 0.03 seconds
  Computation: 0.01 seconds
```

(b) CL-AtSe summary report

Fig. 10. Back-end checkers summary reports.

secKHN', the secrecy of IDN using 'secrecy_of secIDN' and the secrecy of KN using 'secrecy_of secKN'. The authentication is checked using 'authentication_on hubnode_node'. Again, the authentication in the other way could not be checked since AVISPA does not support timestamp authentication as we stated earlier. This ends the role of the environment.

4.2.3. Simulation results

In this section, we specify the simulation results of our protocol. We execute the OFMC back-end checker and the CL-AtSe back-end checker. Fig. 9 is a snapshot of the protocol execution animator using SPAN software [35], showing the full execution of the protocol in the presence of the intruder. The report for the OFMC back-end checker is shown in Fig. 10(a), which shows that our scheme is SAFE and hence satisfies all the specified security goals. The report for the CL-AtSe back-end checker is shown in Fig 10(b), which also shows that our scheme is SAFE under this checker model and satisfies all the specified security goals. Finally, we want to mention that, we were not able to execute the TA4SP back-end checker as this checker does not support XOR operation.

4.3. Discussions of other attacks and services

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

4.3.1. Eavesdropping attack

In the authentication phase of our scheme, an adversary $\mathcal{A}$ is able to record all transmitted parameters between N and HN.

She collects the tuple, $\langle tid_N, y_N, a_N, b_N\rangle$ from N to HN and the tuple, $\langle \alpha, \beta, \eta, \mu\rangle$ from HN to N. Notice that the session key $k_s = h(id_N, r_N, f_N, x_N)$. From the intercepted parameters, $\mathcal{A}$ cannot reach id_N from tid_N as it is protected by the one-wayness of $h(.)$. The parameter $y_N = a_N \oplus id_N \oplus r_N$ does not allow $\mathcal{A}$ to compute r_N as she does not know id_N. Also $\mathcal{A}$ does not know f_N picked at random by HN. The parameter $\alpha = r_N \oplus f_N$ does not allow $\mathcal{A}$ to reach neither r_N nor f_N as they are both unknown to $\mathcal{A}$. The parameter $\beta = h(x_N, r_N, f_N, \eta, \mu)$ does not allow $\mathcal{A}$ to reach any of the unknown inputs from the properties of $h(\cdot)$. Finally, $\mathcal{A}$ may compute $\eta \oplus \mu = a_N^+ \oplus b_N^+$, but since both a_N^+ and b_N^+ are unknown to $\mathcal{A}$, she cannot compute any of them. Therefore the privacy of the session key k_s is preserved and hence, the scheme protects against an eavesdropping attack.

4.3.2. Anonymous and unlinkable sessions

The goals of this service is that, from the intercepted communication parameters, an adversary $\mathcal{A}$ cannot obtain the real identity id_N of any communicating node N and cannot link any session to any other session of the same node N. Consider the tuple, $\langle tid_N, y_N, a_N, b_N\rangle$ from N to HN. We have $tid_N = h(id_N \oplus t_N, r_N)$, and since r_N is chosen independent, random and fresh by N each session, then $\mathcal{A}$ cannot link any two different tid_N's to the same N. The parameter $y_N = x_N \oplus r_N$ is also fresh each session. The two parameters a_N and b_N are renewed for N by HN every conducted session using an independent, random and fresh k_N picked locally by HN. Now consider the tuple $\langle \alpha, \beta, \eta, \mu\rangle$, the values α, β, η and μ are all functions of an independent, fresh and random parameter f_N picked locally by HN. Our design ensures that these randomly picked parameters cannot be figured out by $\mathcal{A}$ in order to obtain some fixed parameter. Hence, the communication parameters are random, fresh and independent every conducted session. An adversary $\mathcal{A}$ is unable to link two or more sessions to the same node N, therefore, our scheme preserves anonymity of the nodes and unlinkability of the established sessions.

