A Lightweight and Anonymous Authentication and Key Agreement Protocol for Wireless Body Area Networks

Cong Pu, *Member, IEEE*, Haleigh Zerkle, Andrew Wall, Sunho Lim, *Senior Member, IEEE*,
Kim-Kwang Raymond Choo, *Senior Member, IEEE*, and Imtiaz Ahmed, *Member, IEEE*

Abstract*—As a major building block of Healthcare 4.0, wireless body area networks (WBANs) play an important role in collecting patient's real-time physical phenomena through small wearable or implantable intelligent medical devices and communicating with remote medical experts using short-range wireless communication techniques. However, the challenges of securing information access are partly evidenced by the difficulty in designing secure and efficient security protocols. For example, existing authentication and key agreement schemes have either potential security vulnerabilities or high communication and computation overhead. In this paper, we propose a lightweight and anonymous authentication and key agreement protocol, also called liteAuth, for WBANs. In our approach, mutual authentication and session key agreement are achieved using Tinkerbell map-based random shuffling, physical unclonable function, one-way hash function, and bitwise exclusive OR operation. The security of *liteAuth* is first verified using the AVISPA tool, and then its cyber resilience is analyzed. In addition, we develop a real-world testbed, implement *liteAuth* and two existing schemes (i.e., PSLAP and HARCI), and conduct experiments for performance evaluation and analysis. Experimental results indicate that *liteAuth* can improve the performance of communication overhead and computation time as well as reduce energy consumption, while meeting all security requirements.

***Index Terms*—Wireless Body Area Networks, Security and Privacy, Authentication and Key Agreement Protocol, Lightweight and Anonymous**

I. Introduction

Industry 4.0 and its major enabling technologies (i.e., automation and artificial intelligence) are revolutionizing traditional manufacturing and industrial practices [1]. This is particularly the case in healthcare domain. Taking advantage of Internet of Things, 5G, and artificial intelligence that drive Industry 4.0, the healthcare ecosystem is moving towards the era of Healthcare 4.0 [2], where various cyber and physical systems are seamlessly integrated to create digitalized healthcare products as well as services. For example, in the age of Healthcare 4.0, telesurgery can provide a precise diagnosis and deliver healthcare surgical services to remote patients using high-speed communication system [3]. According to [4], the size of global healthcare market is forecasted to reach about $12 trillion by 2022. In addition, the innovations in wearable and implantable techniques (a.k.a. wellness space) are expected to radically extend the human lifespan [5]. With the current state of technology, it is envisioned in the near future that Healthcare 4.0 will bring about a reformed healthcare system that makes the services of early diseases prediction and prevention available to everyone.

As a major building block of Healthcare 4.0, wireless body area networks (WBANs) are playing an important role in achieving the vision of ''pervasive healthcare'' that provides healthcare for anyone, anytime, and anywhere by removing time, locational, and any other restraints [6]. Generally speaking, WBANs are a collection of wearable or implantable medical sensors attached to or implanted in the patient body, where real-time vital signs (i.e., blood pressure, pulse, temperature, and heart rhythm, etc.) are collected and sent to nearby controller node (i.e., smartphone) via wireless channel [7]. The controller node then forwards the information to the cloud server, where the data will be stored and processed for further medical diagnosis by medical professionals. Therefore, WBANs are seen as a silver bullet that can provide reliable and robust health-monitoring services and help patients with disabilities to recuperate to normal conditions.

Despite all of WBANs' supposed benefits, the security and privacy issues have received growing attention in social media as well as the academic world over the past few years. A telling example is the recent revelation that the medical gear maker Medtronic [8] does not implement authentication or authorization in the Conexus telemetry protocol [9]. As a result, any adversary who is located within a range of roughly 25 feet from Medtronic devices utilizing the Conexus telemetry protocol can compromise the communication [10]. In the academic world, the authors in [7] discuss the primary threats and vulnerabilities existing in WBANs, and summarize state-of-the-art approaches that can provide the appropriate security and privacy protection for WBANs. Undoubtedly, the

Manuscript received April 19, 2021; revised August 16, 2021. This research was supported by West Virginia Research Higher Education Policy Commission, Division of Science and Research Grant # dsr.20.1698-001. (Corresponding author: Cong Pu.)

Cong Pu, Haleigh Zerkle, and Andrew Wall are with the Dept. of Computer Sciences and Electrical Engineering, Marshall University, Huntington, WV, 25755, United States (e-mail: cong.pu@ieee.org, zerkle15@marshall.edu, wall48@marshall.edu).

Sunho Lim is with the Dept. of Computer Science, Texas Tech University, Lubbock, TX 79409, United States (e-mail: sunho.lim@ttu.edu).

Kim-Kwang Raymond Choo is with the Dept. of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249, United States (e-mail: raymond.choo@fulbrightmail.org).

Imtiaz Ahmed is with the Dept. of Electrical Engineering and Computer Science, Howard University, Washington, DC 20059, United States (e-mail: imtiaz.ahmed@howard.edu).

abovementioned facts have demonstrated the importance of security and privacy while using WBANs.

To rebuild people's confidence in WBANs, it is imperative to develop secure communication protocols to resolve the security and privacy issues. However, it is easier said than done. First, the wireless communication between medical devices and controller node opens the door for adversaries to eavesdrop and steal private patient medical logs or personal information. Second, due to resource constraints (i.e., limited battery resource and computational power), traditional well-known but energy-hungry security techniques (i.e., RSA and AES [11]) can not be directly employed by WBANs. Third, a long-term permanent security key can be pre-loaded into medical devices and the authorized controller node can use a common master key to communicate with medical devices. However, the disclosure of master key will put all medical devices and the controller node in danger [12]. Therefore, the lightweight security techniques with the design consideration of balancing the tradeoff between performance and security are worth exploring, which becomes the main focus of this paper as a matter of course. In summary, the key contributions of this paper are summarized in threefold:

- We propose a lightweight and anonymous authentication and key agreement protocol, also called $liteAuth$, for WBANs, where mutual authentication and session key agreement are achieved using Tinkerbell map-based random shuffling, physical unclonable function (PUF), one-way hash function, and bitwise exclusive OR (XOR) operation.
- We implement $liteAuth$ in High-Level Protocol Specification Language (HLPSL) and verify its security using the well-known Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [13]. In addition, we analyze the cyber resilience of $liteAuth$ to show that it can defend against various WBAN-specific security attacks.
- We develop a real-world testbed consisting of one Dell laptop [14] and one Latte Panda development board [15]. We also revisit prior security protocols, PSLAP [16] and HARCI [17], implement them along with $liteAuth$ in Python, and deploy the programs in the testbed for performance comparison and analysis.

We conduct extensive experiments and measure the performance of $liteAuth$, PSLAP, and HARCI in terms of communication overhead, computation time, energy consumption, CPU time, and CPU cycles. Experimental results demonstrate that $liteAuth$ can not only achieve better performance compared to prior approaches, but also meet all security requirements, indicating a viable and competitive approach for ensuring security and data privacy in WBANs. To promote the broad adoption and drive creative advancement in the realm of security protocols within the WBAN community, we make the program codes available to the public at the

https://github.com/congpu/liteAuth.

The remaining parts of the paper is organized as follows. We present and analyze some existing literature in Section II. Section III gives a brief introduction to various techniques used in the paper. Section IV describes the network model, adversary model, and security requirements. In Section V, we present $liteAuth$ with details. Both security verification and security analysis are provided in Section VI. We conduct an experimental study and present results in Section VII. We further discuss the proposed scheme in Section VIII. Finally, Section IX makes a summary for the paper.

II. Related Work

Authentication and key agreement protocols are security mechanisms used in untrusted networks where all participants show their legality, verify other participants' identities, and distribute shared secret keys among them. Over the past few years, many authentication and key agreement protocols have been developed for WBANs.

In [18], the authors investigate the security issues in cloud-assisted WBANs, and design an identity-based anonymous authentication and key agreement scheme. Patient's medical and personal data are stored on a cloud server so that the cost of computation and storage can be optimized. In addition, a patient's identity information is hidden from other entities during all phases except the registration phase when the network administrator registers leaf node, root node, target node, and cloud servers, and computes their private keys. Although the protocol provides several security properties, it fails to achieve perfect forward secrecy as well as user revocation. The authors in [19] propose an anonymous mutual authentication and key agreement protocol in WBANs. The protocol has four phases: initialization, registration, authentication, and dynamic node update. The system administrator generates system parameters (i.e., master key) and registers intermediate and sensor nodes in the initialization and registration phase, respectively. In the authentication phase, sensor node and hub node authenticate each other and establish a session key using cryptographic operations. If necessary, new sensor nodes can be added into WBANs during dynamic node update phase. A major drawback of the protocol is the high computation and communication overhead. To secure data communication in the medical IoT, a mutual authentication scheme is proposed in [20]. The authors mainly focus on how to realize security objectives without requiring a server verification table that stores users' authentication parameters and sensitive data.

