Computer Communications 160 (2020) 311–325

Contents lists available at ScienceDirect

Computer Communications

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

A provably secure and efficient anonymous mutual authentication and key agreement protocol for wearable devices in WBAN

Ankur Gupta [a,*], Meenakshi Tripathi [a], Aakar Sharma [b]

[a] Malaviya National Institute of Technology Jaipur, Rajasthan, India
[b] Indian Institute of Technology Jammu, Jammu and Kashmir, India

ARTICLE INFO

Keywords:
WBAN
Healthcare
Mutual authentication
Anonymity
BAN-Logic
ROR model
AVISPA tool

ABSTRACT

Wireless body area networks (WBAN) is a novel paradigm that is gaining popularity in a scenario of current wireless communication systems. It plays an essential role in healthcare applications like remote monitoring of health data. For instance, the crucial and confidential data about the condition of the patient's physical health can be gathered and transferred through WBAN. Therefore, authentication and session key-agreements are integral security concerns for wearable sensors in WBAN. Moreover, as the wearable devices are resource-constraints, there is a need to develop a lightweight protocol to ensure authenticity, confidentiality, and integrity of the information. Li et al. presented an anonymous mutual authentication protocol to establish a session-key among wearable sensor nodes and the local hub node. However, after an in-depth analysis, we found that their scheme is susceptible to an intermediate node capture attack, and sensor node/hub node impersonation with intermediate node capture attacks. The scheme also does not provide anonymity with unlinkable sessions. This paper proposes a new anonymous mutual authentication and key agreement protocol in WBAN to overcome the security weaknesses in Li et al.'s protocol. The proposed protocol uses only basic symmetric cryptosystems like simple XOR and cryptographic hash functions; hence, it is efficient and lightweight. The validity and the correctness of the proposed protocol are evaluated using BAN-Logic, Real-Or-Random (ROR) model, and the broadly accepted AVISPA tool. The performance comparison of the proposed protocol with the existing related protocols shows the efficiency regarding communication and computational complexities. Hence, it is suitable to be used in real-life applications.

1. Introduction

Internet of Things (IoT) is becoming popular nowadays both from the technical and commercial point of view due to its simplicity, low cost, and easy deployment [1]. The rapid growth of sensor nodes in wireless networks leads to huge consumption of bandwidth and energy reducing battery life. Many resource allocation and optimization algorithms [2,3] result in reducing energy consumption in heavy applications of wireless sensor networks for industrial systems security and confidentiality [4,5]. Wireless body area network(WBAN) [6,7] is an essential application of IoT, which plays a significant role in healthcare services [8] to collect real-time vital health data of a patient. WBAN helps a doctor to monitor the patient's health state remotely via wireless communication technologies [9]. The wearable sensors attached to the patient's body collect sensitive and private information of a user [10]. This data helps the medical advisor to diagnose the patient's health condition for the treatment of the various diseases. Hence, for the privacy and security of a person, it is necessary to ensure that only authorized personnel can have access to this data. This scenario indicates the usefulness of secure mutual authentication and key agreement schemes for the wireless network. The wearable devices [11] are resource constraints, i.e., have limited capabilities in terms of communication and processing power, therefore high computing security mechanisms like AES [12], RSA [13], Diffie–Hellman [14], etc. cannot be implemented in WBAN. The overall energy consumptions of AES, DES, RSA, ECC as well as hash operations have been shown in the paper [15].

Fig. 1 shows the multi-hop centralized architecture for the wireless Body Area Network system. It consists of three types of nodes (*i*) second-level nodes or wearable sensing devices, (*ii*) first-level nodes or intermediate/gateway nodes, and (*iii*) the hub node. The central node, also known as the hub node or local server, collects all the physiological information from the sensor nodes via a gateway/mobile device. This architecture is divided into three tiers, as shown in Fig. 1. The first tier connects the second-level nodes or wearable devices with the first-level nodes or gateway/mobile device. Here, wearable devices sense the patient's health data such as blood pressure, heart rate, sleep cycle,

* Corresponding author.
E-mail addresses: 2015rcp9543@mnit.ac.in (A. Gupta), mtripathi.cse@mnit.ac.in (M. Tripathi).

https://doi.org/10.1016/j.comcom.2020.06.010
Received 12 February 2020; Received in revised form 3 May 2020; Accepted 12 June 2020
Available online 17 June 2020
0140-3664/

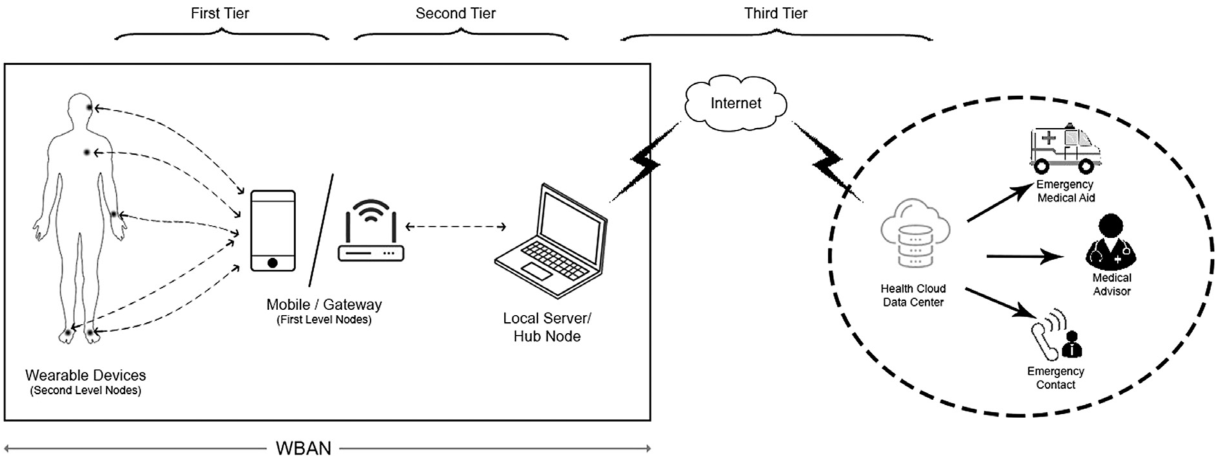

Fig. 1. Architecture of a WBAN system.

body temperature, ECG and EEG, and send it to the mobile/gateway device. The second tier represents the connection between the first-level node and the hub node, where an intermediate node forwards the received data from the wearable sensors to the hub node or local server. The third tier connects the hub node to the health cloud data center via internet.

The hub node sends all the information to the cloud data center where all the critical decisions are taken for the patients, such as to provide emergency medical aid, call an ambulance, etc. The information administered in the wireless BAN is highly sensitive and confidential; therefore, security and privacy become significant issues that must be guaranteed. Moreover, it also becomes a pivotal challenge to enable mutual authentication and secure shared cryptographic key establishment in a resource-constraint architecture, i.e., having limited computation and communication abilities. Li et al. [16] designed a scheme having anonymous mutual authentication and key agreement components for Wireless BAN. Their scheme provided features like mutual authentication, secrecy, security against different known attacks such as replay, eavesdropping, man-in-the-middle attacks, etc. However, we analyzed that this scheme is vulnerable to the intermediate node capture attack, sensor node impersonation and hub node impersonation with intermediate node capture attacks. Also, this scheme does not provide anonymity with unlinkable sessions.

1.1. Motivation and contribution

The wearable devices used in the healthcare monitoring systems are resource-constraints. Therefore, the authentication and key-agreement protocol must be lightweight as well as secure to protect the sensitive and confidential information of a patient. It is a challenging task to design such an authentication protocol that also facilitates numerous security features. This motivated us to design a provably secure and efficient anonymous mutual authentication and key agreement protocol for wearable devices in WBAN. Our protocol uses basic symmetric cryptosystems like simple XOR and cryptographic hash functions; hence, it is efficient and lightweight. The main contributions of this paper are as follows:

- We first analyze the security of Li et al.'s protocol and deduce that it is susceptible to various attacks.
- We propose a provably secure and efficient anonymous mutual authentication and key agreement protocol to provide security against well-known attacks.
- We prove the establishment of secure session-key and resilience to various known attacks by using BAN-Logic, ROR model and AVISPA tool.
- Finally, we show the efficiency of the improved scheme regarding storage, computational, and communication costs.

The rest of the paper is organized as follows. Section 2 discusses the existing related work done in this field. Section 3 presents the system model used throughout the paper. Section 4 reviews Li et al.'s protocol in detail. Section 5 discusses the security analysis of Li et al.'s protocol. Section 6 presents an improved protocol in detail. Section 7 gives the security analysis of our proposed protocol. Section 8 provides the comparative analysis of the proposed protocol with Li et al.'s protocol and the other related existing schemes. Finally, Section 9 concludes the paper.

2. Related work

In recent years, numerous research has been proposed in the field of authentication and key-establishment [17] for enhancing the security of wireless sensor networks. Most protocols focus on the establishment of secure session-key based on asymmetric key cryptosystems like AES, RSA, ElGamal, ECC, Paillier cryptosystem, etc., but require high resource utilization such as computation and communication power. Such cryptosystems are not suitable for energy constraint WSN environment, especially in the area of wireless body area networks (WBAN) where wearable devices are highly resource-constrained. Hence, lightweight encryption techniques are gaining popularity in the aspect of WSN security based on symmetric cryptosystem.

In 2006, Wong et al. [18] introduced a lightweight user authentication scheme based only on XOR and hash operations for resource-constrained WSN. Unfortunately, this scheme was susceptible to multiple attacks like stolen verifier, forgery, and replay attacks found by Das [19]. Das enhanced the security of this scheme in 2009 by adding third-party user authentication with the help of the gateway node. Khan et al. [20] found insider attack, impersonation, and node-capture attacks in Das's scheme and proposed an improved scheme with hashed password. Vaidya et al. [21] later proved that Khan et al.'s scheme was also susceptible to stolen smart card and impersonation attack. Many smart-card based remote user authentication schemes had also been proposed in the past [22–24].

Turkanovic et al. [25] proposed a novel user authentication scheme for heterogeneous WSN claiming user anonymity and secure mutual authentication using simple XOR and hash computation. Later [26–28] showed various security flaws such as stolen smart card attack, user impersonation attack, and unassertive backward secrecy. In 2016, Gope et al. [29] introduced a lightweight realistic authentication protocol in WSN implementing security features such as perfect forward secrecy, untraceability, user anonymity, etc. Unfortunately, Jolfaie et al. [30] showed security drawbacks in Gope et al.'s scheme i.e., vulnerable in disclosure of session-key.

In 2017, Li et al. [16] proposed a lightweight anonymous mutual authentication protocol for WBAN having centralized 2-hop architecture.

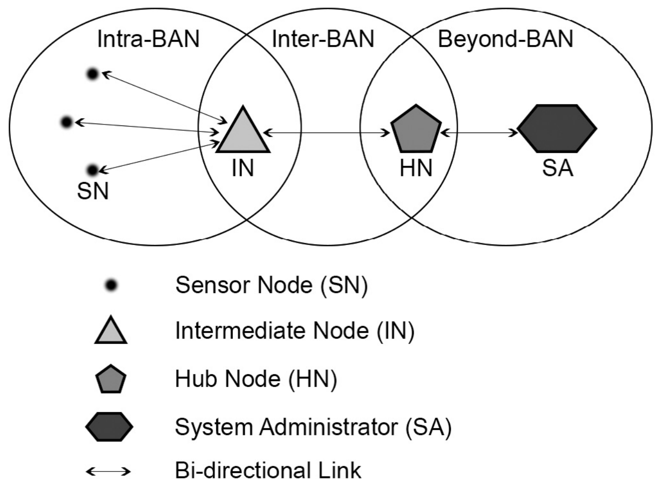

Fig. 2. Network model for WBAN.