4.3.3. Sensor node impersonattion attack

A successful impersonation attack means that $\mathcal{A}$ has the capability to create a valid tuple $\langle tid_A, y_A, a_A, b_A, t_A\rangle$. Here we have two situations: (i) $\mathcal{A}$ captures a transmitted tuple $\langle tid_A, y_A, a_A, b_A, t_A\rangle$ from eavesdropping on the channel but she does not know the corresponding id_A. (ii) $\mathcal{A}$ knows id_A and the pair (a_A, b_A). The second situation means that the adversary has fully captured the node and hence we are facing a different attack. In the first situation, the captured tuple will not allow $\mathcal{A}$ to impersonate a node without the corresponding id_A since $tid_A = h(id_A \oplus t_A, r_A)$ is tied to the timestamp, and the adversary is unable to create a valid temporary identity without the corresponding id_A.

4.3.4. Replay attack

Protection against replay attack of the sensor node message in our scheme is achieved using timestamps. The timestamp t_N generated by node N is inserted in a way that ensures it cannot be wiped out or replaced by an adversary $\mathcal{A}$. A replay of HN's message will fail without knowing the correct r_N expected by N, since r_N plays the role of a random nonce as will be verified by the simulation.

4.3.5. Hub node spoofing attack

In order for $\mathcal{A}$ to masquerade as the hub node HN to a node N, she must generate a valid tuple $\langle \alpha, \beta, \eta, \mu\rangle$. The parameter $\alpha = x_N \oplus f_N$, where $x_N = h(k_{HN}, k_N)$. A valid x_N is a function of two parameters k_{HN} and k_N totally unknown to $\mathcal{A}$. Also β, η and μ are all function of x_N. An adversary $\mathcal{A}$ is not able to produce any valid parameter of these parameters, therefore, our scheme protects against HN spoofing attack.

4.3.6. Sensor node capture attack

An adversary $\mathcal{A}$ is able to capture as many sensor nodes as she can, and knows the tuples $\langle id_N, a_N, b_N\rangle$. We need to ensure that the revealed tuples do not reveal any information about the HN's master secret key k_{HN}. $\mathcal{A}$ now knows both a_N and b_N. Recall that $a_N = id_N \oplus h(k_{HN}, k_N)$ and $b_N = k_{HN} \oplus a_N \oplus k_N$. $\mathcal{A}$ can obtain the quantity $h(k_{HN}, k_N) = a_N \oplus id_N$ where both k_{HN} and k_N are protected by the one-wayness of $h(.)$. From the knowledge of b_N and a_N, the equation $b_N = k_{HN} \oplus a_N \oplus k_N$ does not reveal any information about neither k_{HN} nor k_N, since all values of k_{HN} and k_N are possible. Therefore, knowing the pairs, $\{(h(k_{HN}, k_N^{(1)}), k_{HN} \oplus k_N^{(1)}), \cdots, (h(k_{HN}, k_N^{(m)}), k_{HN} \oplus k_N^{(m)})\}$ does not allow $\mathcal{A}$ to extract any $k_N^{(i)}$ as they all different. We conclude that, the only way for $\mathcal{A}$ is brute force on both k_{HN} and k_N. Hence, no other sensor node is affected by the capture of any number of sensor nodes, therefore, our scheme still performs securely.

4.3.7. Forward/backward security

The goal of this service is that, if any session key k_s is revealed, this must not affect the privacy of any past or future sessions. It is easily observed that our scheme achieves this service. Recall that the session key is computed as $k_s = h(id_N, r_N, f_N, x_N)$. All sensitive parameters are protected with the one-wayness of the hash function h, and r_N, f_N and x_N are all dynamic change with the sessions. Therefore, our scheme achieves forward/backward security.

4.3.8. Hub node stolen database attack

One of the merits of our scheme is that, except the master secret key k_{HN}, HN does not need to store any secret information in its database. Only the real identities (id'_{lN})s of the registered first level nodes are stored which are not secret. These real identities are required only to verify the identity of the intermediate node. Therefore, our scheme does not need to worry about this attack.

4.3.9. Secure against ephemeral secret key leakage

The session key k_s is computed as $k_s = h(id_N, r_N, f_N, x_N)$. This key is not used as an input to any function during the authentication phase. It is computed at the end of the protocol. The leakage of k_s does not threaten the security of the master secret key k_{HN}.

4.3.10. Man-in-the-middle attack

Protection against man-in-the-middle attack follows from the protection against sensor node impersonation attack, hub node spoofing attack and replay attacks.