The authors in [21] propose a hash-chain-based and forward secure authentication scheme for WBANs in the healthcare IoT. The scheme consists of four phases: initialization, registration, authentication, and password change. The initialization phase initializes WBANs through finalizing secret key and cryptographic hash function. In the registration phase, sensor nodes and users register and synchronize secret information with gateway. With the help of gateway, sensor node and user can authenticate each other in the authentication phase. If a user needs to change the password, a sequence of steps should be taken during the password change phase. In [22], the authors develop a mutual authentication and session establishment protocol for tactile Internet driven remote surgery setups. Taking advantage of elliptic curve cryptography (ECC) and biometrics, secure communications can be established between the surgeon, the robotic arm, and the trusted authority.

In [23], the authors propose a security protocol using blockchain technique for WBAN. The personal digital assistant and blockchain node first create a secret key for secure communication. Then, a blind signature is generated by blockchain nodes so that the node anonymity can be guaranteed and the node eligibility can be verified. The authors in [24] propose an authentication protocol for telecare medicine information systems using hash functions and bitwise XOR operations. In [25], an authentication and encryption protocol is designed for WBAN, where the signal propagation characteristics and butterfly algorithm are adopted to achieve the security goals. Specifically, in order to distinguish the adversary from legitimate devices, an authentication scheme relying on signal propagation variations is designed, where the butterfly algorithm is adopted to generate random numbers which represent the signal propagation variations.

In [26], a biometrics-based authentication scheme is proposed for WBANs, where patients' physiological parameters are collected and used by wearable or implantable medical device(s) to generate a unique identifier (or biometrics) and authenticate with other communication entities. However, one of the requirements for the success of biometrics-based authentication schemes is the high performance and excellent stability of physiological parameters. In addition, the physiological features might vary significantly or even become unavailable during the phase of physiological parameters collection, which makes the biometrics-based authentication scheme unfeasible. Compared to the biometrics-based authentication schemes, our approach *liteAuth* adopts noise-resistant and reliable PUF to generate the critical information for mutual authentication and key agreement. This will guarantee that the critical information for security scheme can be regenerated and the security scheme is always feasible.

Generally speaking, our approach *liteAuth* has five absolute advantages over existing security solutions in WBANs. First, *liteAuth* is designed with the adoption of various lightweight computing operations such as PUF, chaotic system, hash function, as well as bitwise XOR so that it can keep its computational overhead to a minimum without sacrificing security concerns. Second, *liteAuth* supports anonymous authentication in WBANs, where the pseudonym, instead of real identity, is being used for communications between entities. As a result, the genuine identity of wireless medical device and control node can only be revealed by the trusted cloud server. Third, *liteAuth* requires both wireless medical device and control node to change their pseudonyms after each communication session. Thus, *liteAuth* can defend against identity fixation attack as well as prevent adversary from tracing wireless medical device's and control node's pattern of behavior. Fourth, *liteAuth* assumes that the control node is an untrusted entity, which differs from the opposite assumption made in other approaches. Fifth, to the best of our knowledge, *liteAuth* is the first security scheme using PUF and chaotic system based random shuffling to realize the mutual authentication and session key agreement between communication entities in WBANs.

In summary, most prior schemes primarily focus on how to secure WBANs with various techniques. However, little attention has been paid to the lightweight and anonymous approach focusing on Tinkerbell map-based random shuffling, physical unclonable function, one-way hash function, and bitwise XOR operation to achieve mutual authentication and session key agreement in WBANs.

III. Preliminary Background

A. Tinkerbell Map

Chaotic systems are regarded as dynamical systems, where the underlying patterns and deterministic laws determine the system's random states of disorder and irregularities. Since chaotic systems are extremely sensitive to the initial conditions, extensively diverging system outcomes can be generated with small differences in initial conditions. It is also believed that predicting long-term system behavior is impossible in general. Tinkerbell map [27] is one of such chaotic systems that exhibits very rich dynamics and chaotic behaviors. Its general form can be represented as follows

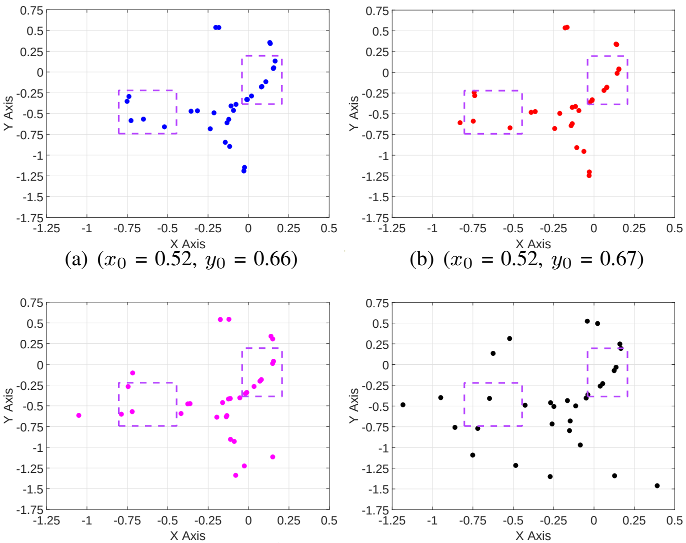
$$\begin{cases} x_{n+1} = x_n^2 - y_n^2 + ax_n + by_n \\ y_{n+1} = 2x_ny_n + cx_n + dy_n \end{cases} \tag{1}$$

where a, b, c, and d are system parameters, and n represents the discrete iteration step. According to Eq. (1), Tinkerbell map can also be viewed as a two-dimensional discrete-time dynamical system that maps a point $(x_n,\ y_n)$ to a new point $(x_{n+1},\ y_{n+1})$ in the two-dimensional coordinate plane. With certain system parameter values and initial condition $(x_0,\ y_0)$, Tinkerbell map demonstrates chaos, which is represented as a random sequence of points in the coordinate plane. However, without the same initial condition $(x_0,\ y_0)$, Tinkerbell map is unable to produce the same chaos (i.e., the same sequence of points) even if the same system parameter values are provided. To show perfect chaotic behaviors, various system parameter values have been extensively studied in the past [28]. For example, with a = 0.9, b = −0.6, c = 2.0, and d = 0.5, any change in the initial condition $(x_0,\ y_0)$ will produce totally different chaos, which are shown in Fig. 1.

In our approach *liteAuth*, a random sequence of points which are generated by Tinkerbell map with a specific initial condition (PUF challenge-response pair) are used to randomly shuffle the communication message which is represented as a byte array. The basic idea is that the first coordinate point in the sequence is converted into a unique integer, which indicates the new location of the first byte of the communication message in the output array; and the second coordinate point in the sequence is converted into another unique integer, which represents the new location of the second byte of the communication message in the output array. The same idea will be applied to the rest of bytes in the communication message array. When the last byte in the communication message array is placed at the new location in the output array, the shuffling process is completed, and the newly generated output array is the encrypted communication message.

B. Physical Unclonable Function

Every integrated circuit has minor physical differences even though they come from the same production line. This unique



(a) (x_0 = 0.52, y_0 = 0.66)

(b) (x_0 = 0.52, y_0 = 0.67)

(c) (x_0 = 0.72, y_0 = 0.57)

(d) (x_0 = 0.72, y_0 = 0.77)

Fig. 1. Tinkerbell map with different initial conditions after 30 iterations, where the same regions are highlighted with dashed-boxes to show different chaos.

characteristic of integrated circuit has been widely adopted to design a physical unclonable function (PUF) [29], which is considered to be the unique identity of an electronic device. General speaking, we can use a chaotic one-way function to represent a PUF. The input value of a PUF is termed ‘‘challenge’’, while the corresponding output value is termed ‘‘response’’. The input-output bundle is called a challenge-response pair (CRP). As the PUF is designed based on the physical differences in the integrated circuit, the CRP is exclusive to each PUF. In other words, if we feed the same response into the PUF, the same challenge will be outputted. If the same response is provided to different PUFs, totally distinct responses will be generated. For the sake of simplicity, in this paper a PUF is represented as a 256-bit hash function $R = F_{puf}(C)$, where the challenge (C) and the response (R) are both in the form of a string of bits [30]. In recent years, various designs of noise-resistant and reliable PUF [31] have been investigated, where almost 0% bit error rates in noisy environment with voltage fluctuations and wide temperature ranges can be achieved. Thus, in this paper, we assume that an ideal and noise-resistant PUF is adopted.

C. One-Way Hash Function

An one-way hash function is a mathematical function that takes a variable-length input string and converts it into a fixed-length binary sequence which is widely known as hash value [11]. The hash value is computationally difficult to invert, that is, generate the original string from the hash value. In general, an one-way hash function can be expressed as $h = H(M)$, where $H(\cdot)$ is the one-way hash function, $M = \{0,1\}^*$ is a set of variable length strings, and $h = \{0,1\}^m$ is a set of fixed length (saying m bits) strings.