Their protocol provided anonymity, perfect forward/backward secrecy with resilient to many attacks. However, in-depth analysis establishes that their scheme was vulnerable to intermediate node capture attack, sensor node impersonation and hub node impersonation with intermediate node capture attacks. Also, Chen et al. [31] found similar attacks in Li et al.'s scheme and Koya et al. [32] proved their scheme has a key-escrow problem. In 2018, Chen et al. proposed a repaired protocol providing better anonymity and security against impersonation attacks but their scheme is vulnerable to first-level node capture attack, sensor node and hub node impersonation with first-level node capture attacks similar to Li et al.'s scheme. Therefore, it is not suitable for practical applications. Also, Kompara et al. [33] highlighted that Koya et al.'s scheme resolves some of Li et al.'s drawbacks using physiological signals but making their scheme highly computational. The main issue is collecting and transforming physiological signals at all sensors, and the additional cost is required to keep all the sensors synchronized. Gupta et al. [34] also proposed a lightweight anonymous user authentication scheme for wearable devices. The technique provided secure mutual authentication with less computational and communication overheads. It is based on the assumption that the mobile terminal/gateway node cannot be captured by an attacker.

From the above analysis, we found that there is a need of more secure as well as a lightweight authentication protocol for the wearable devices in an IoT environment to provide secure communication of private information of patient's health. Hence, the proposed protocol aims to provide a lightweight secure authentication process to help exchange confidential data with efficiency.

3. System model

In this Section, we introduce the two models followed in Li et al. and our proposed protocol.

3.1. Network model

The centralized network model for wireless body area networks (WBAN) is shown in Fig. 2 containing four types of network nodes — the Sensor Node (SN), the Intermediate Node (IN), the Hub Node (HN) and the System Administrator (SA). The sensor nodes are resource-constrained wearable devices that sense the real-time health data of a patient. Sensor nodes forward the collected data to the hub node via an intermediate node. The connection between the sensor node and an intermediate node represents an Intra-BAN communication. The intermediate node has more computational, and communication capabilities than sensor nodes and are responsible for transferring the data to the hub node. An intermediate node must validate the legitimacy of a sensor node before receiving any data. It helps both the sensor node and the hub node to establish a session key to communicate securely. The connection between the intermediate node and the hub node represents Inter-BAN communication. The hub node processes the patient's physiological vital information and transmits the critical data to the system administrator over a public network for further analysis and storage to the healthcare service provider servers.

Table 1
Notations used throughout the paper.

Symbols	Explanations
SA	System Administrator
SN	Sensor Node
HN	Hub Node
IN	Intermediate Node
id_n	Identity of a sensor node
id_{in}	Identity of an intermediate node
tid_n	Temporary identity of a sensor node
K_{hn}	Secret Key of a hub node
K_n	Temporary secret key of sensor node
X_{n-in}	Long-Term shared secret key
t_i	Current timestamp
ΔT	Maximum transmission delay
K_s	Session key between SN and HN
h(·)	Cryptographic hash function
$\oplus$	Bitwise XOR Operation
$\|$	Concatenation Operation

3.2. Threat model

Here, we adopted the well known Dolev–Yao threat model [35] in which all entities involved in a communication transmit the messages over an unsecured channel. In this model

- It is assumed that an adversary knows the authentication protocol used and may control the public channel completely.
- An adversary can eavesdrop all the communications link or modify, corrupt, redirect, delete or replay any message transmitted over an unsecured channel.
- An adversary may also physically capture any number of sensor nodes and able to extract the stored information from memory using power analysis attack.
- In addition, an adversary may also be able to physically capture an intermediate node and extract all the stored information from its memory.
- However, an adversary cannot intercept the message transmitted over a secure channel.

We analyze the security of our protocol using this model.

4. Review of Li et al.'s [16] protocol

In this section, we shortly review the anonymous mutual authentication and key agreement protocol proposed by Li et al. for wearable devices in WBAN. The scheme has three phases namely initialization phase, registration phase, and authentication phase. We present the detailed overview of these phases of Li et al.'s protocol in Fig. 3 to find out the security weakness in this scheme. Table 1 summarizes all the notations used in Li et al.'s and in our improved protocol throughout the paper.

4.1. Initialization phase

According to Li et al.'s scheme, this phase is performed by System Administrator (SA) in an offline mode. The System Administrator (SA) initializes the Hub Node (HN) by selecting a master secret key K_{HN}. Then, SA stores the master key K_{HN} in HN's memory.

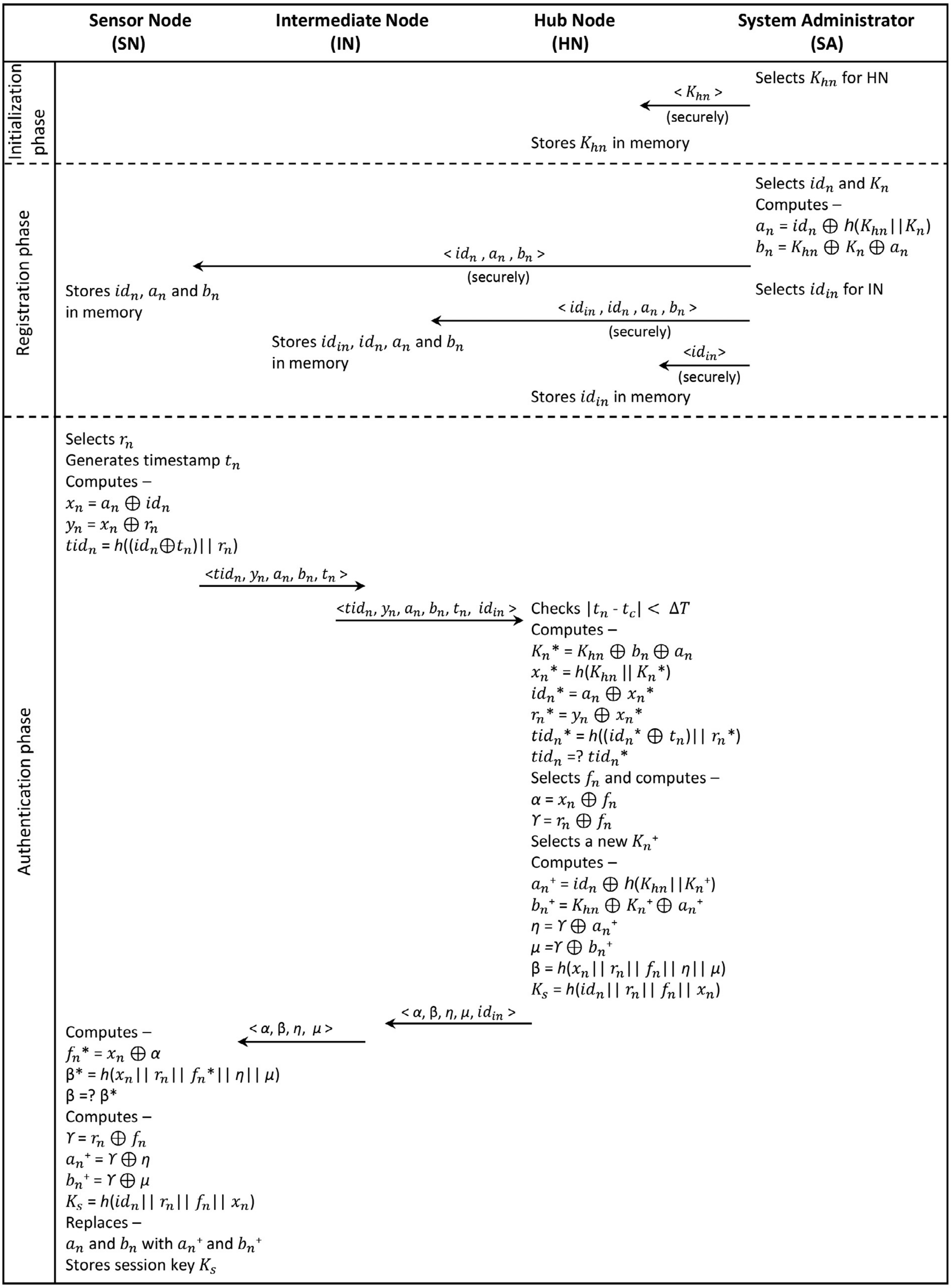

Fig. 3. Initialization, registration and authentication phase in Li et al.'s protocol.

4.2. Registration phase

The System Administrator (SA) registers a sensor node (SN) by executing the following steps:

- SA chooses a unique secret identity id_n and a temporary secret key K_n for SN.
- It then computes $a_n = id_n \oplus h(K_{hn} \parallel K_n)$ and $b_n = a_n \oplus K_{hn} \oplus k_n$.
- SA also chooses a unique identity id_{in} for the intermediate node (IN).
- It finally stores $\langle id_n, a_n, b_n \rangle$ in the sensor node (SN's) memory and $\langle id_{in}, id_n, a_n, b_n \rangle$ in the intermediate node (IN's) memory.
- It also stores the unique identity id_{in} of Intermediate node (IN) in HN's memory.

4.3. Authentication phase

This phase executes the following steps:

- The sensor node (SN) first computes $x_n = a_n \oplus id_n$ and chooses a random number r_n to compute $y_n = x_n \oplus r_n$.
- It then computes the temporary identity $tid_n = h((id_n \oplus t_n) \parallel r_n)$, where t_n is the current timestamp, and sends $\langle tid_n, y_n, a_n, b_n, t_n \rangle$ to the intermediate node (IN) via an unsecure channel.
- The intermediate node (IN) forwards this received message by adding its own identity id_{in} i.e., $\langle tid_n, y_n, a_n, b_n, t_n, id_{in} \rangle$ to HN.
- Upon receiving this message, the hub node (HN) checks the id_{in} in its database and verifies the timestamp validity i.e., $|t_n - t_c| < \Delta T$ or not, where t_c is the current timestamp of HN.
- If any check fails, HN aborts the connection. Otherwise, HN computes $K_n^* = a_n \oplus b_n \oplus K_{hn}$, $x_n^* = h(K_{hn} \parallel k_n^*)$, $id_n^* = a_n \oplus x_n^*$ and $r_n^* = y_n \oplus x_n^*$ to calculate $tid_n^* = h((id_n^* \oplus t_n) \parallel r_n^*)$ and checks $tid_n^* = tid_n$ or not.
- If check fails, it aborts the connection. Otherwise, HN chooses f_n randomly and computes $\alpha = x_n \oplus f_n$ and $\gamma = r_n \oplus f_n$. It also selects new K_n^+ and computes $a_n^+ = id_n \oplus h(K_{hn} \parallel K_n^+)$, $b_n^+ = a_n^+ \oplus K_{hn} \oplus K_n^+$, $\eta = \gamma \oplus a_n^+$, $\mu = \gamma \oplus b_n^+$ and $\beta = h(x_n \parallel r_n \parallel f_n \parallel \eta \parallel \mu)$.
- HN finally computes the session key as $K_s = h(id_n \parallel r_n \parallel f_n \parallel x_n)$ and sends the message $\langle \alpha, \beta, \eta, \mu, id_{in} \rangle$ to IN.
- IN drops its own identity id_{in} from the received message and forwards it to the sensor node SN.
- Upon receiving the message, SN computes $f_n^* = x_n \oplus \alpha$, $\beta^* = h(x_n \parallel r_n \parallel f_n^* \parallel \eta \parallel \mu)$ and checks whether $\beta = \beta^*$ or not. If they are not equal, SN aborts the connection. Otherwise, it calculates $\gamma = r_n \oplus f_n$, $a_n^+ = \gamma \oplus \eta$, $b_n^+ = \gamma \oplus \mu$ and computes session key K_s as $K_s = h(id_n \parallel r_n \parallel f_n \parallel x_n)$. It also updates the a_n^+, b_n^+ in its memory.

5. Security analysis of Li et al.'s scheme

This section provides the security weaknesses found in Li et al.'s protocol. The protocol has several security shortcomings such as intermediate node capture attack, sensor node impersonation attack, hub node impersonation attack, Linkable sessions etc. The description of the following attacks in Li et al. is presented below:

5.1. Intermediate node capture attack

In this attack, an adversary $\mathcal{A}$ is able to compromise the intermediate node and access all the secret information stored in it. An adversary thus knows $\langle id_n, a_n, b_n \rangle$ and therefore able to calculate $x_n = a_n \oplus id_n$ and $r_n = x_n \oplus y_n$. Next, $\mathcal{A}$ computes $f_n = x_n \oplus \alpha$ and the session key $K_s = h(id_n \parallel r_n \parallel f_n \parallel x_n)$. An adversary $\mathcal{A}$ has successfully computed the session key and compromised the security of the protocol.