4.3.11. Jamming/desynchronization attacks

An authentication scheme is vulnerable to a desynchronization attack if it requires the two parties to update their state in synchronism. In this case, after one party updates its state, the attacker corrupts the link leaving the other party unable to update its state. All authentication protocols that requires the server to store verification tables are vulnerable to such attack. In our scheme, the hub node is not required to store any verification tables. In other words, the hub node does not store any state for the sensor nodes. If an authentication attempt fails due to link corruption by jammers, preventing the sensor node from computing the new parameters (a_N^+, b_N^+), the sensor is still able to start a new authentication attempt using the already stored parameters (a_N, b_N) without the need to reinitialize.

5. Efficiency evaluation and comparisons

In this section we concretely evaluate the efficiency of our scheme. We evaluate the storage requirements, the computation cost, the energy consumption and the communication overheads.

Table 3
Storage and computation cost of our scheme .

Node	Storage cost (in bits)	Computation cost
N	640	$3t_h + 6t_{xor} \approx 3t_h$
HN	$16m + 160$	$5t_{hash} + 11t_{xor} \approx 5t_h$

Table 4
Computation time and energy consumption of our scheme on 32-bit Cortex-M3 micro-controller at 72 MHz .

Node	Computation time (ms)	Energy consumption (mJ)
N	0.18	0.021
HN	0.3	0.035

5.1. Storage requirements

In our scheme, the hub node is required to store its own master secret key k_{HN} as well as id'_N of the registered first level sensor nodes. On the other hand, each second level sensor node is required to store the tuple $\langle id_N, a_N, b_N\rangle$, in addition to the session key k_s. For a first level node, it is required to store also id'_N which is assumed short (16 bits). We use SHA-1 as an example of hash function, and the output of SHA-1 is 160 bits. By applying these settings, then $|id_N| = |a_N| = |b_N| = |k_s| = |k_{HN}| = 160$ bits, while $|id'_N| = 16$ bits. The total storage required by HN is $(16m + 160)$ bits, where m is the number of registered first level sensor nodes. Each second level node N is required to store 640 bits. The storage requirements are summarized in Table 3

5.2. Computation cost

Our scheme uses two operations, i.e. hash function and XOR operation. Let t_h and t_{xor} to be the computation time of one hash invocation and one XOR operation, respectively. Considering the authentication phase of Fig. 3. The hub node HN performs 5 hash invocations and 11 XOR operations. These total $5t_h + 11t_{xor}$. On the other hand, the sensor node N performs 3 hash invocations and 6 XOR operations. These total $3t_h + 5t_{xor}$.

The computation time of XOR operation is very trivial and can be ignored assuming $t_{xor} \approx 0$. Therefore, the computation cost required by HN becomes $5t_h + 11t_{xor} \approx 5t_h$. The computation cost of the sensor node N becomes $3t_h + 6t_{xor} \approx 3t_h$. These results are summarized in Table 3.

5.3. Computation time and energy consumption

On a 32-bit Cortex-M3 micro-controller running at 72 MHz, a SHA-1 hash invocation takes 0.06 ms [36,37]. It follows from Table 3 that a sensor node N takes 0.18 ms while the hub node HN takes 0.3 ms.

The same micro-controller, in ambient/room temperature (at 300^o K) in active mode consumes 36 mA under 3.3 V [36]. Therefore, the power consumed in active mode is 118.8 mW. This power consumption can be used to give a rough estimate of the energy consumed during computations. The sensor node N takes about 0.18 ms, therefore the energy consumed by N is $0.198 * 118.8/1000 = 0.024$ mJ. On the other hand, the hub node, HN consumes $0.3 * 118.8/1000 = 0.042$ mJ. These are summarized in Table 4.

5.4. Communication overheads

The communication overheads of our scheme is shown in Table 5. In the transmission ($N \rightarrow IN$), N sends the tuple, $\langle tid_N, y_N, a_N, b_N, t_N\rangle$. Assume $|t_N| = 32$ bits, therefore, the size of this tuple is $4(160) + 32 = 672$ bits. In the transmission ($IN \rightarrow HN$), IN sends the tuple, $\langle tid_N, y_N, a_N, b_N, t_N, id'_{IN}\rangle$ of size 688 bits. In the transmission ($HN \rightarrow IN$), HN sends the tuple, $\langle \alpha, \beta, \eta, \mu, id'_{IN}\rangle$ of size 656 bits. Finally, in the transmission ($IN \rightarrow N$), IN sends the tuple, $\langle \alpha, \beta, \eta, \mu\rangle$ of size 640 bits.