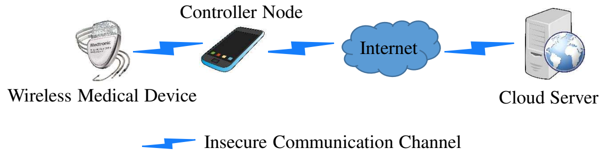


Fig. 2. Network model.

IV. System Model

A. Network and Adversary Model

As shown in Fig. 2, the network model consists of three major entities: wireless (wearable or implantable) medical device(s), control node, and cloud server.

- Cloud Server: It is considered as a trusted entity. It is responsible for registering every wireless medical device and control node via storing their real identity, initial pseudonym, and initial CRP. In addition, it assists the wireless medical device and control node to achieve mutual authentication and establish a secure session key before they share any critical information.
- Wireless Medical Device: It gathers patient’s physical phenomena and sends the information to nearby control node via insecure communication channel. After the wireless medical device and control node verify each other’s validity, they establish a secret session key for secure communication. Each wireless medical device is equipped with a PUF-enabled integrated circuit. In addition, the wireless medical device stores its real identity, the challenge of initial CRP, and the initial condition of Tinkerbell map.
- Control Node: It collects patient’s information from the wireless medical device and send them to the cloud server via insecure communication channel. Before accessing and communicating with the wireless medical device, the identity of control node should be verified by the cloud server. It is also assumed to be furnished with a PUF chip, and store its real identity, the challenge of initial CRP, and the initial condition of Tinkerbell map.

We adopt the well-known adversary model specified in [11], where two entities are assumed to be untrustworthy if they are communicating over an insecure communication channel. An adversary can eavesdrop on or monitor existing communication and use a replay attack to disrupt operations of legitimate entities. In addition, an adversary may make an effort to authenticate itself to a wireless medical device or control node to cause strategic damage without being detected. For example, if an adversary authenticates with the wireless medical device (i.e., medical insulin pump) as a genuine entity, it may alter device settings, disable device’s critical functionalities, or even deliver a shock on command (i.e., over dosage of insulin) to threaten patient’s life. Therefore, an authentication and key agreement protocol is needed to protect WBANs from potential security threats.

B. Security Requirements

According to the criteria of secure authentication and key agreement scheme [11], we design $liteAuth$ to meet the following security requirements:

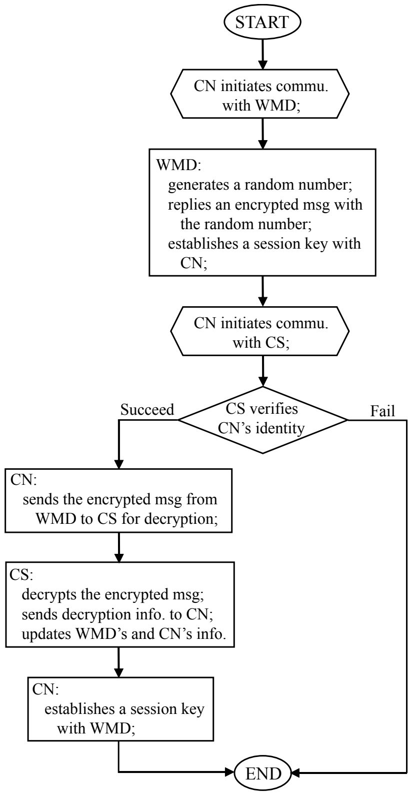

START
CN initiates commu. with WMD;
WMD: generates a random number; replies an encrypted msg with the random number; establishes a session key with CN;
CN initiates commu. with CS;
CS verifies CN's identity
Succeed
Fail
CN: sends the encrypted msg from WMD to CS for decryption;
CS: decrypts the encrypted msg; sends decryption info. to CN; updates WMD's and CN's info.
CN: establishes a session key with WMD;
END

Fig. 3. A flowchart of lightweight authentication and key agreement protocol *liteAuth*.

- Mutual Authentication: Our approach shall assure that two communicating entities are authentic and be able to verify each other's identity.
- Integrity: Our approach shall assure that the source of messages and the content of messages can be verified by the receiving entity.
- Anonymity: Our approach shall guarantee that the real identity of each entity is unknown to any other entities except the trusted cloud server, even if an adversary captured messages.
- Session Key Agreement: Our approach shall assure that a secret session key will be established between legitimate communicating entities for further communication.
- Immune Against Attacks: Our approach shall withstand various attacks such as wireless medical device impersonation attack, control node spoofing attack, etc.

V. The Proposed Authentication and Key Agreement Protocol

The basic idea of *liteAuth* is to use PUF and Tinkerbell map based random shuffling to achieve mutual authentication and session key agreement between WBAN communication entities before sharing any critical information. First, the control node initiates the communication with the wireless medical device to collect patient's information by sending a message piggybacked with a random number. After receiving the communication request from the control node, the wireless medical device replies an encrypted message piggybacked with a random number to the control node. In order to decrypt the encrypted message from the wireless medical device, the control node needs to authenticate with the cloud server and retrieve the decryption information. Finally, two random numbers, one from the control node and another from the wireless medical device, will be used to establish a secure session key between the control node and the wireless medical device for future communication. A flowchart of *liteAuth* is shown in Fig. 3.

TABLE I
Notations and Their Descriptions

Notation	Description
WMD_i	Real identity of the *i*th wireless medical device
CN_j	Real identity of the *j*th control node
CS_k	Real identity of the *k*th cloud server
N	Random number
t	Current timestamp
$P_{WMD_i^t}$	Pseudonym of WMD_i at time t
$P_{CN_j^t}$	Pseudonym of CN_j at time t
N^t	Random number generated at time t
$(C^t,\, R^t)$	PUF CRP at time t
$F_{puf}(\cdot)$	PUF function
$S(\cdot)_{(C^t,R^t)}$	Random shuffling with CRP
$S^{-1}(\cdot)_{(C^t,R^t)}$	Reverse process of random shuffling with CRP
$C(\cdot)$	Message authentication code (MAC) function
$H(\cdot)$	Collision-resistant one-way hash function
$\oplus$	Bitwise XOR operation
$\|$	Concatenation operation
M_x	The *x*th message
MAC_x	MAC of the *x*th message
SK	Secure session key

$*$ The subscript is used to represent the entity who is associated with the notation. E.g., $(C_i^t,\, R_i^t)$ is the CRP of PUF for the entity *i* at time *t*; $N_{j,i}^t$ is the random number being used by the entity *j* and *i* at time *t*.

Considering the scenario that a wireless medical device WMD_i periodically wakes up and gathers vital signs for the estimation of patient's physical state. Every once in a while, a control node CN_j communicates with the wireless medical device WMD_i to collect patient's information, and then sends the information to the cloud server CS_k for further processing and analysis. Before sharing any patient's information, mutual authentication and secret session key agreement have to be established between the wireless medical device WMD_i and control node CN_j as well as between the control node CN_j and cloud server CS_k, respectively. Table I lists all notations used in this paper. The detailed steps are as follows:

1) The control node CN_j first computes its pseudonym as $P_{CN_j^t} = H(CN_j \parallel R_j^t)$ using its real identity CN_j and the response of CRP R_j^t. Then it generates a random number $N_{j,i}^t$ and calculates the message authentication code (MAC) MAC_0 as follows
$$MAC_0 = C(P_{CN_j^t} \| P_{WMD_i^t} \| N_{j,i}^t).$$
Finally, it sends the message $[P_{CN_j^t},\, N_{j,i}^t,\, MAC_0]$ to the wireless medical device WMD_i to initialize the communication. Here, $P_{WMD_i^t}$ is the pseudonym of wireless medical device WMD_i.
2) The wireless medical device WMD_i first computes its pseudonym as $P_{WMD_i^t} = H(WMD_i \parallel R_i^t)$ using its real identity WMD_i and the response of CRP R_i^t. Then it generates two random numbers, $N_{i,j}^t$ and $N_{i,k}^t$. $N_{i,j}^t$ and $N_{j,i}^t$ are used to establish the secret session key between the wireless medical device WMD_i and control node

CN_j, while $N_{i,j}^t$ and $N_{i,k}^t$ are used to update wireless medical device WMD_i's CRP and pseudonym. Next, it calculates the encrypted message M_1 and M_2 as follows

$$M_1 = S(P_{WMD_i^t} \| P_{CN_j^t} \| N_{i,j}^t)_{(C_i^t, R_i^t)},$$

$$M_2 = S(P_{WMD_i^t} \| CS_k \| N_{i,k}^t)_{(C_i^t, R_i^t)}.$$

It also calculates MAC MAC_1 and MAC_2 as

$$MAC_1 = C(M_1 \| P_{WMD_i^t} \| P_{CN_j^t} \| N_{i,j}^t),$$

$$MAC_2 = C(M_2 \| P_{WMD_i^t} \| CS_k \| N_{i,k}^t).$$

After that, it calculates a new CRP, $C_i^{t+1} = S(N_{i,j}^t \| N_{i,k}^t)_{(C_i^t, R_i^t)}$ and $R_i^{t+1} = F_{puf}(C_i^{t+1})$. Next, it calculates the encrypted message M_3 and its corresponding MAC MAC_3 as follows

$$M_3 = S(P_{WMD_i^t} \| CS_k \| R_i^{t+1})_{(C_i^t, R_i^t)},$$

$$MAC_3 = C(M_3 \| P_{WMD_i^t} \| CS_k \| R_i^{t+1}).$$

Finally, it obtains the secret session key as $SK_{i,j} = H(N_{i,j}^t) \oplus H(N_{j,i}^t)$ which will be used to communicate with the control node CN_j later, and sends the message $[P_{WMD_i^t}, M_1, MAC_1, M_2, MAC_2, M_3, MAC_3]$ to the control node CN_j.