5.2. Sensor node impersonation attack

In order to impersonate as a legitimate sensor node, an adversary $\mathcal{A}$ must send a valid login request to the hub node and if the hub node accepts the falsify request, that means an adversary $\mathcal{A}$ has successfully impersonate as a legal sensor node. Suppose $\mathcal{A}$ had already compromised the intermediate node as shown in Section 5.1. $\mathcal{A}$ generates a random number r_n and computes $x_n = a_n \oplus id_n$, $y_n = x_n \oplus r_n$ and $tid_n = h((id_n \oplus t_n) \parallel r_n)$ and sends $\langle tid_n, y_n, a_n, b_n, t_n \rangle$ to the intermediate node. Upon receiving $\langle \alpha, \beta, \eta, \mu \rangle$, $\mathcal{A}$ computes $f_n = x_n \oplus \alpha$ and the session key $K_s = h(id_n \parallel r_n \parallel f_n \parallel x_n)$. Therefore, $\mathcal{A}$ has succeeded in impersonating as the sensor node.

5.3. Hub node impersonation attack

In order to impersonate as a legitimate hub node, an adversary $\mathcal{A}$ must generate an authentic message $\langle \alpha, \beta, \eta, \mu \rangle$. After compromising the intermediate node as shown in Section 5.1, $\mathcal{A}$ computes $x_n = a_n \oplus id_n$ and $r_n = y_n \oplus x_n$. $\mathcal{A}$ selects f_n and computes $\alpha = x_n \oplus f_n$, $\gamma = r_n \oplus f_n$. It also generates fake K_{hn}^* and K_n^*, and computes $a_n^+ = id_n \oplus h(K_{hn}^* \parallel K_n^*)$, $b_n^+ = K_{hn}^* \oplus K_n^* \oplus a_n^+$, $\eta = \gamma \oplus a_n^+$, $\mu = \gamma \oplus b_n^+$, $\beta = h(x_n \parallel r_n \parallel f_n \parallel \eta \parallel \mu)$ and $K_s = h(id_n \parallel r_n \parallel f_n \parallel x_n)$ and sends the message $\langle \alpha, \beta, \eta, \mu \rangle$ to the intermediate node. Therefore, $\mathcal{A}$ has succeeded in impersonating as the hub node to the sensor node.

5.4. Linkable sessions and traceability

If an attacker $\mathcal{A}$ cannot link two different sessions with the same sensor node, then we can say that sessions are unlinkable and untraceable. However, Li et al.'s protocol do not satisfy this property, and an attacker $\mathcal{A}$ can link two messages originating from the same sensor node. $\mathcal{A}$ gets the value y_n and α by simply eavesdropping the publicly sent messages. By XORing y_n and α, an attacker $\mathcal{A}$ knows γ, which is used to construct η and μ using new a_n^+ and b_n^+, respectively. These a_n^+ and b_n^+ will now be used by the same sensor in the next authentication process, and therefore $\mathcal{A}$ can link these two sessions with a single sensor node. Hence, unlinkability and untraceability property are not satisfied by Li et al.'s protocol.

6. Proposed improved scheme

Unlike Li et al.'s protocol, the proposed protocol has four phases namely initialization, registration, authentication, and dynamic node update phase. The initialization and registration phase is shown in Fig. 4.

6.1. Initialization phase

The system administrator (SA) starts the initialization process. SA selects a master key K_{hn} for Hub node (HN) and stores it securely in the HN's memory.

6.2. Registration phase

The System Administrator (SA) registers the intermediate node (IN) and sensor node (SN) as follows:

- SA chooses a unique identity id_n and a temporary secret key K_n for SN.
- It also selects a unique secret shared key X_{n-in} for intermediate node and sensor node.
- It then computes $a_n = id_n \oplus h(K_{hn} \parallel K_n)$, $b_n = K_{hn} \oplus K_n \oplus a_n$, and $c_n = h(K_{hn} \parallel id_n)$.
- It also chooses a unique identity id_{in} for the intermediate node (IN).

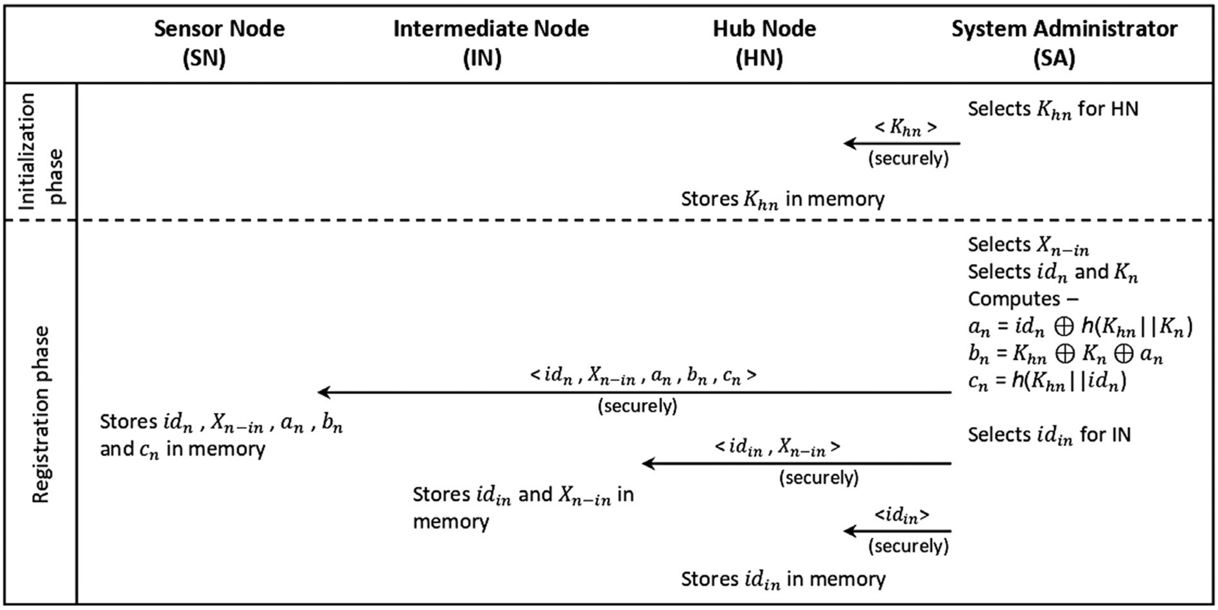

Fig. 4. Initialization and registration phase of our proposed protocol.

- It finally stores $\langle id_n, X_{n-in}, a_n, b_n, c_n\rangle$ in the sensor node (SN's) memory and $\langle id_{in}, X_{n-in}\rangle$ in the intermediate node (IN's) memory.
- It also stores the unique identity id_{in} of intermediate node (IN) in HN's memory.

The System Administrator does not store K_n anywhere. It is just used for the creation of a_n and b_n only.

6.3. Authentication phase

The authentication phase of the proposed protocol is shown in Fig. 5. This phase executes the following steps:

- The sensor node (SN) selects a random number r_n and computes $x_n = a_n \oplus id_n$ and $y_n = x_n \oplus r_n$.
- It also generates the current timestamp t_1 to compute the temporary identity $tid_n = h((id_n \oplus t_1) \parallel r_n)$ and $V_n = h(X_{n-in} \parallel t_1)$ and sends $\langle tid_n, y_n, a_n, b_n, V_n, t_1\rangle$ to the intermediate node (IN) via an unsecure channel.
- Upon receiving this message, the intermediate node (IN) first checks the timestamp validity i.e., $|t_1 - t_c| < \Delta T$ or not, where t_c is the current timestamp of IN. It also computes $V_n^* = h(X_{n-in} \parallel t_1)$ and checks whether $V_n^* = V_n$ or not.
- If any check fails, IN aborts the connection. Otherwise, it generates the current timestamp t_2 to compute $V_i = h(id_{in} \parallel t_2)$ and sends $\langle tid_n, y_n, a_n, b_n, V_i, t_1, t_2\rangle$ to the hub node (HN) via an unsecure channel.
- The hub node (HN) checks the timestamp validity i.e., $|t_2 - t_c| < \Delta T$ or not, where t_c is the current timestamp of HN. It then computes $V_i^* = h(id_{in} \parallel t_2)$ and checks whether $V_i^* = V_i$ or not.
- If the check fails, HN aborts the connection. Otherwise, it computes $K_n^* = K_{hn} \oplus b_n \oplus a_n$, $x_n^* = h(K_{hn} \parallel k_n^*)$, $id_n^* = a_n \oplus x_n^*$ and $r_n^* = y_n \oplus x_n^*$. HN then calculates $tid_n^* = h((id_n^* \oplus t_1) \parallel r_n^*)$ and checks whether $tid_n^* = tid_n$ or not.
- After successful verification, HN selects new K_n^+ and generates f_n randomly. It then computes $a_n^+ = id_n \oplus h(K_{hn} \parallel K_n^+)$, $b_n^+ = K_{hn} \oplus K_n^+ \oplus a_n^+$, $\alpha = h(K_{hn} \parallel id_n) \oplus f_n$, $\eta = h(h(K_{hn} \parallel id_n) \parallel f_n) \oplus a_n^+$, $\mu = h(h(K_{hn} \parallel id_n) \parallel r_n) \oplus b_n^+$ and $\beta = h(x_n \parallel r_n \parallel f_n \parallel \eta \parallel \mu)$.
- HN generates the current timestamp t_3 and computes the session key as $K_s = h(id_n \parallel r_n \parallel f_n \parallel h(K_{hn} \parallel id_n))$ and $V_h = h(id_{in} \parallel t_3)$. It then sends $\langle \alpha, \beta, \eta, \mu, V_h, t_3\rangle$ to IN.
- IN first verifies the timestamp validity i.e., $|t_3 - t_c| < \Delta T$ or not, and computes $V_h^* = h(id_{in} \parallel t_3)$ to check whether $V_h^* = V_h$ or not.
- If any check fails, IN terminates the connection. Otherwise, it generates the current timestamp t_4 to compute $V_i = h(X_{n-in} \parallel t_4)$ and sends $\langle \alpha, \beta, \eta, \mu, V_i, t_4\rangle$ to SN.
- Upon receiving this message, SN checks the timestamp validity i.e., $|t_4 - t_c| < \Delta T$ or not, where t_c is the current timestamp of SN. It then computes $V_i^* = h(X_{n-in} \parallel t_4)$ and checks whether $V_i^* = V_i$ or not.
- SN then computes $f_n^* = c_n \oplus \alpha$, $\beta^* = h(x_n \parallel r_n \parallel f_n^* \parallel \eta \parallel \mu)$ and checks whether $\beta = \beta^*$ or not. If they are not equal, it aborts the connection. Otherwise, it calculates $a_n^+ = h(c_n \parallel f_n) \oplus \eta$, $b_n^+ = h(c_n \parallel r_n) \oplus \mu$ and computes session key K_s as $K_s = h(id_n \parallel r_n \parallel f_n \parallel c_n)$. It also updates a_n^+ and b_n^+ in its memory.

6.4. Dynamic node update phase

It may be the case that a new wearable sensor is required to sense some data. Therefore a new node must be added dynamically into a wireless body area networks. Fig. 6 shows the dynamic node update phase of the proposed scheme. If a new wearable device enters in an existing network say SN^{new}, System Administrator (SA) performs the following steps:

- SA picks a unique identity $id_n{}^{new}$ and a temporary secret key $K_n{}^{new}$ for SN^{new}.
- It then computes $a_n{}^{new} = id_n{}^{new} \oplus h(K_{hn} \parallel K_n{}^{new})$, $b_n{}^{new} = K_{hn} \oplus K_n{}^{new} \oplus a_n{}^{new}$ and $c_n{}^{new} = h(K_{hn} \parallel id_n{}^{new})$.
- It also selects a unique secret shared key $X_{n-in}{}^{new}$ and stores the information $\langle id_n{}^{new}, X_{n-in}{}^{new}, a_n{}^{new}, b_n{}^{new}, c_n{}^{new}\rangle$ in the sensor node's (SN^{new}) memory securely before its deployment.
- It finally adds $X_{n-in}{}^{new}$ in intermediate node's (IN) memory securely.