Table 5
Communication cost of our scheme .

Communication between nodes	Communication cost
$N \rightarrow IN$	672 bits
$IN \rightarrow HN$	688 bits
$HN \rightarrow IN$	656 bits
$IN \rightarrow N$	640 bits

Table 6
Computation cost of cryptographic operations with t_h as the time unit .

Symbol	Description	Cost
t_h	One invocation of SHA-1 hash	t_h
t_{sym}	Symmetric encryption	t_h
t_{ecsm}	Scalar multiplication on ECC	$72.5t_h$
t_{ecpa}	Point addition on ECC	$13t_h$
t_{mm}	Modular multiplication	$2.5t_h$
t_{ma}	Modular addition	$0.3t_h$
t_{exp}	Modular exponentiation	$600t_h$
t_{map}	Map-to-point on ECC	$450t_h$

5.5. Simplicity of key management

In our scheme, the only secret information stored at the hub node, is its master secret key k_{HN}. The hub node is not required to store any other secret parameters for the sensors. This makes the management of the keys on the hub node very simple compared to other schemes.

5.6. Comparisons with recent schemes

To facilitate the comparison with previous schemes, we recall the experimental results obtained in [38,39]. From these experimental results, we construct Table 6 that shows the computation time of different cryptographic operations, which are mapped to the hashing time t_h as the time unit. We use Table 6 to provide a real comparison of our proposed scheme with previously proposed schemes.

In the authentication protocol [27], the AAL server executes one hash invocation, three symmetric encryption operations, and one elliptic curve point multiplication operation, which is $1t_h + 4t_{sym} + 1t_{ecsm} = 77.5t_h$. The hub in this authentication protocol executes one hash function operation, two symmetric encryption/decryption operations, and two elliptic cure point multiplication operations. Thus, the execution time of the hub is $1t_h + 2t_{sym} + 2t_{ecsm} = 148t_h$. The user/node performs $2t_h + 2t_{sym} + 3t_{ecsm} = 221.5t_h$. In the protocol [25], the user performs $4t_h + 1t_{sym} + 3t_{ecsm} = 222.5t_h$. On the other hand, the hub performs $5t_h + 1t_{sym} + 6t_{ecsm} = 441t_h$. In the protocol [24], the user performs $3t_h + 1t_{sym} + 4t_{ecsm} + 1t_{exp} = 894t_h$ and the hub node performs $6t_h + 2t_{sym} + 4t_{ecsm} + 1t_{exp} + 1t_{pair} = 2534t_h$. The recent scheme of He et al. [29], is not lightweight since it requires the user to perform complex map-to-point operations and the AP to perform elliptic curve pairings. The user is required to perform four elliptic curve scalar multiplications, one map-to-point operation, one point addition and four hashes. The user's computation cost is $4t_{ecsm} + 1t_{map} + 1t_{ecpa} + 4t_h \approx 757t_h$. In the recent scheme of Liu et al. [30], the user is required to perform two hash invocations, two elliptic curve point additions, two elliptic curve scalar multiplications, three modular multiplications and one modular addition. The user's computation cost is

Table 7
Comparisons of the computation costs and energy consumption of the hub node among existing protocols and our protocol .