3) The control node CN_j temporarily stores the received message $[P_{WMD_i^t}, M_1, MAC_1, M_2, MAC_2, M_3, MAC_3]$ for future use. It generates a random number $N_{j,k}^t$ and calculates the message M_4 and MAC MAC_4 as follows

$$M_4 = S(P_{CN_j^t} \| CS_k \| N_{j,k}^t)_{(C_j^t, R_j^t)},$$

$$MAC_4 = C(M_4 \| P_{CN_j^t} \| CS_k \| N_{j,k}^t).$$

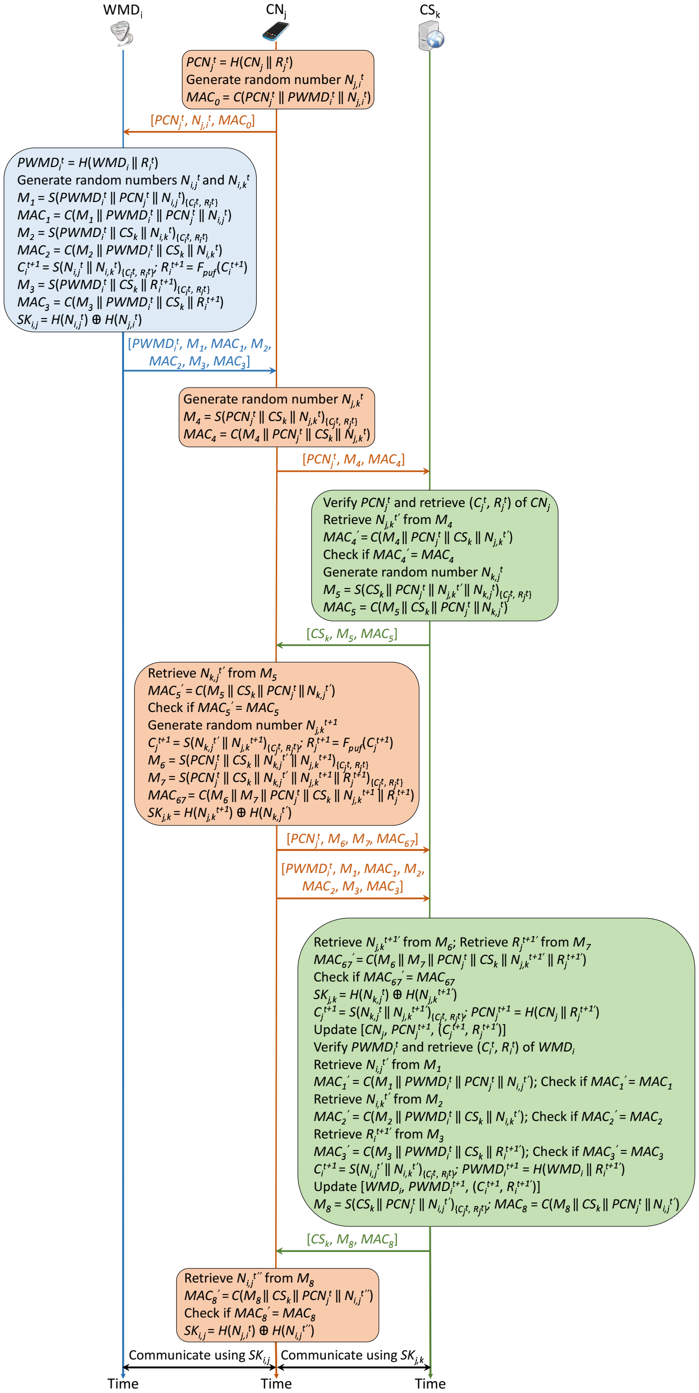
Then, it sends the message $[P_{CN_j^t}, M_4, MAC_4]$ to the cloud server CS_k.

4) After receiving the message from the control node CN_j, the cloud server CS_k first tries to locate $P_{CN_j^t}$ in the database. If $P_{CN_j^t}$ is not found, the communication is rejected. Otherwise, it fetches the entry $[CN_j, P_{CN_j^t}, (C_j^t, R_j^t)]$ for the control node CN_j. Then, it retrieves ${N_{j,k}^t}'$ from M_4 through $S^{-1}(M_4)$, which is the reverse process of shuffling with the CRP (C_j^t, R_j^t) as initial condition. With ${N_{j,k}^t}'$, it can calculate $MAC_4' = C(M_4 \| P_{CN_j^t} \| CS_k \| {N_{j,k}^t}')$ and check it with the received MAC_4. If $MAC_4' = MAC_4$, the message verification succeeds. Otherwise, it discards the message and terminates the communication. Next, it generates a random number $N_{k,j}^t$, and calculates the encrypted message M_5 and its MAC MAC_5 as follows

$$M_5 = S(CS_k \| P_{CN_j^t} \| N_{j,k}^{t'} \| N_{k,j}^t)_{(C_j^t, R_j^t)},$$

$$MAC_5 = C(M_5 \| CS_k \| P_{CN_j^t} \| N_{k,j}^t).$$

Finally, it sends the message $[CS_k, M_5, MAC_5]$ to the control node CN_j.



WMD$_i$
CN$_j$
CS$_k$
$PCN_j^t = H(CN_j \| R_j^t)$
Generate random number $N_{j,i}^t$
$MAC_0 = C(PCN_j^t \| PWMD_i^t \| N_{j,i}^t)$
$[PCN_j^t, N_{j,i}^t, MAC_0]$
$PWMD_i^t = H(WMD_i \| R_i^t)$
Generate random numbers $N_{i,j}^t$ and $N_{i,k}^t$
$M_1 = S(PWMD_i^t \| PCN_j^t \| N_{i,j}^t)_{\{C_i^t, R_i^t\}}$
$MAC_1 = C(M_1 \| PWMD_i^t \| PCN_j^t \| N_{i,j}^t)$
$M_2 = S(PWMD_i^t \| CS_k \| N_{i,k}^t)_{\{C_i^t, R_i^t\}}$
$MAC_2 = C(M_2 \| PWMD_i^t \| CS_k \| N_{i,k}^t)$
$C_i^{t+1} = S(N_{i,j}^t \| N_{i,k}^t)_{\{C_i^t, R_i^t\}}$; $R_i^{t+1} = F_{puf}(C_i^{t+1})$
$M_3 = S(PWMD_i^t \| CS_k \| R_i^{t+1})_{\{C_i^t, R_i^t\}}$
$MAC_3 = C(M_3 \| PWMD_i^t \| CS_k \| R_i^{t+1})$
$SK_{i,j} = H(N_{i,j}^t) \oplus H(N_{j,i}^t)$
$[PWMD_i^t, M_1, MAC_1, M_2, MAC_2, M_3, MAC_3]$
Generate random number $N_{j,k}^t$
$M_4 = S(PCN_j^t \| CS_k \| N_{j,k}^t)_{\{C_j^t, R_j^t\}}$
$MAC_4 = C(M_4 \| PCN_j^t \| CS_k \| N_{j,k}^t)$
$[PCN_j^t, M_4, MAC_4]$
Verify PCN_j^t and retrieve (C_j^t, R_j^t) of CN_j
Retrieve $N_{j,k}^{t'}$ from M_4
$MAC_4' = C(M_4 \| PCN_j^t \| CS_k \| N_{j,k}^{t'})$
Check if $MAC_4' = MAC_4$
Generate random number $N_{k,j}^t$
$M_5 = S(CS_k \| PCN_j^t \| N_{j,k}^{t'} \| N_{k,j}^t)_{\{C_j^t, R_j^t\}}$
$MAC_5 = C(M_5 \| CS_k \| PCN_j^t \| N_{k,j}^t)$
$[CS_k, M_5, MAC_5]$
Retrieve $N_{k,j}^{t'}$ from M_5
$MAC_5' = C(M_5 \| CS_k \| PCN_j^t \| N_{k,j}^{t'})$
Check if $MAC_5' = MAC_5$
Generate random number $N_{j,k}^{t+1}$
$C_j^{t+1} = S(N_{k,j}^{t'} \| N_{j,k}^{t+1})_{\{C_j^t, R_j^t\}}$; $R_j^{t+1} = F_{puf}(C_j^{t+1})$
$M_6 = S(PCN_j^t \| CS_k \| N_{k,j}^{t'} \| N_{j,k}^{t+1})_{\{C_j^t, R_j^t\}}$
$M_7 = S(PCN_j^t \| CS_k \| N_{k,j}^{t'} \| N_{j,k}^{t+1} \| R_j^{t+1})_{\{C_j^t, R_j^t\}}$
$MAC_{67} = C(M_6 \| M_7 \| PCN_j^t \| CS_k \| N_{j,k}^{t+1} \| R_j^{t+1})$
$SK_{j,k} = H(N_{j,k}^{t+1}) \oplus H(N_{k,j}^{t'})$
$[PCN_j^t, M_6, M_7, MAC_{67}]$
$[PWMD_i^t, M_1, MAC_1, M_2, MAC_2, M_3, MAC_3]$
Retrieve $N_{j,k}^{t+1'}$ from M_6; Retrieve $R_j^{t+1'}$ from M_7
$MAC_{67}' = C(M_6 \| M_7 \| PCN_j^t \| CS_k \| N_{j,k}^{t+1'} \| R_j^{t+1'})$
Check if $MAC_{67}' = MAC_{67}$
$SK_{j,k} = H(N_{k,j}^t) \oplus H(N_{j,k}^{t+1'})$
$C_j^{t+1} = S(N_{k,j}^t \| N_{j,k}^{t+1'})_{\{C_j^t, R_j^t\}}$; $PCN_j^{t+1} = H(CN_j \| R_j^{t+1'})$
Update $[CN_j, PCN_j^{t+1}, (C_j^{t+1}, R_j^{t+1'})]$
Verify $PWMD_i^t$ and retrieve (C_i^t, R_i^t) of WMD_i
Retrieve $N_{i,j}^{t'}$ from M_1
$MAC_1' = C(M_1 \| PWMD_i^t \| PCN_j^t \| N_{i,j}^{t'})$; Check if $MAC_1' = MAC_1$
Retrieve $N_{i,k}^{t'}$ from M_2
$MAC_2' = C(M_2 \| PWMD_i^t \| CS_k \| N_{i,k}^{t'})$; Check if $MAC_2' = MAC_2$
Retrieve $R_i^{t+1'}$ from M_3
$MAC_3' = C(M_3 \| PWMD_i^t \| CS_k \| R_i^{t+1'})$; Check if $MAC_3' = MAC_3$
$C_i^{t+1} = S(N_{i,j}^{t'} \| N_{i,k}^{t'})_{\{C_i^t, R_i^t\}}$; $PWMD_i^{t+1} = H(WMD_i \| R_i^{t+1'})$
Update $[WMD_i, PWMD_i^{t+1}, (C_i^{t+1}, R_i^{t+1'})]$
$M_8 = S(CS_k \| PCN_j^t \| N_{i,j}^{t'})_{\{C_j^t, R_j^t\}}$; $MAC_8 = C(M_8 \| CS_k \| PCN_j^t \| N_{i,j}^{t'})$
$[CS_k, M_8, MAC_8]$
Retrieve $N_{i,j}^{t''}$ from M_8
$MAC_8' = C(M_8 \| CS_k \| PCN_j^t \| N_{i,j}^{t''})$
Check if $MAC_8' = MAC_8$
$SK_{i,j} = H(N_{j,i}^t) \oplus H(N_{i,j}^{t''})$
Communicate using $SK_{i,j}$
Communicate using $SK_{j,k}$
Time
Time
Time