7. Security analysis of our proposed protocol

This section analyzes the security of the proposed protocol using both the formal and the informal security analysis methods. The formal security analysis for the proposed scheme is done using BAN-Logic, real-or-random (ROR) model, and the widely accepted AVISPA tool.

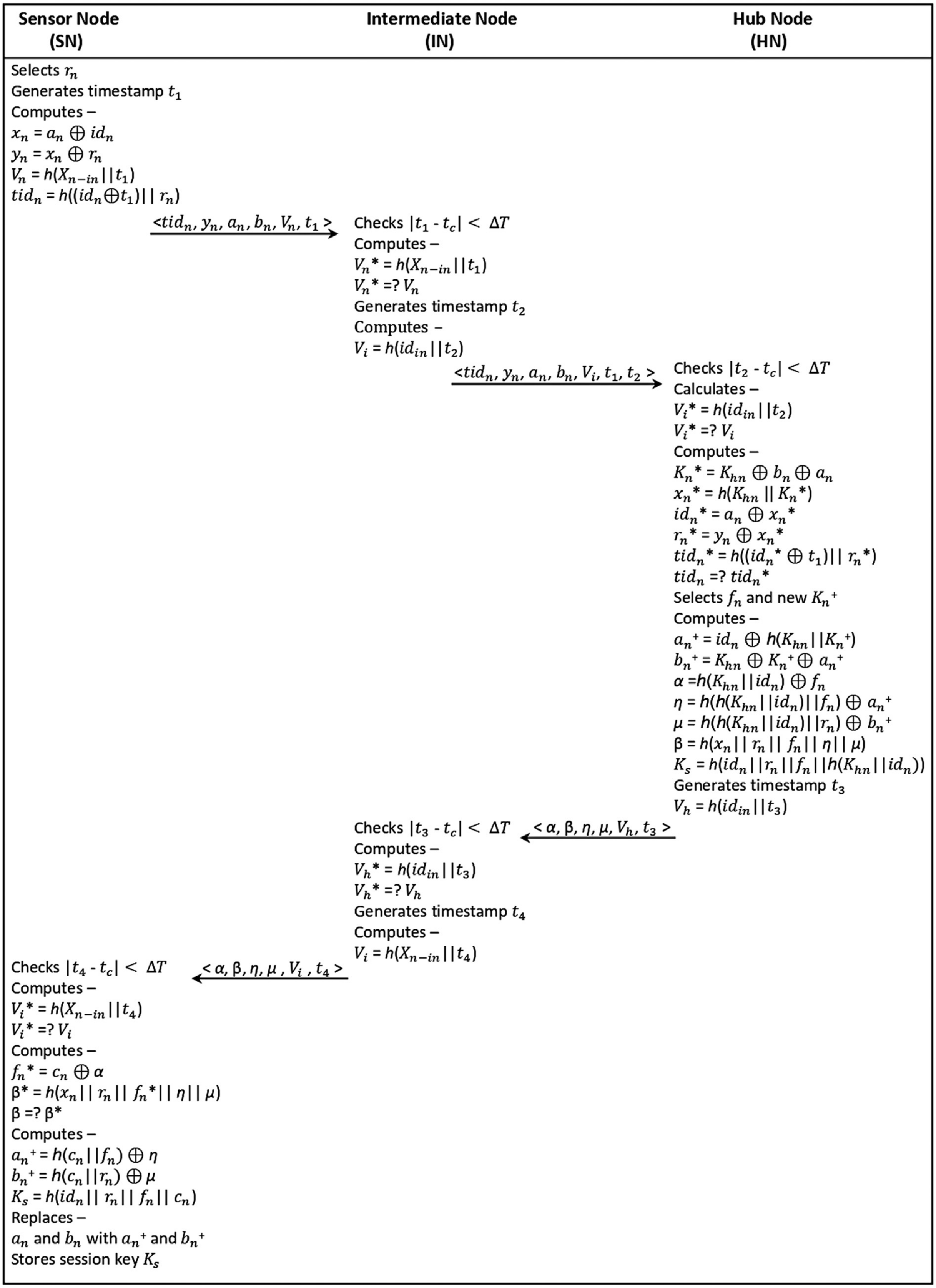

Fig. 5. Authentication phase of our proposed protocol.

BAN-Logic proves that the proposed protocol establishes a secure mutually authenticated session-key between a sensor node and the hub node. ROR model proves the semantic security (session-key security against an adversary attack) of the proposed scheme while the AVISPA

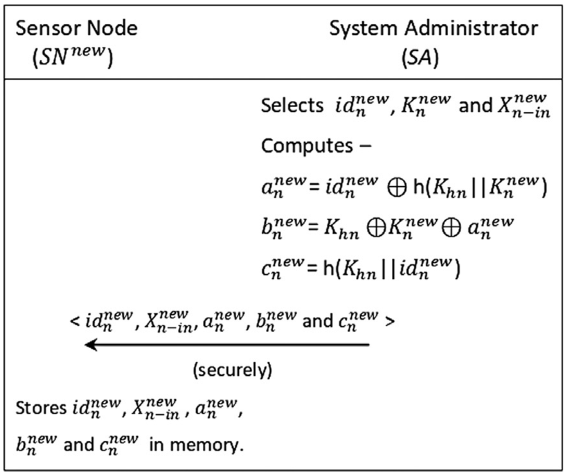

Fig. 6. Dynamic node update phase of our proposed protocol.

security analyzer tool ensures the security of the proposed protocol against replay, impersonation and man-in-the-middle attack. Furthermore, non-mathematical informal security analysis is also shown to prove the resilience of the proposed scheme against various popular known attacks.

7.1. Mutual authentication using BAN-logic

This section provides the brief description of the security analysis of the proposed scheme using Burrows–Abadi–Needham (BAN)logic [36]. Our BAN-Logic proof is similar to the proofs shown in [16,32]. The following basic logical notations are used to analyze our scheme:

- $A \mid\equiv S$: A believes the statement S
- $A \triangleleft S$: A sees the statement S
- $A \mid\sim S$: A once said S
- $A \mid\Rightarrow S$: A has jurisdiction over S
- $\#(S)$: S is a fresh statement
- $A \overset{K}{\leftrightarrow} B$: K is a shared secret key between A and B
- $\{S\}_K$: S is encrypted with key K
- $(S)_K$: S is hashed with key K
- (S, Y) : S or Y is one part of the formula (S, Y)

The following rules are used in the BAN-logic to describe the main logical postulates:

7.1.1. Message meaning rule

If A sees a statement S encrypted with key K and A believes K is a shared secret key between A and B, then A believes B once said S.

$$\frac{A \mid\equiv A \overset{k}{\leftrightarrow} B, A \triangleleft \{S\}_k}{A \mid\equiv B \mid\sim S}$$

7.1.2. Nonce verification rule

If A believes that the statement S is fresh and A also believes that B once said S, then A believes B believes the statement S.

$$\frac{A \mid\equiv \#\{S\}, A \mid\equiv B \mid\sim S}{A \mid\equiv B \mid\equiv S}$$

7.1.3. Jurisdiction rule

If A believes B has jurisdiction over the statement S and A believes B believes the statement S, then A believes the statement S.

$$\frac{A \mid\equiv B \mid\Rightarrow S, A \mid\equiv B \mid\equiv S}{A \mid\equiv S}$$

7.1.4. Freshness rule

If A believes the part of the statement S is fresh, then A believes that the statement $\{S, Y\}$ is fresh.

$$\frac{A \mid\equiv \#(S)}{A \mid\equiv \#(S, Y)}$$

7.1.5. Belief rule

If A believes B believes the statement (S, Y), then A believes B believes the part of the statement S.

$$\frac{A \mid\equiv B \mid\equiv (S, Y)}{A \mid\equiv B \mid\equiv S}$$

Our main goal is to prove:

- Goal 1: $HN \mid\equiv (SN \overset{x_n}{\leftrightarrow} HN)$
- Goal 2: $HN \mid\equiv SN \mid\equiv (SN \overset{x_n}{\leftrightarrow} HN)$
- Goal 3: $SN \mid\equiv (SN \overset{K_s}{\leftrightarrow} HN)$
- Goal 4: $SN \mid\equiv HN \mid\equiv (SN \overset{K_s}{\leftrightarrow} HN)$

In our scheme, the messages sent over the insecure channel are:

- M1: $SN \to IN : \langle tid_n, y_n, a_n, b_n, V_n, t_1 \rangle$
- M2: $IN \to HN : \langle tid_n, y_n, a_n, b_n, V_i, t_1, t_2 \rangle$
- M3: $HN \to IN : \langle \alpha, \beta, \eta, \mu, V_h, t_3 \rangle$
- M4: $IN \to SN : \langle \alpha, \beta, \eta, \mu, V_i, t_4 \rangle$

Therefore, the idealized form of the messages transferred in the authentication phase between a sensor node (SN) and the hub node (HN) are:

Msg1: $SN \to HN : \langle SN \overset{x_n}{\leftrightarrow} HN, r_n, t_i \rangle_{SN \overset{id_n}{\longleftrightarrow} HN}$

Msg2: $HN \to SN : \langle SN \overset{x_n}{\leftrightarrow} HN, r_n, f_n, SN \overset{K_s}{\leftrightarrow} HN \rangle_{SN \overset{id_n}{\longleftrightarrow} HN}$

To prove the session key establishment between a sensor node and the hub node in the proposed protocol, the following assumptions are made:

- A1: $HN \mid\equiv (SN \overset{id_n}{\longleftrightarrow} HN)$
- A2: $HN \mid\equiv \#(t_i), SN \mid\equiv \#(t_i)$
- A3: $HN \mid\equiv SN \mid\Rightarrow SN \mid\sim (SN \overset{x_n}{\longleftrightarrow} HN)$
- A4: $SN \mid\equiv (SN \overset{id_n}{\longleftrightarrow} HN)$
- A5: $SN \mid\equiv \#(r_n), HN \mid\equiv \#(f_n)$
- A6: $SN \mid\equiv HN \mid\Rightarrow HN \mid\sim (SN \overset{K_s}{\longleftrightarrow} HN)$

Analysis: Now, We will analyze the proposed protocol using BAN-logic rules, postulates and assumptions.

