Security-Enhanced Lightweight and Anonymity-Preserving User Authentication Scheme for IoT-Based Healthcare

Xin Zhou, Shengbao Wang, Kang Wen, Bin Hu, Xiao Tan, and Qi Xie

***Abstract*—Ensuring trust within the healthcare system and addressing privacy and security challenges in the Internet of Medical Things (IoMT) is of paramount importance. Based on our preliminary analysis results of Masud et al.'s authentication protocol, we propose an improved solution building upon their protocol. Our improved protocol incorporates various security measures to enhance its security. To validate the effectiveness of our improved protocol, we employ a comprehensive range of heuristic and formal security analysis methods. Comparative evaluations with other relevant protocols reveal that our proposed solution achieves satisfactory operational performance in resource-constrained IoMT scenarios.**

***Index Terms*—Authenticated key exchange, BAN logic, fuzzy extractor, Internet of Medical Things (IoMT), physical unclonable function (PUF), ProVerif.**

I. Introduction

THE Internet of Medical Things (IoMT) is the convergence of medical devices and applications that can be connected to healthcare information technology systems using network technologies [1]. The development of IoMT has been driven by the widespread use of wireless medical sensor networks (WMSNs) in the healthcare sector over the past few years. The development of IoMT has been driven by the widespread use of WMSNs in the healthcare field over the past few years [2]. In IoMT scenarios such as this, a variety of sophisticated sensor devices are placed on patients to collect and monitor their physiological parameters without compromising their comfort and transmit the data wirelessly to doctors' handheld devices, such as tablets, smartphones, and other devices. Based on these data, the doctor can assess the patient's health status more comprehensively. Although all data are collected from harmless wearable devices, these physiological data are very sensitive and can pose serious privacy concerns [3], [4], [5]. Additionally, most electronic devices communicate via wireless networks, which are vulnerable to a variety of attacks. Therefore, authentication mechanisms and data protection means are essential to secure wireless medical sensor networks.

Manuscript received 4 August 2023; revised 10 September 2023; accepted 5 October 2023. Date of publication 11 October 2023; date of current version 7 March 2024. This work was supported by the National Natural Science Foundation of China under Grant U21A20466. *(Corresponding author: Shengbao Wang.)*

Xin Zhou is with the School of Information Science and Technology and the Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China, and also with the School of Mathematics and Statistics, Fujian Normal University, Fuzhou 350007, China.

Shengbao Wang, Kang Wen, Bin Hu, and Xiao Tan are with the School of Information Science and Technology and the Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China (e-mail: shengbaowang@hznu.edu.cn).

Qi Xie is with the School of Mathematics and the Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China.

Digital Object Identifier 10.1109/JIOT.2023.3323614

A. Related Work

1) Two-Party Protocols: Existing security mechanisms focus on mutual authentication between two parties. Iqbal and Bayoumi [6] proposed an end-to-end authentication protocol for authenticating resource-constrained IoMT devices. The protocol uses a Diffie–Hellman key establishment scheme and outsources the heavy computational tasks to a trusted computing center. Park et al. [7] introduced a lightweight, provably secure scheme for IoMT healthcare systems. They designed a secure authentication protocol between sensor entities and servers using nonverification table (NVT) technology. Amin et al. [8] proposed a three-factor (password, biometric, and smart card) user authentication scheme for e-health systems and claimed that their scheme is resistant to most common attacks.

Fan et al. [9] applied radio frequency identification (RFID) technology to healthcare systems and proposed a lightweight mutual authentication protocol with privacy-preserving properties. Aghili et al. [10] proposed an enhanced authentication protocol called SecLAP to prevent the attacks in [9] by reducing the traffic between the tag and the reader. However, the single-tag authentication protocol suffers from high latency and low efficiency and is not suitable for large healthcare scenarios where a large number of tags exist. Therefore, Kang et al. [11] proposed a low-overhead batch-tag authentication protocol. The protocol uses the encryption of tags using the solution of homogeneous linear equations as a key to reduce the tag overhead.

2) TTP-Based Multiparty Protocols: However, IoT-enabled healthcare applications involving multiple parties, such as patients, electronic healthcare testing devices, doctors, and cloud servers, require multiparty authentication for secure communication. Some schemes use trusted third party (TTP) to identify and authenticate communicating parties to reduce the higher computational and communication costs of resource-constrained devices. Mahmood et al. [5] proposed

an authentication protocol based on elliptic curve ciphers. The protocol can generate symmetric security keys between the patient, doctor, or nurse and a trusted server to ensure secure communication between the patient and the doctor after the authentication is over. Yanambaka et al. [12] proposed a device authentication protocol that authenticates devices in a network without storing the information in memory. At the same time, the protocol uses a physical unclonable function (PUF) to provide a unique identity for each device and authenticate it when transmitting data to the server via a TTP. Tsai et al. [13] proposed a multikey exchange protocol based on trusted third parties. It uses 2-D operations, elliptic curve cryptography, and current time encryption keys to exchange respective session keys. The protocol can generate 40 keys simultaneously to support 40 sessions at the same time. However, the authors have yet to provide a formal proof of the scheme and its use in resource-constrained IoT sensors is yet to be evaluated.