Fig. 4. Lightweight authentication and key agreement protocol $liteAuth$.

5) The control node CN_j first retrieves ${N_{k,j}^t}'$ from M_5 through $S^{-1}(M_5)$ and calculates MAC_5' as follows

$$MAC_5' = C(M_5 \| CS_k \| P_{CN_j^t} \| N_{k,j}^{t'}).$$

If $MAC_5' = MAC_5$, the message verification succeeds. Otherwise, it discards the message. Then, it generates a random number $N_{j,k}^{t+1}$ and computes its new CRP as follows

$$C_j^{t+1} = S(N_{k,j}^{t'} \| N_{j,k}^{t+1})_{(C_j^t, R_j^t)},$$

$$R_j^{t+1} = F_{puf}(C_j^{t+1}).$$

After that, it calculates the following

$$M_6 = S(P_{CN_j^t} \| CS_k \| N_{k,j}^{t'} \| N_{j,k}^{t+1})_{(C_j^t, R_j^t)},$$

$$M_7 = S(P_{CN_j^t} \| CS_k \| N_{k,j}^{t'} \| N_{j,k}^{t+1} \| R_j^{t+1})_{(C_j^t, R_j^t)},$$

$$MAC_{67} = C(M_6 \| M_7 \| P_{CN_j^t} \| CS_k \| N_{j,k}^{t+1} \| R_j^{t+1}).$$

Finally, it sends the message [$P_{CN_j^t}$, M_6, M_7, MAC_{67}] to the cloud server CS_k, and calculates the secret session key to be used for the communication with the cloud server CS_k as follows

$$SK_{j,k} = H(N_{j,k}^{t+1}) \oplus H(N_{k,j}^{t'}).$$

It also sends the previously received message [$P_{WMD_i^t}$, M_1, MAC_1, M_2, MAC_2, M_3, MAC_3] to the cloud server CS_k for the decoding of random number $N_{i,j}^t$ and $N_{i,k}^t$.

6) The cloud server CS_k first retrieves $N_{j,k}^{t+1'}$ and $R_j^{t+1'}$ from M_6 and M_7 through $S^{-1}(M_6)$ and $S^{-1}(M_7)$, respectively. Then, it calculates $MAC_{67}^{'}$ as follows

$$MAC_{67}^{'} = C(M_6 \| M_7 \| P_{CN_j^t} \| CS_k \| N_{j,k}^{t+1'} \| R_j^{t+1'}).$$

If $MAC_{67}^{'} = MAC_{67}$, the message verification succeeds. Otherwise, it discards the message. Next, it calculates the secret session key for the communication with the control node CN_j as follows

$$SK_{j,k} = H(N_{k,j}^t) \oplus H(N_{j,k}^{t+1'}).$$

After that, it computes control node CN_j's new CRP challenge C_j^{t+1} and new pseudonym $P_{CN_j^{t+1}}$,

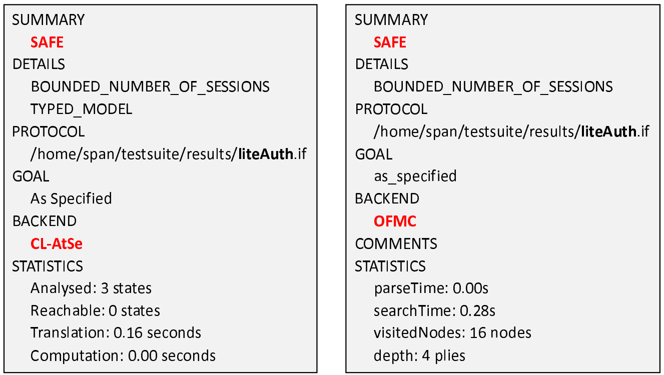
$$C_j^{t+1} = S(N_{k,j}^t \| N_{j,k}^{t+1'})_{(C_j^t, R_j^t)},$$

$$P_{CN_j^{t+1}} = H(CN_j \| R_j^{t+1'}),$$

and then updates the entry [CN_j, $P_{CN_j^{t+1}}$, (C_j^{t+1}, $R_j^{t+1'}$)] in the database. Then, it retrieves $N_{i,j}^{t}{'}$, $N_{i,k}^{t}{'}$, and $R_i^{t+1'}$ from M_1, M_2, and M_3 through $S^{-1}(M_1)$, $S^{-1}(M_2)$, and $S^{-1}(M_3)$, respectively. After verifying with MAC_1, MAC_2, and MAC_3, it computes wireless medical device WMD_i's new CRP challenge C_i^{t+1} and new pseudonym $P_{WMD_i^{t+1}}$, and then updates the entry [WMD_i, $P_{WMD_i^{t+1}}$, (C_i^{t+1}, $R_i^{t+1'}$)] in the database. Finally, it shares the random number $N_{i,j}^{t}{'}$ with the control node CN_j by sending the message [CS_k, M_8, MAC_8] to the control node CN_j.