From Message Msg1, A1, and MMR rule, we get

$$\frac{HN \mid\equiv (SN \overset{id_n}{\leftrightarrow} HN), HN \triangleleft \{SN \overset{x_n}{\leftrightarrow} HN, r_n, t_i\}_{SN \overset{id_n}{\longleftrightarrow} HN}}{HN \mid\equiv SN \mid\sim (SN \overset{x_n}{\leftrightarrow} HN, r_n, t_i)} \tag{1}$$

Using Eq. (1), A2 and FR, we get

$$\frac{HN \mid\equiv \#(t_i)}{HN \mid\equiv \#(SN \overset{x_n}{\longleftrightarrow} HN, r_n, t_i)} \tag{2}$$

Using Eqs. (1) and (2), and applying NVR, we get

$$\frac{HN \mid\equiv \#(SN \overset{x_n}{\leftrightarrow} HN, r_n, t_i), HN \mid\equiv SN \mid\sim (SN \overset{x_n}{\leftrightarrow} HN, r_n, t_i)}{HN \mid\equiv SN \mid\equiv (SN \overset{x_n}{\longleftrightarrow} HN, r_n, t_i)} \tag{3}$$

Next, from Eq. (3) and applying belief rule, we get

$$\frac{HN \mid\equiv SN \mid\equiv (SN \overset{x_n}{\longleftrightarrow} HN, r_n, t_i)}{HN \mid\equiv SN \mid\equiv (SN \overset{x_n}{\longleftrightarrow} HN)} \quad \textbf{Goal 2} \tag{4}$$

From Eq. (4), A3 and applying JR, we get

$$\frac{HN \mid\equiv SN \mid\Rightarrow (SN \overset{x_n}{\longleftrightarrow} HN), HN \mid\equiv SN \mid\equiv (SN \overset{x_n}{\longleftrightarrow} HN)}{HN \mid\equiv (SN \overset{x_n}{\longleftrightarrow} HN)} \quad \textbf{Goal 1} \tag{5}$$

From Message Msg2, A4 and MMR, we get

$$\frac{SN \mid\equiv (SN \overset{id_n}{\leftrightarrow} HN), SN \triangleleft \{c_n, f_n, r_n, SN \overset{K_s}{\leftrightarrow} HN\}_{SN \overset{id_n}{\longleftrightarrow} HN}}{SN \mid\equiv HN \mid\sim (c_n, f_n, r_n, SN \overset{K_s}{\leftrightarrow} HN)} \tag{6}$$

Using Eq. (6), A5 and FR, we get

$$\frac{SN \mid\equiv \#(r_n)}{SN \mid\equiv \#(c_n, f_n, r_n, SN \overset{K_s}{\longleftrightarrow} HN)} \tag{7}$$

From Eqs. (6), (7) and NVR rule, we get

$$\frac{SN \mid\equiv \#(Q_A, SN \overset{id_n}{\longleftrightarrow} HN), SN \mid\equiv HN \mid\sim (Q_A)}{SN \mid\equiv HN \mid\equiv (Q_A)} \tag{8}$$

where $Q_A = (c_n, f_n, r_n, SN \overset{K_s}{\longleftrightarrow} HN)$

From Eq. (8) and BR, we get

$$\frac{SN \mid\equiv HN \mid\equiv (c_n, f_n, r_n, SN \overset{K_s}{\longleftrightarrow} HN)}{SN \mid\equiv HN \mid\equiv (SN \overset{K_s}{\longleftrightarrow} HN)} \quad \textbf{Goal 4} \tag{9}$$

From Eq. (9), A6 and JR we get

$$\frac{SN \mid\equiv HN \mid\Rightarrow (SN \overset{K_s}{\leftrightarrow} HN), SN \mid\equiv HN \mid\equiv (SN \overset{K_s}{\leftrightarrow} HN)}{SN \mid\equiv (SN \overset{K_s}{\longleftrightarrow} HN)} \quad \textbf{Goal 3} \tag{10}$$

Hence, the proposed scheme obtains mutual authentication and session key establishment between a sensor node and the hub node.

7.2. Formal security analysis based on ROR model

This section thoroughly explains the security analysis of the proposed scheme using probabilistic mathematical model i.e. ROR model [37,38]. ROR model is used to prove the session-key security i.e., the sustainment of the session-key against active and passive attacks by the adversary. First, we provide the basic description of the ROR model and then show the mathematical proof subsequently.

7.2.1. ROR model

The participants involved in the network are (i) Sensor Node (SN), (ii) Intermediate Node (IN), and (iii) Hub Node (HN). The following components are used in the ROR model:

- **Participants:** Let the instances t_1, t_2 and t_3 of SN, IN and HN be denoted by $\pi^{t_1}_{SN}, \pi^{t_2}_{IN}$ and $\pi^{t_3}_{HN}$ respectively. These instances are also termed as oracles.
- **Accepted state:** An instance π^t is in an accepted state if it jumps to the accept state after receiving the last expected message of the protocol. When all the communicated (received and sent) messages of the instance π^t are concatenated in order, it will represent session identification (sid) of π^t for the current session.
- **Partnering:** Two instances π^{t_1} and π^{t_2} are partnered to each other if (i) both the instances π^{t_1} and π^{t_2} are in accept state, (ii) they share same sid and also mutually authenticate each other, and (iii) both π^{t_1} and π^{t_2} are mutual partners.
- **Freshness:** $\pi^{t_1}{}_{SN}$ or $\pi^{t_3}{}_{HN}$ is called fresh if the session-key SK created between SN and HN is not disclosed to the adversary $\mathcal{A}$ by using the Reveal query ($RVL(\pi^t)$).
- **Adversary:** ROR model uses Dolev–Yao threat model in which an adversary has full control over the communicational network i.e., $\mathcal{A}$ can eavesdrop, modify, delete or construct new messages in the network where legitimate devices are communicating to each other. $\mathcal{A}$ can ascertain the following queries [39]:
  1. $EXE(\pi^{t_1}, \pi^{t_2}, \pi^{t_3})$: An adversary runs this query to get all the messages transmitted between the two valid entities. The execution of this query is also known as an eavesdropping attack.
  2. $RVL(\pi^t)$: An adversary execute this query to reveal the session-key generated by an instance π^t (and its partner) in an on-going session.
  3. $SND(\pi^t, message)$: It is modeled as an active attack where an adversary $\mathcal{A}$ sends a message to the participating instance π^t and receives a response message from π^t.
  4. $CPTIN(\pi^t_{IN})$: It is modeled as an intermediate node capture attack where all the secret parameters of an intermediate node are revealed to an adversary $\mathcal{A}$ by executing this query.
  5. $CPTSN(\pi^t_{SN})$: It is modeled as a sensor node capture attack where all the secret parameters stored in a sensor node are revealed to an adversary $\mathcal{A}$ by executing this query.
  6. $TST(\pi^t)$: This query is modeled to examine the semantic security of the session-key between SN and HN following the indistinguishability in the ROR model. Under this query, before the starting of the experiment an unbiased coin c is tossed and the result is known only to $\mathcal{A}$. This result determines the output of the test query. If an adversary $\mathcal{A}$ executes the $TST(\pi^t)$ query and also the session-key (SK) is fresh, π^t returns SK if $c = 1$ or returns a random number if $c = 0$, it returns a null value ($\perp$) otherwise.

It is to be noted that an adversary $\mathcal{A}$ can access any number of $TST(\pi^t)$ queries however, only a limited number of $CPTSN$ (π^t_{SN}) query can be acquired by $\mathcal{A}$.

Semantic security of the session-key: The ROR model desires that an adversary $\mathcal{A}$ must be able to distinguish between the real session-key of an instance with the random key. $\mathcal{A}$ can execute several $TST(\pi^t)$ queries to either $\pi^{t_1}_{SN}$ or $\pi^{t_3}_{HN}$ and the output must be consistent or uniform to random bit c. After the completion of the game, $\mathcal{A}$ returns a guessed bit c' and wins if $c' = c$ is achieved. Let $Succ$ denotes the event for $\mathcal{A}$ to win a game, the advantage Adv^{MAKA} of an adversary $\mathcal{A}$ to break the semantic security of our protocol mutual authentication and key agreement ($MAKA$) scheme is defined as $Adv^{MAKA}_t = |2.Pr(Succ) - 1|$. Our protocol $MAKA$ is secure if $Adv^{MAKA}_t \leq \epsilon$, for the run time t and sufficiently small $\epsilon > 0$.

Random oracle: The one-way cryptographic hash function $h(.)$ is modeled as the random oracle say H, to have access to all the communicating entities involved in a network including an adversary $\mathcal{A}$.

Theorem. *Suppose $\mathcal{A}$ be an adversary running in a polynomial time t against our mutual authentication and key agreement scheme (MAKA) in the random oracle model and $q_h, |Hash|, q_{send}, |PD|$ and $Adv^{MAKA}_A(t)$ denote the number of hash queries, the range space of h(.), the number of send queries, the size of uniformly distributed password dictionary and $\mathcal{A}$'s advantage in breaking the $MAKA$ secure symmetric cypher in time t respectively. Then $\mathcal{A}$'s advantage for deriving the session-key SK between SN and HN is estimated as:*

$$Adv^{MAKA}_A(t) \leq \frac{q_h^2}{|Hash|} + \frac{q_{send}}{|PD|}$$

Proof. The proof followed here is similar to the proof shown in [27,40,41]. In this proof, we define a sequence of four games G_i, where (i = 0,1,2,3). Let $Succ_i$ be an event for $\mathcal{A}$ to guess the bit c correctly in a game G_i, the advantage of winning the game G_i by $\mathcal{A}$ is represented as $Adv^{G_i}_A = Pr[Succ_i]$. Given below are the detailed description of each game G_i:

- G_0: G_0 is an actual attack in the ROR model performed by $\mathcal{A}$ against our proposed protocol $MAKA$ in which $\mathcal{A}$ selects a bit c prior to the beginning of the game G_0. Therefore by definition,

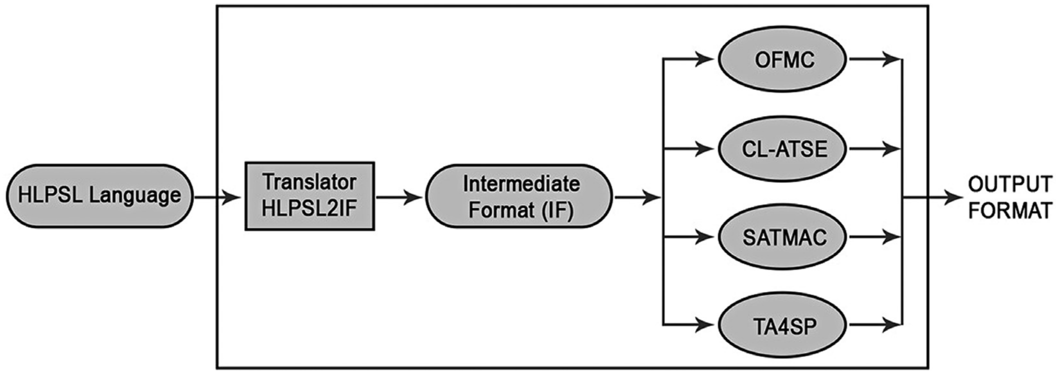

Fig. 7. AVISPA tool architecture.

the following result is obtained:

$$Adv_{A}^{MAKA}(t) = |2.Pr[Succ_0] - 1| \tag{11}$$

- G_1: G_1 is an eavesdropping attack performed by $\mathcal{A}$ by executing the $EXE(\pi^{t_1}, \pi^{t_2}, \pi^{t_3})$ query to intercept the transmitted messages $\langle tid_n, y_n, a_n, b_n, V_n, t_1\rangle, \langle tid_n, y_n, a_n, b_n, V_i, t_1, t_2\rangle, \langle \alpha, \beta, \eta, \mu, V_h, t_3\rangle$ and $\langle \alpha, \beta, \eta, V_i, t_4\rangle$ during the authentication and key agreement phase of the proposed scheme and then executing $TST(\pi^t)$ query. The output of $TST(\pi^t)$ query is examined whether the session-key between SN and HN is real key or a random value. In our protocol, the session-key is computed as $SK = h(id_n \parallel r_n \parallel f_n \parallel c_n)$ and the intercepted messages do not reveal the secret parameters f_n, c_n, r_n and id_n. Thus, $\mathcal{A}$'s probability of winning the game G_1 by eavesdropping attack is not increased. Hence, it gives:

$$Pr[Succ_1] = Pr[Succ_0] \tag{12}$$

- G_2: G_2 is an active attack performed by $\mathcal{A}$ by simulating send and hash queries in order to deceive a legitimate node into accepting an illegal message. $\mathcal{A}$ can make any number of hash queries (q_h) for creating hash collisions however all the messages contain the current timestamps and random number and it is not feasible in a polynomial time for hash collision occurrence by executing send and hash queries. Therefore, using birthday paradox, the following result is obtained:

$$|Pr[Succ_2] - Pr[Succ_1]| \leq \frac{q_h^2}{2.|Hash|} \tag{13}$$

- G_3: Under this game, $\mathcal{A}$ performs a node capture attack by executing $CPTN(\pi^t)$ query and extracts all the secret parameters stored in it. It is categorized into two parts, (i) where $\mathcal{A}$ captures an intermediate node and uses all information of it. Intermediate node does not store any parameters of either SN or HN, therefore no new information is gained as from G_2, (ii) where $\mathcal{A}$ captures a sensor node and uses all information of it. The secret key K_{HN} of HN is not stored in SN and secret key K_n is encrypted using one-way hash function $h(.)$. $\mathcal{A}$ tries to guess K_n using password dictionary attack from $a_n = id_n \oplus h(K_{hn} \parallel K_n)$ and $b_n = K_{hn} \oplus K_n \oplus a_n$. It is difficult for $\mathcal{A}$ to apply PD attack due to one-way collision resistance hash function and it becomes infeasible to guess K_{hn} of HN. Hence, we have the following result:

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{q_{send}}{2.|PD|} \tag{14}$$

All the oracle queries are executed by $\mathcal{A}$ to break the semantic security of our protocol $MAKA$, $\mathcal{A}$ can only guess the bit c at last for winning the game after $TST(\pi^t)$ query. It gives $|Pr[Succ_3]| = \frac{1}{2}$.