Protocol	Computation cost	Computation time	Energy
Liu et al. 2014 [24]	$6t_h + 2t_{sym} + 4t_{ecsm} + 1t_{exp} + 1t_{pair} = 2534t_h$	152.04 ms	18.06 mJ
He-Zeadally 2016 [29]	$t_{ecsm} + 1t_{map} + 1t_{ecpa} + 4t_h = 757t_h$	45.42 ms	5.4 mJ
Zhao 2014 [25]	$5t_h + 1t_{sym} + 6t_{ecsm} = 441t_h$	26.46 ms	3.14 mJ
Liu et al. 2016 [30]	$2t_h + 2t_{ecpa} + 2t_{ecsm} + 1t_{map} + 3t_{mm} + 1t_{ma} = 180.8t_h$	10.848 ms	1.29 mJ
He-Zeadally 2015 [27]	$1t_h + 2t_{sym} + 2t_{ecsm} = 148t_h$	8.88 ms	1.06 mJ
Proposed scheme	$5t_h + 11t_{xor} = 5t_h$	0.3 ms	0.035 mJ

$2t_h + 2t_{ecpa} + 2t_{ecsm} + 1t_{map} + 3t_{mm} + 1t_{ma}$, the user's computation cost is $180.8t_h$.

Table 7 summarizes the computation time and energy consumption required by the hub node in our protocol and other previous existing related recently proposed protocols during the authentication and key agreement phase. From this table, it is clear that our protocol is extremely more efficient compared to other protocols.

6. Conclusions

In this paper, we proposed a lightweight authentication scheme for two-hop centralized WBAN, and it provides anonymous and unlinkable features for wearable sensors while achieving the mutual authentication between wearable sensors and hub node. Our scheme just need to execute hash operations and XOR operations, and it is more efficient than previously related schemes. Specifically, the sensor node and the hub node are just need to perform three and five hash operations, respectively. Due to the low computational cost, the energy consumption of the wearable sensors is very low and hence, and the running time of WBAN can be guaranteed. As to storage aspects, the hub node does not need to store any verification tables or any secret information in its database. The security of the proposed scheme is formally proved by using BAN logic, and our scheme achieves its security goals. Besides, we evaluated the security of our scheme by simulation using the Automated Validation of Internet Security Protocols and Applications (AVISPA), and the simulation results show that our scheme is safe. Furthermore, informal security analysis of our scheme is given, and our scheme withstands the best known attacks. Except the security analysis, we compared our scheme with other proposed schemes and showed that our scheme outperforms recently proposed schemes.

Although there are some security solutions for WBAN, either of these schemes are not designed for wearable devices of WBAN or they are not lightweight and efficient for wearable sensors. Besides, most of these schemes did not consider the anonymous and unlinkability feature of the wearable deivces. We can say that the research of security mechanisms of WABN has just started, and there are many challenges to designing practical security solutions for WBAN. How to balance the conflict among security, efficiency, and practicality is an eternal issue in the design of security schemes for WBAN. Privacy protection would be another important issue in WBAN security. On the one hand, the collected data of WBAN contains important privacy of patients. On the other hand, these data may be easily obtained by an adversary due to the fragility of the open wireless channels. In the future, we will continue to explore these security problems of WBAN, and design wearable sensors applicable security solutions.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61300220 & 61572013 & 61572188, the Scientific Research Fund of Hunan Provincial Education Department under Grant No. 16B089. Saru Kumari is sponsered by the University Grants Commission, India through UGC-BSR Start-up grant under Grant no. 3(A)(60)31.

References

[1] B. Latré, B. Braem, I. Moerman, C. Blondia, P. Demeester, A survey on wireless body area networks, Wireless Networks 17 (2011) 1–18.
[2] K. Akkaya, M. Younis, M. Youssef, Efficient aggregation of delay-constrained data in wireless sensor networks, in: Computer Systems and Applications, 2005. The 3rd ACS/IEEE International Conference on, IEEE, 2005, pp. 904–909.
[3] U. Varshney, Pervasive healthcare: applications, challenges and wireless solutions, Commun. Assoc. Inf. Syst. 16 (2005) 3.
[4] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, K. Ren, A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing, IEEE Trans. Inf. Forensics Secur. 11 (2016) 2594–2608.
[5] Z. Xia, X. Wang, X. Sun, Q. Wang, A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data, IEEE Trans. Parallel Distrib. Syst. 27 (2016) 340–352.
[6] Z. Fu, X. Sun, Q. Liu, L. Zhou, J. Shu, Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing, IEICE Trans. Commun. 98 (2015) 190–200.
[7] Y. Kong, M. Zhang, D. Ye, A belief propagation-based method for task allocation in open and dynamic cloud environments, Knowl Based Syst 115 (2017) 123–132.