7) The control node CN_j first retrieves $N_{i,j}^{t}{''}$ from M_8 through $S^{-1}(M_8)$, and calculates $MAC_8^{'} = C(M_8 \| CS_k \| P_{CN_j^t} \| N_{i,j}^{t}{''})$ and check it with the received MAC_8. If $MAC_8^{'} = MAC_8$, the message verification succeeds and it calculates the secret session key for the communication with the wireless medical device WMD_i as follows

$$SK_{i,j} = H(N_{j,i}^t) \oplus H(N_{i,j}^{t''}).$$



```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/span/testsuite/results/liteAuth.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed: 3 states
  Reachable: 0 states
  Translation: 0.16 seconds
  Computation: 0.00 seconds
```

(a)

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/testsuite/results/liteAuth.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.28s
  visitedNodes: 16 nodes
  depth: 4 plies
```

(b)

Fig. 5. Results of security verification using CL-AtSe and OFMC back-ends in AVISPA.

By this time, mutual authentication and secret session key establishment have been completed for the communication between the wireless medical device WMD_i and control node CN_j, and between the control node CN_j and cloud server CS_k, respectively. The above process is shown in Fig. 4.

VI. Security Verification and Analysis

A. *Security Verification Using AVISPA*

In order to verify whether *liteAuth* can defend against man-in-the-middle attack and replay attack, we evaluate the security performance of *liteAuth* using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool [32]. AVISPA is a push-button tool that integrates various back-ends. And each back-end is implemented using advanced automatic analysis techniques. In AVISPA, security protocols and their properties can be specified as a security problem using High-Level Protocol Specification Language (HLPSL) [32]. HLPSL is a role-based and modular programming language, and provides a variety of constructs such as data structures, control flow, intruder models, cryptographic primitives, etc. The detailed description of AVISPA and the reference manual of HLPSL can be found in [32]. We implement *liteAuth* in HLPSL, where there are three basic roles: wireless medical device, control node, and cloud server. In addition to these three basic roles, the other four mandatory roles, such as session, goal, environment, and intruder roles, are also implemented. We select Constraint-Logic-based Attack Searcher (CL-AtSe) and On-the-fly Model-Checker (OFMC) back-ends to evaluate the security performance of *liteAuth*. Using CL-AtSe, the security protocol specification can be translated into a set of constraints which are used to discover potential attacks on the protocols. OFMC is widely used for detecting attacks as well as proving the correctness of protocol for a bounded number of sessions. Finally, we set up a complete and fully functional SPAN+AVISPA [33] on Ubuntu 10.04 which is running in Virtual Box [34] on a Dell Inspiron 15 Plus laptop [14]. The results of security verification are shown in Fig. 5. As we can see that *liteAuth* is secure against replay attack and man-in-the-middle attack. The HLPSL security verification program of CL-AtSe and OFMC can be found at the https://github.com/congpu/liteAuth.

B. Security Analysis

This subsection exhibits how *liteAuth* satisfies the required security requirements and defends against control node capture attack, wireless medical device impersonation attack, message modification attack, and cloud server spoofing attack.

Mutual Authentication: *liteAuth* can achieve mutual authentication between communication entities in WBANs. This is because the cloud server will verify the identity of wireless medical device and control node using their real identity, initial pseudonym, and initial CRP. Therefore, *liteAuth* can achieve mutual authentication.

Integrity: *liteAuth* can achieve integrity so that the source of messages and the content of messages can be verified by the receiving entity. This is because each communication message is encrypted using the proposed PUF and Tinkerbell map based random shuffling scheme. In addition, a message authentication code (MAC) is also generated using the one-way hash function for each encrypted communication message. Therefore, *liteAuth* can achieve integrity.

Anonymity: *liteAuth* can support anonymous communication in WBANs. This is because the real identity of wireless medical device and control node are not transmitted directly in plaintext, but in the pseudonym format. In addition, the pseudonym of wireless medical device and control node will be updated after each communication session. Therefore, *liteAuth* can achieve anonymity.

Session Key Agreement: *liteAuth* can achieve session key agreement between communication entities in WBANs. This is because the control server will assist wireless medical device and control node to obtain the unique random numbers so that they can compute the secure session key and use it for future communications. Therefore, *liteAuth* can achieve session key agreement.

Control Node Capture Attack: We assume that an adversary has successfully captured a control node CN_j who is communicating with the cloud server CS_k using session key $SK_{j,k}$. Through physical memory disclosure attacks, the adversary can retrieve stored information such as control node's real identity CN_j and session key $SK_{j,k}$. As a result, the current communication session between the control node CN_j and cloud server CS_k would be compromised by the adversary with the obtained session key $SK_{j,k}$. In order to compromise future communications with the cloud server CS_k, the adversary has to obtain the valid CRP (C_j^{t+1}, R_j^{t+1}) from the control node CN_j. This is because a new session key (i.e., requiring new random numbers and random shuffling with new CRP) will be generated for each communication session. Thus, the adversary may try to probe or alter the integrated circuit of control node CN_j to retrieve CRP (C_j^{t+1}, R_j^{t+1}). However, the adversary can only obtain the CRP challenge C_j^{t+1}, since the CRP response R_j^{t+1} is dynamically calculated via $F_{puf}(C_j^{t+1})$. In addition, this probing or alteration attempt will inevitably change integrated circuit's physical characteristics, and finally destroy the PUF. Thus, the adversary cannot obtain the valid CRP (C_j^{t+1}, R_j^{t+1}) to compromise future communications. Last but not least, since each control node will use a different secret session key to communicate with the cloud server CS_k, the communication sessions between other non-captured control nodes and the cloud server CS_k are still secure. In summary, *liteAuth* is secure against control node capture attack.

Control Node Impersonation Attack: We assume that an adversary tries to masquerade as a legitimate control node CN_j to communicate with the cloud server CS_k for malicious purposes. First, the adversary has to send an authentication request, [$P_{CN_j^t}$, M_4, MAC_4], to the cloud server CS_k. The adversary can easily generate a random number $N_{j,k}^t$. However, it cannot shuffle and calculate valid M_4 which can be correctly decoded by the cloud server CS_k. This is because the adversary does not have the valid CRP (C_j^t, R_j^t) which is stored in the cloud server CS_k's database. Thus, the authentication request will be rejected by the cloud server CS_k and the adversary cannot establish valid communication with the cloud server CS_k by impersonating a legitimate control node CN_j. Thus, *liteAuth* can defend against control node impersonation attack.

Message Modification Attack: We assume that an adversary captures and modifies the message, either M_1, M_2, M_3, M_4, or M_5, transmitting between a wireless medical device WMD_i and a control node CN_j, or between a control node CN_j and a cloud server CS_k. Since both the control node CN_j and cloud server CS_k will verify the received message by checking the piggybacked MAC, i.e., $MAC_4 \overset{?}{=} MAC_4^{'}$, they can easily detect any modification of messages. As a result, *liteAuth* is immune against message modification attack.

Cloud Server Spoofing Attack: We assume that an adversary already captured the message [PID_j^t, M_4, MAC_4] and attempts to imitate a legitimate cloud server CS_k to communicate with a control node CN_j. Since the adversary does not have the valid CRP (C_j^t, R_j^t), thus, it cannot retrieve the random number $N_{j,k}^t$ piggybacked in M_4 correctly. The adversary can make up a random number to generate the message [M_5, MAC_5]. However, when the control node CN_j receives the message [M_5, MAC_5], it can easily detect the misbehavior through checking whether MAC_5 equals to $MAC_5^{'}$ and reject the following communication. Finally, *liteAuth* is resilient against cloud server spoofing attack.

VII. Performance Evaluation

A. Experimental Testbed and Benchmark Schemes

For experimental study, we build a real-world testbed which consists of one Dell Inspiron 15 Plus laptop [14] and one Latte Panda development board [15]. The Latte Panda development board is equipped with a power bank which can supply energy for hours. With regard to testbed specifications, the Dell Inspiron 15 Plus laptop is running a 64-bit Windows 10 Home operating system, and its central processing unit (CPU) is the 10th Generation Intel Core i7-10750H, 12MB Cache, up to 5.0 GHz. The Latte Panda development board comes preinstalled with a full version of Windows 10 Home operating system, and has Intel Cherry Trail Z8350 Quad Core processor, 2M cache, up to 1.92 GHz, and 4GB random-access memory (RAM). The snapshot of testbed is shown in Fig 6, where the laptop is used to simulate the cloud server, and the Latte Panda development

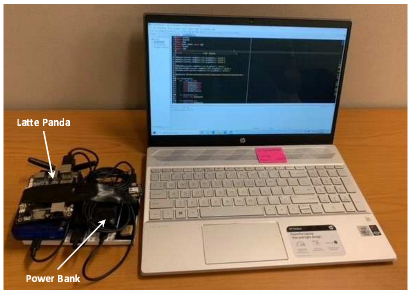