By using Eqs. (11) and (12), we get the following result for the game G_i:

$$\begin{aligned} Adv_{A}^{MAKA}(t) &= |2.Pr[Succ_0] - 1| \\ &= |2.Pr[Succ_1] - 1| \\ &= 2.|Pr[Succ_1] - \frac{1}{2}| \\ &= 2.|Pr[Succ_1] - Pr[Succ_3]| \end{aligned} \tag{15}$$

By triangular inequality, we have

$$\begin{aligned} |Pr[Succ_1] - Pr[Succ_3]| \leq{} & |Pr[Succ_1] - Pr[Succ_2]| \\ & + |Pr[Succ_2] - Pr[Succ_3]| \end{aligned}$$

Using Eqs. (13)–(15), we get

$$\frac{1}{2} Adv_{A}^{MAKA}(t) \leq \frac{q^2}{2.|Hash|} + \frac{q_{send}}{2.|PD|}$$

Multiplying both sides by 2, we get the result

$$Adv_{A}^{MAKA}(t) \leq \frac{q^2}{|Hash|} + \frac{q_{send}}{|PD|}$$

Hence, our proposed mutual authentication and key agreement $(MAKA)$ scheme is secure for the larger size of password dictionary and range space of hash function.

7.3. Formal security verification based on AVISPA tool

In this section, we simulate our proposed scheme using widely-used AVISPA simulation tool [42,43]. AVISPA is a push button formal verification tool which uses modular and expressive High-Level Protocol Specification Language (HLPSL) [44] for code implementation to identify security vulnerabilities in a protocol. AVISPA integrates four back-ends namely, (i) *On-the-fly-Model-Checker (OFMC)* [45], (ii) *Constraint-Logic-based Attack Searcher (CL-AtSe)* [46], (iii) *SAT-based Model-checker (SATMC)* [47], and (iv) *Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)*, with HLPSL to analyze the security protocols. The architecture of the AVISPA tool is shown in Fig. 7 where HLPSL language is converted into an intermediary form (IF) with the help of HLPSL2IF translator. This intermediate form is provided to AVISPA tool's back-ends for security check and the output shows whether the protocol is Safe or Unsafe for practical implementation.

The output format contains the following major fields:

- *SUMMARY* indicating if the test protocol is SAFE or UNSAFE, or that analysis found to be INCONCLUSIVE.
- *DETAILS* depicting conditions in which test protocol is proclaimed to be safe or attack findings condition or lastly why the inspection were inconclusive.
- *PROTOCOL* depicting the name of the protocol.
- *GOAL* indicating the goal of the analysis.

```
role admin(SA,SN,IN,HN :agent,SND,RCV :channel(dy),
      SKsasn,SKsain,SKsahn :symmetric_key,
      KHN,KN,IDN,XNIN :text,H :hash_func)
played_by SA
def=
local State :nat,
      AN,BN,CN,IDIN :text
const secKHN,secIDN,secKN,secXNIN,secIDIN,
        sn_in_gi,sn_in_beta,in_sn_gn,in_hn_li,
        in_hn_gn,in_hn_tidn :protocol_id
init State:=0
transition
1. State=0/\RCV(start)=|>State':=1/\IDIN':=new()
   /\AN':=xor(IDN,H(KHN.KN))
   /\BN':=xor(xor(KHN,AN'),KN)
   /\CN':=H(KHN,IDN)
   /\SND({IDN.XNIN.AN'.BN'.CN'}_SKsasn)
   /\SND({IDIN'.XNIN}_SKsain)
   /\SND({IDIN'}_SKsahn)
   /\secret(IDIN, secIDIN,{IN,HN})
end role
```

Fig. 8. HLPSL code for System Administrator.

- *BACKEND* depicting the name of the back-end utilized.
- *STATISTICS* depicting the parse-time, search-time, visited nodes and the depth of the nodes analyzed by the back-end in executing the protocol.

7.3.1. Implementation and analysis of results

The security of our proposed protocol is evaluated by the AVISPA tool using HLPSL specification. The implementation of the proposed protocol involved four types of roles — *role admin* for the system administrator as depicted in Fig. 8, *role snode* for the wearable sensor nodes as depicted in Fig. 9, *role inode* for the intermediate node as depicted in Fig. 10, and *role hnode* for the hub node as depicted in Fig. 11.

The proposed protocol is simulated using the SPAN (Security ANimator for AVISPA) simulation tool in Ubuntu 10.10 (32-bit) Operating System having 4096 MB of RAM. The output of *OFMC* back-end and *CL-AtSe* back-end are shown in Figs. 12 and 13 respectively. *SATMC* and *TA4SP* back-ends both do not support bitwise XOR operation at present and therefore these back-ends show inconclusive results. Hence, these are not included in the paper. The output of the other two back-ends clearly show that our proposed protocol is safe from various known attacks like replay, impersonation and man-in-the-middle attacks based on Dolev–Yao threat model and the secrecy of the session-key is satisfied. Hence, our scheme can be used for practical implementation.

7.4. Informal security analysis

In this section, we discuss the detailed security analysis of the proposed scheme against well-known attacks and vulnerabilities. The proposed scheme withstands anonymity, mutual authentication and key agreement, perfect forward/backward secrecy, and also resilient to eavesdropping, impersonation, replay, intermediate node capture and man-in-the-middle attacks.

```
role snode(SA,SN,IN,HN :agent,SND,RCV :channel(dy),
   SKsasn :symmetric_key,H :hash_func)
played_by SN
def=
local State :nat,
   IDN, IDIN, XNIN, AN, BN, CN, RN, T1, XN, YN, GN,
   TIDN, KN, KHN, GI, T4, FN, Alpha, Beta, Eta, MU,
   ANnew, BNnew, KS :text
const secKHN, secIDN, secKN, secXNIN, secIDIN,
   sn_in_gi, sn_in_beta, in_sn_gn, in_hn_li, in_hn_gn,
   in_hn_tidn :protocol_id
init State:=0
transition
1. State=0 /\ RCV({IDN'.XNIN'.AN'.BN'}_SKsasn)=|> State':=1
   /\ RN':= new() /\ T1':= new() /\ XN':= xor(AN',IDN')
   /\ YN':= xor(XN',RN') /\ GN':= H(XNIN'.T1')
   /\ TIDN':= H(xor(IDN',T1').RN') /\ secret(KN,secKN,{SA,IN,HN})
   /\ secret(KHN,secKHN,{SA,IN,HN})
   /\ secret(IDN,secIDN,{SA,IN,HN})
   /\ secret(XNIN,secXNIN,{SA,IN,HN})
   /\witness(SN,IN,in_sn_gn,GN') /\ SND(TIDN'.YN'.GN'.AN'.BN'.T1')
2. State=1 /\ RCV(Alpha'.Beta'.Eta'.MU'.GI'.T4')=|> State':=2
   /\ GI':= H(XNIN.T4') /\ request(SN,IN,sn_in_gi,GI)
   /\ FN':= xor(CN,Alpha') /\ Beta':= H(XN.RN.FN'.Eta'.MU')
   /\ request(SN,IN,sn_in_beta,Beta) /\ ANnew':= xor(H(CN.FN'),Eta')
   /\ BNnew':= xor(H(CN.FN'),MU') /\ KS':= H(IDN.RN.FN'.CN)
   /\ AN':= ANnew' /\ BN':= BNnew' /\ secret(KN,secKN,{SA,IN,HN})
   /\ secret(KHN,secKHN,{SA,IN,HN}) /\secret(IDN,secIDN,{SA,IN,HN})
   /\ secret(XNIN,secXNIN,{SA,IN,HN})
end role
```

Fig. 9. HLPSL code for Sensor Node.

```
role inode(SA,SN,IN,HN:agent,SND,RCV:channel(dy),
   SKsain:symmetric_key,H:hash_func)
played_by IN
def=
local State :nat,
   IDN, IDIN, XNIN, AN, BN, CN, T1, GN, TIDN, KN, KHN,
   T3, T2, T4, Alpha, Beta, Eta, MU, JI, LI, GI, YN :text
const secKHN, secIDN, secKN, secXNIN, secIDIN,
   sn_in_gi, sn_in_beta, in_sn_gn, in_hn_li, in_hn_gn,
   in_hn_tidn :protocol_id
init State:=0
transition
1. State=0 /\ RCV({IDIN'.XNIN}_SKsain)=|> State':=1
     /\ secret(KN,secKN,{SA,IN,HN}) /\ secret(KHN,secKHN,{SA,IN,HN})
     /\ secret(IDN,secIDN,{SA,IN,HN}) /\ secret(XNIN,secXNIN,{SA,IN,HN})
     /\secret(IDIN,secIDIN,{IN,HN})
2. State=1 /\ RCV(TIDN'.YN'.GN'.AN'.BN'.T1')=|> State':=2
     /\ GN':= H(XNIN.T1') /\ request(IN,SN,in_sn_gn,GN)
     /\ T2':= new() /\ JI':= H(IDIN.T2') /\ witness(IN,HN,in_hn_gn,JI')
     /\ witness(IN,HN,in_hn_tidn,TIDN')
     /\ SND(TIDN'.YN'.AN'.BN'.JI'.T1'.T2')
3. State=2 /\ RCV(Alpha'.Beta'.Eta'.MU'.LI'.T3')=|> State':=3
     /\ LI':= H(IDIN.T3') /\ request(IN,HN,in_hn_li,LI) /\T4':= new()
     /\ GI':= H(XNIN.T4') /\ witness(IN,SN,sn_in_gi,GI')
     /\ witness(IN,SN,sn_in_beta,Beta')
     /\ SND(Alpha'.Beta'.Eta'.MU'.GI'.T4')/\secret(KN,secKN,{SA,IN,HN})
     /\ secret(KHN,secKHN,{SA,IN,HN}) /\secret(IDN,secIDN,{SA,IN,HN})
     /\ secret(XNIN,secXNIN,{SA,IN,HN}) /\ secret(IDIN,secIDIN,{IN,HN})
end role
```

Fig. 10. HLPSL code for Intermediate Node.

```
role hnode(SA,SN,IN,HN:agent,SND,RCV:channel(dy),
   SKsahn :symmetric_key,H :hash_func)
played_by HN
def=
local State :nat,
   IDN,IDIN,XNIN,AN,BN,CN,T1,T2,TIDN,KHN,T3,Alpha,
   Beta,Eta,MU,JI,KN,RN,FN,YN,XN,KS,LI,TKN :text
const secKHN,secIDN,secKN,secXNIN,secIDIN,sn_in_gi,sn_in_beta,
   in_sn_gn,in_hn_li,in_hn_gn,in_hn_tidn :protocol_id
init State:=0
transition
1. State=0/\RCV({IDIN'}_SKsahn)=|>State':=1
  /\secret(KN,secKN,{SA,IN,HN})/\secret(KHN, secKHN,{SA,IN,HN})
  /\secret(IDN, secIDN,{SA,IN,HN})/\secret(XNIN,secXNIN,{SA,IN,HN})
  /\secret(IDIN,secIDIN,{IN,HN})
2. State=1/\RCV(TIDN'.YN'.AN'.BN'.JI'.T1'.T2')=|>State':=2
  /\JI':=H(IDIN.T2')/\request(HN,IN,in_hn_gn,JI)
  /\KN':=xor(xor(KHN,BN'),AN')/\XN':=H(KHN.KN')
  /\IDN':=xor(AN',XN')/\RN':=xor(YN',XN')
  /\TIDN':=H(xor(IDN',T1').RN')/\request(HN,IN,in_hn_tidn,TIDN)
  /\FN':=new()/\TKN':=new()/\AN':=xor(IDN',H(KHN.TKN'))
  /\BN':=xor(xor(KHN,TKN'),AN')/\Alpha':=xor(FN',H(KHN.IDN'))
  /\Eta':=xor(H(H(KHN.IDN').FN'),AN')/\MU':=xor(H(H(KHN.IDN').FN'),BN')
  /\Beta':=H(XN'.RN'.FN'.Eta'.MU')/\KS':=H(IDN'.RN'.FN'.H(KHN.IDN'))
  /\T3':=new()/\LI':=H(IDIN.T3')/\witness(HN,IN,in_hn_li,LI')
  /\SND(Alpha'.Beta'.Eta'.MU'.LI'.T3')/\secret(KN,secKN,{SA,IN,HN})
  /\secret(KHN,secKHN,{SA,IN,HN})/\secret(IDN,secIDN,{SA,IN,HN})
  /\secret(XNIN,secXNIN,{SA,IN,HN})/\secret(IDIN,secIDIN,{IN,HN})
end role
```

Fig. 11. HLPSL code for Hub Node.