[8] P.K. Sahoo, Efficient security mechanisms for mhealth applications using wireless body sensor networks, Sensors 12 (2012) 12606–12633.
[9] X. Li, J.-W. Niu, J. Ma, W.-D. Wang, C.-L. Liu, Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards, J. Network Comput. Appl. 34 (2011) 73–79.
[10] X. Li, J. Niu, M.K. Khan, J. Liao, An enhanced smart card based remote user password authentication scheme, J. Network Comput. Appl. 36 (2013) 1365–1371.
[11] Q. Jiang, M.K. Khan, X. Lu, J. Ma, D. He, A privacy preserving three-factor authentication protocol for e-health clouds, J. Supercomput. 72 (2016) 3826–3849.
[12] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, A. Alelaiwi, Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy, Nonlinear Dyn. 83 (2016) 2085–2101.
[13] X. Li, Y. Xiong, J. Ma, W. Wang, An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards, J. Network Comput. Appl. 35 (2012) 763–769.
[14] X. Li, J. Ma, W. Wang, Y. Xiong, J. Zhang, A novel smart card and dynamic id based remote user authentication scheme for multi-server environments, Math. Comput. Model 58 (2013) 85–95.
[15] D. Wang, P. Wang, On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions, Comput. Networks 73 (2014) 41–57.
[16] D. Wang, P. Wang, Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks, Ad Hoc Netw. 20 (2014) 1–15.
[17] K.K. Venkatasubramanian, A. Banerjee, S.K.S. Gupta, Pska: usable and secure key agreement scheme for body area networks, IEEE Trans. Inf. Technol. Biomed. 14 (2010) 60–68.
[18] Z. Zhang, H. Wang, A.V. Vasilakos, H. Fang, Ecg-cryptography and authentication in body area networks, IEEE Trans. Inf. Technol. Biomed. 16 (2012) 1070–1078.
[19] K. Zeng, K. Govindan, P. Mohapatra, Non-cryptographic authentication and identification in wireless networks, IEEE Wireless Commun. 17 (2010).
[20] L. Shi, J. Yuan, S. Yu, M. Li, Ask-ban: authenticated secret key extraction utilizing channel characteristics for body area networks, in: Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, ACM, 2013, pp. 155–166.
[21] A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, A. LaMarca, Ensemble: cooperative proximity-based authentication, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, ACM, 2010, pp. 331–344.
[22] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, N. Mandayam, Proximate: proximity-based secure pairing using ambient wireless signals, in: Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services, ACM, 2011, pp. 211–224.

[23] M. Li, S. Yu, J.D. Guttman, W. Lou, K. Ren, Secure ad hoc trust initialization and key management in wireless body area networks, ACM Trans. Sensor Networks (TOSN) 9 (2013) 18.
[24] J. Liu, Z. Zhang, X. Chen, K.S. Kwak, Certificateless remote anonymous authentication schemes for wirelessbody area networks, IEEE Trans. Parallel Distrib. Syst. 25 (2014) 332–342.
[25] Z. Zhao, An efficient anonymous authentication scheme for wireless body area networks using elliptic curve cryptosystem, J. Med. Syst. 38 (2014) 13.
[26] H. Xiong, Cost-effective scalable and anonymous certificateless remote authentication protocol, IEEE Trans. Inf. Forensics Secur. 9 (2014) 2327–2339.
[27] D. He, S. Zeadally, Authentication protocol for an ambient assisted living system, IEEE Commun. Mag. 53 (2015) 71–77.
[28] J. Shen, H. Tan, S. Moh, I. Chung, Q. Liu, X. Sun, Enhanced secure sensor association and key management in wireless body area networks, J. Commun. Networks 17 (2015) 453–462.
[29] D. He, S. Zeadally, N. Kumar, J.-H. Lee, Anonymous authentication for wireless body area networks with provable security, IEEE Syst. J. (2016).
[30] J. Liu, L. Zhang, R. Sun, 1-Raap: an efficient 1-round anonymous authentication protocol for wireless body area networks, Sensors 16 (2016) 728.
[31] H. Xiong, Z. Qin, Revocable and scalable certificateless remote authentication protocol with anonymity for wireless body area networks, IEEE Trans. Inf. Forensics Secur. 10 (2015) 1442–1455.