Fig. 6. Real-world testbed with one Dell Inspiron 15 Plus laptop and one Latte Panda development board.

board is used to mimic the wireless medical device and the control node, respectively. We implement $liteAuth$ and two benchmark schemes in Python, and deploy the programs in the LiClipse environment [35] which is set up in the Latte Panda development board as well as the laptop.

According to [30], we implement the PUF as a 256-bit hash function [36]. In addition, the random shuffling function is implemented as follows. First, the to-be-shuffled message is represented as an array. Second, the CRP pair (C_i^t, R_i^t) is used as the initial condition of Tinkerbell map (Eq. 1) to generate a sequence of points. Third, starting from the first point in the sequence, the coordinates of a point are converted into an unique integer, indicating the new location where the first element of array is to be put in the output array. Now considering the array element from the second to the last, the abovementioned process is repeated till the last array element is shuffled. Finally, the output array contains the shuffled message. Please note that the random shuffling function is executed differently in every communication session. This is because the wireless medical device, control node, and cloud server will compute a new CRP pair during the process of mutual authentication and session key establishment. Since the CRP pair is used as the initial condition of Tinkerbell map and a minor change of initial condition in the Tinkerbell map will cause the generation of distinct sequence of points (see more details about Tinkerbell map's features in Section III), the random shuffling operation is performed differently in every communication session.

We revisit prior security protocols, PSLAP [16] and HARCI [17], and implement them to work in the testbed for performance comparison and analysis. The original idea of these two benchmark schemes are briefly discussed in the following:

- PSLAP [16]: In PSLAP, there are three phases: initialization, registration, and authentication. In the initialization phase, the sensor node chooses a master key and stores it in the hub server. During the registration phase, the sensor node, access point, and hub server join the network, register at the server administrator, and get their identities and critical information via a secure channel. In the authentication process, the sensor node and hub server validate each other's identity through the access point, and then achieve mutual authentication as well as session key agreement. Finally, they can communicate with each other securely.
- HARCI [17]: In HARCI, the wireless sink node first authenticates itself with the cloud healthcare server and establishes a corresponding session key. With the secure session key, the wireless sink node can communicate with the cloud healthcare server via a secure channel to request the patient node's CRP. Then, the wireless sink node communicates with the patient node to achieve a two-way authentication and a session key agreement. After that, the patient node sends its data to the wireless sink node through a secure channel. Finally, the wireless sink node and patient node generate their new CRPs and share them with the cloud healthcare server.

TABLE II
COMPARISON OF COMMUNICATION OVERHEAD

Metrics	$liteAuth$	PSLAP	HARCI
Number of Messages	7	4*	8
Energy Consumption (joule)	7.88×10^{-4}	4.50×10^{-4}	9.01×10^{-4}

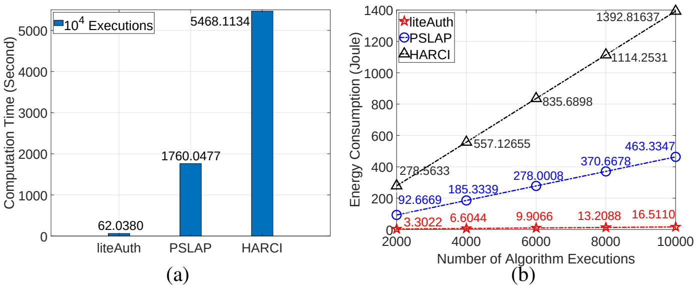
*In PSLAP, there is no message exchange for the identity verification of access point. In other words, the access point is assumed to be a fully trusted entity that relays messages between sensor node and hub server. However, if the sensor node and hub server first verified the identity of access point before exchanging any information, four more messages could be expected in PSLAP.

We measure the performance of $liteAuth$, PSLAP, and HARCI in terms of communication overhead, computation time, energy consumption, CPU time, and CPU cycles.

- Communication Overhead: Communication overhead is measured as the number of exchanged messages and energy consumption of communication in the network.
- Computation Time: Computation time is measured as the length of time spent performing all computations in the algorithm.
- Energy Consumption: Energy consumption is measured as the amount of electronic power consumed during the process of running algorithm.
- CPU Time: CPU time is the amount of time for which the CPU is used for processing instructions of the algorithm.
- CPU Cycles: CPU cycles is the number of clock cycles which is measured as the number of electronic pulses during running the algorithm.

B. Experimental Results and Analysis

First, we measure the communication overhead in terms of the number of exchanged messages and energy consumption of communication in Table II. We directly count the number of exchanged messages for *liteAuth*, PSLAP, and HARCI. The energy consumption of communication is calculated based on the number of sent and received messages [37]. To mutually authenticate sensor node and hub server, PSLAP requires four messages to be exchanged among sensor node, access point, and hub server. To be specific, the sensor node first sends an authentication request message to the access point. Then, the access point adds its identity in the message and forwards the message to the hub server. After the hub server verifies the identity of sensor node, it replies an authentication confirmation message through the access point to the sensor node. Finally, the sensor node and hub server mutually authenticate each other and establish a session key for further



(a) (b)

Fig. 7. Comparison of computation time and energy consumption.

communication. Please note that the sensor node and hub server do not verify the identity of access point before sending any information in PSLAP. In other words, the access point is a fully trusted entity that relays messages between sensor node and hub server. However, this is an unrealistic assumption in the current cyber threat environment. If the sensor node and hub server first verified the identity of access point before sharing any information, four more messages could be expected in PSLAP. In HARCI, the patient node initiates the authentication process by sending a message to wireless sink node. Then, the wireless sink node and cloud healthcare server authenticate each other and establish a session key through exchanging three messages. After that, the cloud healthcare server sends the CRP pair of patient node to the wireless sink node via a secure channel. Next, the patient node and cloud healthcare server exchange two messages for the establishment of session key. Finally, the wireless sink node sends a message to cloud healthcare server to update patient node's CRP pair. In $liteAuth$, as shown in Fig. 4, only seven messages are required to achieve mutual authentication and secure session key agreement between wireless medical device and control node, and between control node and cloud server. In addition, the energy consumption of communication for $liteAuth$, PSLAP, and HARCI is 7.88×10^{-4} joule, 4.50×10^{-4} joule, and 9.01×10^{-4} joule, respectively. It seems that PSLAP has a lower energy consumption than our approach $liteAuth$ according to the results presented in Table II. However, we argue that this amount of conserved energy is achieved by sacrificing security in PSLAP.

Second, we measure the computation time of $liteAuth$, PSLAP, and HARCI and present the results in Fig. 7(a). It is clear that the lowest computation time is achieved by our scheme $liteAuth$ compared to PSLAP and HARCI. In $liteAuth$, lightweight cryptographic operations, e.g., one-way hash function, Tinkerbell map-based random shuffling, and bitwise XOR operation, are adopted to achieve mutual authentication and session key agreement among entities. Those lightweight operations can be performed quickly, thus, a lower computation time is obtained by $liteAuth$. In PSLAP, one-way hash function and bitwise XOR operation are also employed. However, PSLAP repeatedly executes hash function and bitwise XOR operation, as a result, a longer computation time is measured. For example, hash function and bitwise XOR operation are executed ten and fourteen times in PSLAP, respectively. HARCI delivers the highest computation time. In HARCI, the patient node and wireless sink node perform a mutually agreed-upon public key expansion on the response generated from the PUF. In addition, the key expansion function is also used to generate multiple subkeys to be used in HARCI. As a result, a higher computation time is obtained by HARCI, compared to that of $liteAuth$ and PSLAP.

TABLE III
*COMPARISON OF CPU TIME AND CPU CYCLES

Metrics	$liteAuth$	PSLAP	HARCI
CPU Time	62015.63 ms	1740296.88 ms	5231453.13 ms
CPU Cycles	8.93×10^{10}	2.51×10^{12}	7.53×10^{12}

*The results are measured based on 10^4 algorithm executions.

Third, we measure the energy consumption of algorithm execution for $liteAuth$, PSLAP, and HARCI in Fig. 7(b). Overall, as the number of algorithm executions increases, the energy consumption of three schemes increase linearly. This is because the same operations are executed more times, as a result, more energy is consumed by the algorithm. Clearly, the highest energy consumption belongs to HARCI since the heavyweight operation (i.e., key expansion operation) is being used repeatedly in HARCI. Both $liteAuth$ and PSLAP consume less amount of energy as the number of algorithm executions increases. However, our scheme $liteAuth$ still outperforms PSLAP. In $liteAuth$, mutual authentication and session key agreement are achieved by executing lightweight operations, thus, lower energy consumption is obtained.