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/proto.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.52s
 visitedNodes: 26 nodes
 depth: 2 plies
```

Fig. 12. OFMC back-end result.

7.4.1. Eavesdropping attack

In Dolev–Yao threat model, an adversary $\mathcal{A}$ is able to record all the messages between a sensor node and the hub node transmitted over an unsecure channel during the authentication phase. $\mathcal{A}$ now knows the parameters $(tid_n, a_n, b_n, y_n, \alpha, \beta, \eta, \mu)$, still it is not possible to compute session key K_s as $\mathcal{A}$ does not know id_n, r_n and c_n of a sensor node as these are not transmitted directly and id_n is protected by the property of one-way hash function $h(.)$. $\mathcal{A}$ also does not know f_n selected by the hub node randomly to compute session key. Hence, the privacy of the session key is intact and therefore, the scheme is secure against this attack.

```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL
 /home/span/span/testsuite/results/proto.if
GOAL
 As Specified
BACKEND
 CL-AtSe
STATISTICS
 Analysed   : 2 states
 Reachable  : 2 states
 Translation: 0.04 seconds
 Computation: 0.02 seconds
```

Fig. 13. CL-AtSe back-end result.

7.4.2. Replay attack

In this attack, an adversary $\mathcal{A}$ tries to replay the eavesdropped message from the previous session to gain the access to the network. However, all the messages transmitted are bound to current timestamps and entities check transmission delay time before processing. Therefore, $\mathcal{A}$ cannot replay the previously transmitted messages and the protocol is safe against replay attacks.

7.4.3. Sensor node capture attack

Suppose an adversary $\mathcal{A}$ is able to capture one of the sensor nodes and extracts all the parameters stored in it through power analysis attack [48]. $\mathcal{A}$ would still not be able to get hub node's master key K_{hn} as $id_n \oplus a_n = h(K_{hn} \parallel K_n)$ and $a_n \oplus b_n = K_{hn} \oplus K_n$ both does not reveal K_{hn} alone. To extract this, $\mathcal{A}$ has to perform brute force attack on $K_{hn} \oplus K_n$. Hence, capturing of any sensor node does not affect the other sensor nodes and the scheme still operates securely.

7.4.4. Intermediate node impersonation attack

Suppose an adversary $\mathcal{A}$ tries to create a forge packet $\langle \alpha, \beta, \eta, \mu, V_i, t_4 \rangle$ at the fourth step of the authentication phase and send it to the sensor node. Sensor node first calculates the parameter $V_i^* = h(X_{n-in} \parallel t_4)$ and verifies it with V_i. Since, an adversary $\mathcal{A}$ does not know the secret shared key X_{n-in} between the sensor node and the intermediate node, it will not be able to create a valid request and sensor node drops the forge packet. Hence, the proposed protocol is safe against an intermediate node impersonation attack.

7.4.5. Intermediate node capture attack

Suppose an adversary $\mathcal{A}$ captures the intermediate node and extracts its secret parameters id_{in} and X_{n-in}. It is clearly seen that both the parameters does not reveal any information about the sensor nodes and the hub node such as id_n, c_n or x_n of the sensor node and K_{hn} of the hub node. Therefore, we do not need to discard all the parameters and run the registration phase again for all the sensor nodes while replacing an intermediate node with the new one. Hence, the protocol is still secure and performs correctly.

7.4.6. Sensor node impersonation attack

Suppose an adversary $\mathcal{A}$ intercepts the sensor node's message $\langle tid_n, y_n, a_n, b_n, V_n, t_1 \rangle$ during the execution of our protocol and tries to create a valid message, $\mathcal{A}$ requires id_n of a sensor node to compute $x_n = a_n \oplus id_n$ and $y_n = x_n \oplus r_n$. $\mathcal{A}$ cannot acquire id_n of a sensor node

Table 2
Comparison of security and functionality features.

Security properties	[16]	[32]	[31]	[33]	[34]	Proposed
Anonymity and untraceability	✗	✓	✗	✓	✓	✓
Perfect forward secrecy	✓	✓	✗	✓	✓	✓
Replay attack	✓	✓	✓	✓	✓	✓
Node capture attack	✗	✗	✗	✓	✗	✓
Sensor node impersonation attack	✗	✓	✓	✗	✓	✓
Intermediate node impersonation attack	✗	✗	–	✗	✓	✓
Offline guessing attack	✗	✓	✗	✓	✓	✓
Privileged insider attack	✓	✗	✓	✓	✓	✓
Man-in-the-middle attack	✓	✓	✓	✗	✓	✓

even after capturing the intermediate node as it is not sent directly over the network and also not stored in the intermediate node. Therefore an adversary $\mathcal{A}$ cannot impersonate as an authentic sensor node. Therefore, our protocol is resilient to sensor node impersonation with intermediate node capture attack.

7.4.7. Hub node impersonation attack

An adversary tries to impersonate as a valid hub node by sending a message $\langle \alpha, \beta, \eta, \mu \rangle$ to the sensor node via an intermediate node. To compute α, an adversary $\mathcal{A}$ must know f_n *and* K_{hn} of the hub node to calculate as $\alpha = h(H_{hn} \parallel id_n) \oplus f_n$. Even after capturing the intermediate node, $\mathcal{A}$ does not know the secret identity id_n of a sensor node and hence it is not possible to perform this attack. Therefore, our protocol is also resilient to hub node impersonation with intermediate node capture attack.

7.4.8. Perfect forward/backward secrecy

Suppose an adversary $\mathcal{A}$ had compromised the session key K_s, it must not affect the privacy of any past or future sessions. An adversary $\mathcal{A}$ will not be able to compute id_n, r_n, f_n, c_n or x_n from the session key due to the protection of one-way hash function. Also r_n and f_n are chosen randomly and a_n^+ and b_n^+ are updated each time. Therefore, our scheme satisfies the perfect forward/backward secrecy features.

7.4.9. Anonymity and untraceability

An adversary $\mathcal{A}$ must not be able to find out the real identity id_n of a sensor node and also not trace back to any sensor node by eavesdropping any previous messages communicated in the network. In our protocol, the message $\langle tid_n, y_n, a_n, b_n \rangle$ does not reveal id_n as $tid_n = h((id_n \oplus t_1) \parallel r_n)$ is protected with hash property and r_n is chosen randomly every time. Therefore, no two sessions are linkable and an adversary $\mathcal{A}$ cannot identify any node using temporary identities. Hence, our scheme provides anonymity and untraceability features.

8. Performance comparison

In this section, we discuss the performance comparison of our protocol with Li et al.'s protocol and the other related existing schemes, designed for the similar environment as of ours, based on the functionality features, storage requirements, computational and communication overheads in the authentication and key-agreement phase. The following subsection discusses each features separately.

8.1. Functionality features

In Table 2, the detailed comparison of security and functionality features of the proposed protocol with Li et al.'s protocol and the other related schemes is shown. This evaluation shows the effectiveness of our protocol as compared with the other schemes. Our protocol satisfies all the essential security features and resist well-known attacks and hence, well-suited for real life applications.

Table 3
Storage cost comparison.

Schemes	SN	IN	HN	Total (bits)
[16]	640	768	288	1696
[32]	640	640	160	1440
[31]	896	–	160	1056
[33]	640	768	1024	2432
[34]	544	544	736	1824
Proposed	1056	288	288	1632

8.2. Storage requirements

Table 3 shows the storage requirements of our scheme as well as the other related schemes for different nodes. The sensor node stores the parameters $\langle id_n, X_{n-in}, a_n, b_n, c_n \rangle$. The intermediate node stores the parameters id_{in} and X_{n-in}. On the other hand, the hub node stores K_{hn} and id_{in}. In our scheme, we are using SHA3-256 hash function which generates the hash output of 256 bits. Also, the secret keys and identities chosen by the system administrator are 160 bits and 128 bits respectively. Therefore, total storage required by all the nodes are compiled in Table 3. It may be noted that the storage requirements, at intermediate and hub nodes, are shown with respective to one sensor node. It helps in comparing the storage cost with the other related protocols in its simplest form as the number of nodes vary in different networks. To compare the communication overheads with the other schemes, we assumed the same property of Hash digest and computed their overheads.

8.3. Computation cost

In Table 4, we summarize the comparison of computational cost of the proposed protocol with Li et al.'s protocol and the other related schemes for the authentication and key-agreement phase only. Here, we have chosen the selective identical aspects as the composition of other protocols are different from our protocol. The hash operation is denoted by T_h and the XOR operation is denoted by T_X for the time needed by these operations. Li et al. [16], Chen et al. [31] and Kompara et al. [33] schemes do not perform any computation on intermediate nodes and only forwards the packet to the hub node. Hence, the intermediate node's computation field is left blank in Table 4 for these schemes.

8.4. Communication overhead

Table 5 provides the summary of the communication cost analysis. Assuming the size of the timestamp to be 32 bits, in our proposed scheme the sensor node sends the message $\langle tid_n, y_n, a_n, b_n, V_n, t_1 \rangle$ to the intermediate node which is 5*256 + 32 = 1312 bits long. The intermediate node forwards the message $\langle tid_n, y_n, a_n, b_n, V_i, t_1, t_2 \rangle$ to the hub node which is 5*256 + 32 + 32 = 1344 bits long. The hub node sends the message $\langle \alpha, \beta, \eta, \mu, V_h, t_3 \rangle$ back to intermediate node which is 5*256 + 32 = 1312 bits and finally, the intermediate node forwards the message $\langle \alpha, \beta, \eta, \mu, V_i, t_4 \rangle$ to sensor node which is 5*256 + 32 = 1312 bits long. Therefore, total number of bits sent over the network in our proposed scheme is 5280 bits.

9. Conclusion

WBAN plays an important role in remotely monitoring of patient's vital information in the healthcare scenario. The authentication process gathers preeminent attention in the field of, but not limited to, medical IoT where the security and privacy of a user are of dominant interest. Several authentication and key agreement protocols have been proposed in the literature based on WBAN but no one completely protects from all security threats. This paper primarily reviewed Li et al.'s anonymous mutual authentication and key-agreement protocol

Table 4
Computation cost comparison.

Nodes	[16]	[32]	[31]	[33]	[34]	Proposed
SN	$3T_h + 7T_X$	$3T_h + 5T_X$	$5T_h + 5T_X$	$3T_h + 6T_X$	$4T_h + 4T_X$	$7T_h + 6T_X$
IN	–	$3T_h + 5T_X$	–	–	$7T_h + 4T_X$	$4T_h + 0T_X$
HN	$5T_h + 12T_X$	$10T_h + 20T_X$	$8T_h + 11T_X$	$5T_h + 8T_X$	$5T_h + 3T_X$	$10T_h + 11T_X$