[32] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inf. Theory 29 (1983) 198–208.
[33] M. Burrows, M. Abadi, R.M. Needham, A logic of authentication, in: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 426, The Royal Society, 1989, pp. 233–271.
[34] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuéllar, P.H. Drielsma, P.-C. Héam, O. Kouchnarenko, J. Mantovani, et al., The avispa tool for the automated validation of internet security protocols and applications, in: International Conference on Computer Aided Verification, Springer, 2005, pp. 281–285.
[35] Y. Glouche, T. Genet, O. Heen, O. Courtay, A security protocol animator tool for avispa, ARTIST2 Workshop on Security Specification and Verification of Embedded Systems, Pisa, 2006.
[36] 2016, http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32f1-series/stm32f103/stm32f103ve.html.
[37] J. Liu, Q. Li, R. Yan, R. Sun, Efficient authenticated key exchange protocols for wireless body area networks, EURASIP J. Wirel. Commun. Netw. 2015 (2015) 188.
[38] J.-J. Huang, W.-S. Juang, C.-I. Fan, H.-T. Liaw, et al., Robust and privacy protection authentication in cloud computing, Int. J. Innov.Comput., Inf. Control 9 (2013) 4247–4261.
[39] X. Cao, W. Kou, X. Du, A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges, Inf. Sci. 180 (2010) 2895–2903.

Xiong Li now is an associate professor at School of Computer Science and Engineering of the Hunan University of Science and Technology (HNUST), China. He received his masters degree in mathematics and cryptography from Shaanxi Normal University (SNNU), China in 2009 and Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications (BUPT), China in 2012. He has published more than 80 referred journal papers in his research interests, which include cryptography, information security, cloud computing security etc. He has served on TPC member of several international conferences on information security and reviewer for more than 30 ISI indexed journals. He is a winner of the 2015 Journal of Network and Computer Applications Best Research Paper Award.

Maged Hamada Ibrahim received his Ph.D. from Helwan University, Egypt in 2005. He was promoted to Associate Professor in 2012 at Helwan University, Egypt. He is joining several network security projects in Egypt. His main interest is engineering cryptography and communications security. More specifically, working on the design of efficient and secure cryptographic algorithms and protocols, in particular, secure distributed multiparty computations, public key infrastructures, digital signatures, digital rights management protocols and non-cryptographic solutions to telecommunication security problems. He had published more than 40 journal and conference papers in his research fields.

Saru Kumari is currently an Assistant Professor with the Department of Mathematics, C.C.S. University, Meerut, U.P, India. She received Ph.D. degree in Mathematics in 2012 from C.C.S. University, Meerut, Uttar Pradesh, India. She has published 45 papers in international journals and conferences including 30 research publications in SCI indexed journals. Her current research interests include Information Security, Digital Authentication and Security of Wireless Sensor Networks.

Arun Kumar Sangaiah has received his Doctor of Philosophy (PhD) degree in Computer Science and Engineering from the VIT University, Vellore, India. He is presently working as an Associate Professor in School of Computer Science and Engineering, VIT University, India. His area of interest includes software engineering, computational intelligence, wireless networks, bio-informatics, and embedded systems. He has authored more than 100 publications in different journals and conference of national and international repute. Moreover, he has carried out number of funded research projects for Indian government agencies. Also, he was registered a one Indian patent in the area of Computational Intelligence. Besides, Dr. Sangaiah is responsible for Editorial Board Member/Associate Editor of various international journals.

Vidushi Gupta now is a master degree candidate major Information Systems at University of Delhi, India.

Kim-Kwang Raymond Choo received the Ph.D. in Information Security from Queensland University of Technology, Australia. He currently holds the cloud technology endowed professorship at the University of Texas at San Antonio, and is an associate professor at University of South Australia. He was named one of 10 Emerging Leaders in the Innovation category of The Weekend Australian Magazine/Microsofts Next 100 series in 2009, and is the recipient of various awards including ESORICS 2015 Best Research Paper Award, Highly Commended Award from Australia New Zealand Policing Advisory Agency, British Computer Societys Wilkes Award, Fulbright Scholarship, and 2008 Australia Day Achievement Medallion. He is a Fellow of the Australian Computer Society, and a Senior Member of IEEE.