Fourth, we obtain the CPU time and CPU cycles of $liteAuth$, PSLAP, and HARCI, and the results are presented in Table III. As opposed to computation time, CPU time does not include waiting for input/output (I/O) operations or entering low-power (idle) mode. For the total of 10^4 algorithm executions, the CPU time of $liteAuth$, PSLAP, and HARCI is 62015.63 ms, 1740296.88 ms, and 5231453.13 ms, respectively. As expected, our scheme $liteAuth$ has the lowest CPU time. Regarding CPU cycles, 8.93×10^{10}, 2.51×10^{12}, and 7.53×10^{12} are measured for $liteAuth$, PSLAP, and HARCI, respectively. It is clearly shown that our scheme $liteAuth$ outperforms PSLAP and HARCI.

VIII. DISCUSSION

In a temperature-fluctuating and/or boisterous environment, the challenge-specific output of PUF, which is the response, becomes very unstable and tiny difference in the responses can be expected even though the identical challenge is fed into the same PUF. In other words, PUFs are widely believed to be non-noise-resistant [38]. Therefore, the security schemes which are designed based on the unique output of PUF might not be able to re-produce the exact same critical information [39]. Thus, in this paper, an ideal and noise-resistant PUF is assumed to be equipped with the control node and wireless medical device.

As a future work, we plan to resolve this important issue through designing and developing a fuzzy extractor and an error-correcting technique. First, we will design a response generation function which will produce a tuple consisting of the PUF response and a helper string. Here, the helper string will be fed into the PUF along with the original PUF challenge to re-produce the PUF response. In addition, we will design a response restore function, where the same PUF response can be re-produced with the value of helper string, error correcting code, as well as the original PUF challenge.

IX. Conclusion

In this paper, we proposed a lightweight and anonymous authentication and key agreement protocol ($liteAuth$) for WBANs, where mutual authentication and session key agreement are achieved using Tinkerbell map-based random shuffling, physical unclonable function, one-way hash function, and bitwise XOR operation. In $liteAuth$, the wireless medical device and control node first authenticate each other and establish a session key with the assistance of cloud server, and then communicate through a secure channel. We also verified the security of $liteAuth$ using the AVISPA tool and provided a security analysis of $liteAuth$. Our security verification and analysis demonstrated that $liteAuth$ is a secure protocol that can defend against many well-known security attacks. In addition, we developed a real-world testbed, implemented $liteAuth$ and other benchmark schemes, and conducted experiments for performance evaluation and comparison. Experimental results showed that $liteAuth$ has better performance in terms of communication overhead, computation time, energy consumption, CPU time, and CPU cycles, which indicates $liteAuth$ is a viable and competitive approach for ensuring security and data privacy in WBANs. In the future, we plan to integrate $liteAuth$ with blockchain technique and develop a secure data collection and storage mechanism for WBANs, where the cloud server will pack the collected data into blocks and compete to add its blocks into the blockchain.

References

[1] X. Shao, W. Liu, Y. Li, H. Chaudhry, and X. Yue, "Multistage implementation framework for smart supply chain management under industry 4.0," *Elsevier Technological Forecasting and Social Change*, vol. 162, p. 120354, 2021.

[2] M. Kumar and S. Chand, "A Provable Secure and Lightweight Smart Healthcare Cyber-Physical System With Public Verifiability," *IEEE Systems Journal*, pp. 1–1, 2021.

[3] A. Nair and S. Tanwar, "Fog Computing Architectures and Frameworks for Healthcare 4.0," in *Fog Computing for Healthcare 4.0 Environments*, 2021, pp. 55–78.

[4] *US Healthcare Industry in 2021*, Last accessed: Dec 23, 2021, https://www.insiderintelligence.com/insights/healthcare-industry/.

[5] *Human Lifespan*, Last accessed: Dec 23, 2021, https://www.cnbc.com/2019/05/08/techs-next-big-disruption-could-be-delaying-death.html.

[6] A. Vijayalakshmi, D. Jose, and S. Unnisa, "Wearable Sensors for Pervasive and Personalized Health Care," *Springer IoT in Healthcare and Ambient Assisted Living*, pp. 123–143, 2021.

[7] M. Hajar, M. Al-Kadri, and H. Kalutarage, "A survey on wireless body area networks: architecture, security challenges and research opportunities," *Elsevier Computers & Security*, p. 102211, 2021.

[8] *Medtronic*, Last accessed: Dec 23, 2021, https://www.medtronic.com/us-en/index.html.

[9] *Medtronic Conexus Radio Frequency Telemetry Protocol (Update B)*, Last accessed: Dec 23, 2021, https://us-cert.cisa.gov/ics/advisories/ICSMA-19-080-01.

[10] *CVE-2019-6538 Detail*, https://nvd.nist.gov/vuln/detail/CVE-2019-6538.

[11] W. Stallings, *Cryptography and Network Security - Principles and Practices, 8th Edition*. Pearson, 2019.

[12] L. Wu, X. Du, M. Guizani, and A. Mohamed, "Access Control Schemes for Implantable Medical Devices: A Survey," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1272–1283, 2017.

[13] *Automated Validation of Internet Security Protocols and Applications*, http://www.avispa-project.org/.

[14] *Laptop Computers*, Last accessed: July 26, 2021, https://deals.dell.com/en-us/productdetail/9u7v.

[15] *Latte Panda*, Last accessed: July 26, 2021, https://www.lattepanda.com/.

[16] B. Alzahrani, A. Irshad, A. Albeshri, and K. Alsubhi, "A Provably Secure and Lightweight Patient-Healthcare Authentication Protocol in Wireless Body Area Networks," *Wireless Personal Communications*, vol. 117, no. 1, pp. 47–69, 2021.

[17] T. Alladi, V. Chamola, and Naren, "HARCI: A Two-Way Authentication Protocol for Three Entity Healthcare IoT Networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 361–369, 2021.

[18] M. Kumar and S. Chand, "A Lightweight Cloud-Assisted Identity-Based Anonymous Authentication and Key Agreement Protocol for Secure Wireless Body Area Network," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2779–2786, 2021.

[19] A. Gupta, M. Tripathi, and A. Sharma, "A provably secure and efficient anonymous mutual authentication and key agreement protocol for wearable devices in WBAN," *Elsevier Computer Communications*, vol. 160, pp. 311–325, 2020.

[20] K. Park, S. Noh, H. Lee, A. Das, M. Kim, Y. Park, and M. Wazid, "LAKS-NVT: Provably Secure and Lightweight Authentication and Key Agreement Scheme Without Verification Table in Medical Internet of Things," *IEEE Access*, vol. 8, pp. 119 387–119 404, 2020.

[21] M. Fotouhi, M. Bayat, A. Das, H. Far, S. Pournaghi, and M. Doostari, "A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT," *Computer Networks*, vol. 177, p. 107333, 2020.

[22] K. Kaur, S. Garg, G. Kaddoum, and M. Guizani, "Secure Authentication and Key Agreement Protocol for Tactile Internet-based Tele-Surgery Ecosystem," in *Proc. IEEE ICC*, 2020, pp. 1–6.

[23] G. Mwitende, I. Ali, N. Eltayieb, B. Wang, and F. Li, "Authenticated key agreement for blockchain-based WBAN," *Springer Telecommunication Systems*, vol. 74, no. 3, pp. 347–365, 2020.

[24] R. Hajian, S. ZakeriKia, S. Erfani, and M. Mirabi, "SHAPARAK: Scalable healthcare authentication protocol with attack-resilience and anonymous key-agreement," *Computer Networks*, vol. 183, p. 107567, 2020.

[25] M. Umar, Z. Wu, and X. Liao, "Mutual Authentication in Body Area Networks Using Signal Propagation Characteristics," *IEEE Access*, vol. 8, pp. 66 411–66 422, 2020.

[26] T. Wan, L. Wang, W. Liao, and S. Yue, "A lightweight continuous authentication scheme for medical wireless body area networks," *Peer-to-Peer Networking and Applications*, vol. 14, no. 6, pp. 3473–3487, 2021.

[27] A. Goldsztejn, W. Hayes, and P. Collins, "Tinkerbell is Chaotic," *SIAM Journal on Applied Dynamical Systems*, vol. 10, no. 4, pp. 1480–1501, 2011.

[28] H. Nusse and J. Yorke, *Dynamics: Numerical Explorations*. Springer, 2012.

[29] A. Shamsoshoara, A. Korenda, F. Afghah, and S. Zeadally, "A survey on physical unclonable function (PUF)-based security solutions for Internet of Things," *Computer Networks*, vol. 183, p. 107593, 2020.

[30] J. Wallrabenstein, "Practical and Secure IoT Device Authentication Using Physical Unclonable Functions," in *Proc. IEEE FiCloud*, 2016, pp. 99–106.

[31] K. Chuang, E. Bury, R. Degraeve, B. Kaczer, D. Linten, and I. Verbauwhede, "A Physically Unclonable Function Using Soft Oxide Breakdown Featuring 0% Native BER and 51.8 fJ/bit in 40-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 10, pp. 2765–2776, 2019.

[32] *Automated Validation of Internet Security Protocols and Applications*, Last accessed: July 26, 2021, http://www.avispa-project.org/.

[33] *SPAN*, Last accessed: July 26, 2021, http://www.avispa-project.org/.

[34] *VirtualBox*, Last accessed: July 26, 2021, https://www.virtualbox.org/.

[35] *LiClipse*, Last accessed: July 26, 2021, https://www.liclipse.com/.

[36] *Java Security Standard Algorithm Names*, Last accessed: Oct 30, 2021, https://docs.oracle.com/en/java/javase/12/docs/specs/security/standard-names.html.

[37] C. Pu and S. Lim, "A Light-Weight Countermeasure to Forwarding Misbehavior in Wireless Sensor Networks: Design, Analysis, and Evaluation," *IEEE Systems Journal*, vol. 12, no. 1, pp. 834–842, 2018.

[38] C. Pu and Y. Li, "Lightweight Authentication Protocol for Unmanned Aerial Vehicles Using Physical Unclonable Function and Chaotic System," in *IEEE LANMAN*, 2021, pp. 1–6.

[39] C. Pu, A. Wall, K. Choo, I. Ahmed, and S. Lim, "A Lightweight and Privacy-Preserving Mutual Authentication and Key Agreement Protocol

for Internet of Drones Environment," *IEEE Internet of Things Journal (Early Access)*, pp. 1–1, 2022.