Total	$8T_h + 19T_X$	$16T_h + 30T_X$	$13T_h + 16T_X$	$8T_h + 14T_X$	$16T_h + 11T_X$	$21T_h + 17T_X$

Table 5
Communication cost comparison.

Schemes	No. of msgs.	No. of bits
Li et al. [16]	4	4416
Koya et al. [32]	6	5472
Chen et al. [31]	2	2080
Kompara et al. [33]	4	3456
Gupta et al. [34]	5	3808
Proposed	4	5280

for WBAN and then presented various security vulnerabilities associated with it. To fix their security drawbacks, we designed a provably secure and efficient anonymous mutual authentication and key agreement protocol using simple hash and XOR operations. We showed that our proposed protocol overcomes the security vulnerabilities of Li et al.'s protocol using formal verification ROR model, BAN-Logic and the widely accepted AVISPA security tool. Furthermore, informal security cryptanalysis proved the resilience of our protocol against relevant known security attacks. We also compared our protocol with the related existing schemes in terms of computational and communication capabilities and proved that the proposed protocol is relatively better than the other existing schemes. Hence, our proposed approach is suitable for IoT applications.

CRediT authorship contribution statement

Ankur Gupta: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft, Writing - review & editing, Visualization. **Meenakshi Tripathi:** Supervision. **Aakar Sharma:** Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2347–2376, http://dx.doi.org/10.1109/COMST.2015.2444095.

[2] A. Kumar, A.A. Rahmani Hosseinabadi, M. Shareh, A. Zolfagharian, N. Chilamkurti, S. Rad, Iot resource allocation and optimization based on heuristic algorithm, Sensors (2020) 539.

[3] A.K. Sangaiah, M. Sadeghilalimi, A.A.R. Hosseinabadi, W. Zhang, Energy consumption in point-coverage wireless sensor networks via bat algorithm, IEEE Access 7 (2019) 180258–180269.

[4] A.K. Sangaiah, D.V. Medhane, T. Han, M.S. Hossain, G. Muhammad, Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics, IEEE Trans. Ind. Inf. 15 (7) (2019) 4189–4196.

[5] A.K. Sangaiah, D.V. Medhane, G. Bian, A. Ghoneim, M. Alrashoud, M.S. Hossain, Energy-aware green adversary model for cyberphysical security in industrial system, IEEE Trans. Ind. Inf. 16 (5) (2020) 3322–3329.

[6] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, A. Jamalipour, Wireless body area networks: A survey, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1658–1686, http://dx.doi.org/10.1109/SURV.2013.121313.00064.

[7] R. Cavallari, F. Martelli, R. Rosini, C. Buratti, R. Verdone, A survey on wireless body area networks: Technologies and design challenges, IEEE Commun. Surv. Tutor. 16 (3) (2014) 1635–1657.

[8] S.M.R. Islam, D. Kwak, M.H. Kabir, M. Hossain, K. Kwak, The internet of things for health care: A comprehensive survey, IEEE Access 3 (2015) 678–708, URL 10.1109/ACCESS.2015.2437951.

[9] D. Tse, P. Viswanath, Fundamentals of Wireless Communication, Cambridge university press, 2005.

[10] P.J. Soh, G.A.E. Vandenbosch, M. Mercuri, D.M.M. Schreurs, Wearable wireless health monitoring: Current developments, challenges, and future trends, IEEE Microw. Mag. 16 (4) (2015) 55–70, URL 10.1109/MMM.2015.2394021.

[11] S. Seneviratne, Y. Hu, T. Nguyen, G. Lan, S. Khalifa, K. Thilakarathna, M. Hassan, A. Seneviratne, A survey of wearable devices and challenges, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2573–2620, http://dx.doi.org/10.1109/COMST.2017.2731979.

[12] J. Daemen, V. Rijmen, The Design of Rijndael: AES-The Advanced Encryption Standard, Springer Science & Business Media, 2013.

[13] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126, http://dx.doi.org/10.1145/359340.359342, URL http://doi.acm.org/10.1145/359340.359342.

[14] W. Diffie, M. Hellman, New directions in cryptography, IEEE Trans. Inform. Theory 22 (6) (1976) 644–654, http://dx.doi.org/10.1109/TIT.1976.1055638.

[15] N.R. Potlapally, S. Ravi, A. Raghunathan, N.K. Jha, A study of the energy consumption characteristics of cryptographic algorithms and security protocols, IEEE Trans. Mob. Comput. 5 (2) (2006) 128–143, http://dx.doi.org/10.1109/TMC.2006.16.

[16] X. Li, M.H. Ibrahim, S. Kumari, A.K. Sangaiah, V. Gupta, K.-K.R. Choo, Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks, Comput. Netw. 129 (P2) (2017) 429–443, https://doi.org/10.1016/j.comnet.2017.03.013.

[17] M.A. Ferrag, L.A. Maglaras, H. Janicke, J. Jiang, Authentication protocols for internet of things: A comprehensive survey, 2016, ArXiv e-prints.

[18] K.H.M. Wong, Y. Zheng, J. Cao, S. Wang, A dynamic user authentication scheme for wireless sensor networks, in: IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06), Vol. 1, 2006, p. 8, URL 10.1109/SUTC.2006.1636182.

[19] M.L. Das, Two-factor user authentication in wireless sensor networks, IEEE Trans. Wireless Commun. 8 (3) (2009) 1086–1090, URL 10.1109/TWC.2008.080128.

[20] M.K. Khan, K. Alghathbar, Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks', Sensors 10 (3) (2010) 2450–2459, URL 10.3390/s100302450; http://www.mdpi.com/1424-8220/10/3/2450.

[21] B. Vaidya, D. Makrakis, H.T. Mouftah, Improved two-factor user authentication in wireless sensor networks, in: 2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, 2010, pp. 600–606, URL 10.1109/WIMOB.2010.5645004.

[22] 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. Netw. Comput. Appl. 34 (1) (2011) 73–79, https://doi.org/10.1016/j.jnca.2010.09.003.

[23] X. Li, J. Niu, M.K. Khan, J. Liao, An enhanced smart card based remote user password authentication scheme, J. Netw. Comput. Appl. 36 (5) (2013) 1365–1371, https://doi.org/10.1016/j.jnca.2013.02.034.

[24] 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. Modelling 58 (1) (2013) 85–95, Financial IT and Security and 2010 International Symposium on Computational Electronics. https://doi.org/10.1016/j.mcm.2012.06.033.

[25] T. Muhamed, B. Boštjan, H. Marko, A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion, Ad Hoc Netw. 20 (2014) 96–112, https://doi.org/10.1016/j.adhoc.2014.03.009.

[26] S.F. Mohammad, T. Muhamed, K. Saru, H. Marko, An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the internet of things environment, Ad Hoc Netw. 36 (2016) 152–176, URL https://doi.org/10.1016/j.adhoc.2015.05.014.

[27] C. Chang, H. Le, A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks, IEEE Trans. Wireless Commun. 15 (1) (2016) 357–366, URL 10.1109/TWC.2015.2473165.

[28] R. Amin, G. Biswas, A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks, Ad Hoc Netw. 36 (2016) 58–80, URL https://doi.org/10.1016/j.adhoc.2015.05.020.

[29] P. Gope, T. Hwang, A realistic lightweight anonymous authentication protocol for securing real-time application data access in wireless sensor networks, IEEE Trans. Ind. Electron. 63 (11) (2016) 7124–7132, URL 10.1109/TIE.2016.2585081.

[30] A. Adavoudi-Jolfaei, M. Ashouri-Talouki, S.F. Aghili, Lightweight and anonymous three-factor authentication and access control scheme for real-time applications in wireless sensor networks, Peer-to-Peer Netw. Appl. (2017) http://dx.doi.org/10.1007/s12083-017-0627-8.

[31] C.-M. Chen, B. Xiang, T.-Y. Wu, K.-H. Wang, An anonymous mutual authenticated key agreement scheme for wearable sensors in wireless body area networks, Appl. Sci. 8 (7) (2018) 1074.

[32] A.M. Koya, D. P. P., Anonymous hybrid mutual authentication and key agreement scheme for wireless body area network, Comput. Netw. 140 (2018) 138–151, http://dx.doi.org/10.1016/j.comnet.2018.05.006.

[33] M. Kompara, S.H. Islam, M. Hlbl, A robust and efficient mutual authentication and key agreement scheme with untraceability for WBANs, Comput. Netw. 148 (2019) 196–213, https://doi.org/10.1016/j.comnet.2018.11.016.

[34] A. Gupta, M. Tripathi, T.J. Shaikh, A. Sharma, A lightweight anonymous user authentication and key establishment scheme for wearable devices, Comput. Netw. 149 (2019) 29–42, https://doi.org/10.1016/j.comnet.2018.11.021.

[35] D. Dolev, A. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208, http://dx.doi.org/10.1109/TIT.1983.1056650.

[36] M. Burrows, M. Abadi, R. Needham, A logic of authentication, ACM Trans. Comput. Syst. 8 (1) (1990) 18–36, http://dx.doi.org/10.1145/77648.77649, URL http://doi.acm.org/10.1145/77648.77649.

[37] M. Bellare, P. Rogaway, Random oracles are practical: A paradigm for designing efficient protocols, in: Proceedings of the 1st ACM Conference on Computer and Communications Security, in: CCS '93, ACM, New York, NY, USA, 1993, pp. 62–73, http://dx.doi.org/10.1145/168588.168596, URL http://doi.acm.org/10.1145/168588.168596.

[38] S. Paliwal, Hash-based conditional privacy preserving authentication and key exchange protocol suitable for industrial internet of things, IEEE Access 7 (2019) 136073–136093, http://dx.doi.org/10.1109/ACCESS.2019.2941701.

[39] M. Abdalla, P.-A. Fouque, D. Pointcheval, Password-based authenticated key exchange in the three-party setting, in: S. Vaudenay (Ed.), Public Key Cryptography - PKC 2005, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 65–84.

[40] A.G. Reddy, A.K. Das, V. Odelu, A. Ahmad, J.S. Shin, A privacy preserving three-factor authenticated key agreement protocol for client–server environment, J. Ambient Intell. Humaniz. Comput. 10 (2) (2019) 661–680, http://dx.doi.org/10.1007/s12652-018-0716-4.

[41] J. Srinivas, A.K. Das, N. Kumar, J. Rodrigues, Cloud centric authentication for wearable healthcare monitoring system, IEEE Trans. Dependable Secure Comput. (2018) 1, http://dx.doi.org/10.1109/TDSC.2018.2828306.

[42] AVISPA. Automated Validation of Internet Security Protocols and Applications, 2018, http://www.avispa-project.org (accessed May, 2018).

[43] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P.H. Drielsma, P.C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, L. Vigneron, The AVISPA tool for the automated validation of internet security protocols and applications, in: K. Etessami, S.K. Rajamani (Eds.), Computer Aided Verification, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 281–285.

[44] D. Von Oheimb, The high-level protocol specification language HLPSL developed in the EU project AVISPA, in: Proceedings of APPSEM 2005 Workshop, 2005, pp. 1–17.

[45] D. Basin, S. Mödersheim, L. Viganò, OFMC: A symbolic model checker for security protocols, Int. J. Inf. Secur. 4 (3) (2005) 181–208, http://dx.doi.org/10.1007/s10207-004-0055-7.

[46] M. Turuani, The CL-atse protocol analyser, in: F. Pfenning (Ed.), Term Rewriting and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 277–286.

[47] A. Armando, R. Carbone, L. Compagna, SATMC: a SAT-based model checker for security protocols, business processes, and security APIs, Int. J. Softw. Tools Technol. Transfer 18 (2) (2016) 187–204, http://dx.doi.org/10.1007/s10009-015-0385-y.

[48] P. Kocher, J. Jaffe, B. Jun, Differential power analysis, in: M. Wiener (Ed.), Advances in Cryptology — CRYPTO' 99, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999, pp. 388–397.