3) SG-Based Multiparty Protocols: Other schemes delegate the authentication process to smart gateways of neighboring participants. Wu et al. [14] proposed a lightweight authentication protocol based on WSNs. The protocol mainly uses hash functions and XOR operations and is distinguished by its high efficiency and low overhead. Similarly, the protocol proposed by Masud et al. [15] also has lightweight features. Kwon et al. [16] noted that the scheme is not anonymous and suffers from privileged insider attacks and suggested an improved scheme. However, it is noteworthy that they did not utilize the Diffie–Hellman key exchange, thus lacking the forward security property explicitly stated in the scheme. An improved protocol with high efficiency was also proposed by Kim et al. [17] in 2023. However, this improved protocol cannot avoid the problems of [15], and it lacks security. Recently, Shihab and AlTawy [18] proposed a lightweight authentication protocol that is robust to desynchronization attacks based on the one-way hash chain technique. However, the scheme is not resistant to physical attacks or cloning attacks.

Masud et al.'s protocol [15] is characterized by the utilization of hash and XOR operations, rendering it a lightweight solution with lower computational and communication overheads. However, our previous analysis [19] revealed two design flaws within Masud et al.'s protocol. And we also demonstrated that it is susceptible to session key disclosure attacks, offline password guessing attacks, and traceability attacks. While we did provide remarks pertaining to each of these identified security vulnerabilities, a comprehensive solution was not offered.

To fill this gap, in this article, we further present a complete improved protocol based on our previous work. More importantly, we conduct an in-depth analysis of the improved protocol, including its security and operational efficiency. The results of the analysis show that the new improved protocol not only eliminates the original defects of their protocol but also retains the advantages of being lightweight and efficient.

B. Contribution

The contributions of this article are as follows.

1) In light of our prior analysis of the Msaud et al.'s protocol [15] and our subsequent proposal for its enhancement, this article introduces a lightweight authentication protocol. This protocol primarily relies on hash functions and XOR operations to facilitate authentication and key exchange between resource-constrained IoMT devices.
2) To significantly improve the security of our protocol while preserving excellent performance, we adopt a strategic blend of cryptographic tools and techniques. These include biometrics, fuzzy extractors, secret salts, and physically unclonable functions. By leveraging these advanced measures in tandem, our approach ensures a robust defense against potential threats while maintaining efficient and seamless operation. This comprehensive security enhancement contributes to the protocol's reliability and trustworthiness in real-world applications.

C. Organization

The remainder of this article is structured as follows. Section II describes security mechanisms, such as hash functions, physically unclonable functions, and fuzzy extractors, as well as common security properties in authentication protocol design. We review the flaws and insecurities of Masud et al.'s protocol in Section III. Section IV presents our modified protocol. Both heuristic and formal security analyses of the protocol are given in Section V. Section VI presents a security and performance comparison among our protocol and other related protocols. Finally, we conclude this article in Section VII.

II. Preliminaries

Our proposed protocol utilizes hash functions, PUFs, and fuzzy extractors. The definitions and properties are given in this section, respectively.

A. Hash Function

The one-way hash function $H(\cdot)$ generates a fixed-length output (e.g., l bits) from an input of arbitrary length, defined as $H : \{0,1\}^* \rightarrow \{0,1\}^l$. Its security properties are as follows [20], [21].

1) *Preimage Resistance:* For a given hash value h and a one-way hash function $H(\cdot)$, it is difficult to find any preimage m such that $h = H(m)$.
2) *Second Preimage Resistance:* For a given preimage a, it is difficult to find another preimage b such that $H(a) = H(b)$.
3) *Strong Collision Resistance:* For the same one-way hash function $H(\cdot)$, it is difficult to find two different inputs such that $H(a) = H(b)$.

B. Physical Unclonable Function

Physically unclonable functions [22] are physical entities embedded in a physical structure [e.g., an integrated chip (IC)] which, by means of $R \leftarrow \text{PUF}(C)$, can generate a corresponding output (response) for a given input (challenge). The PUF provides an efficient way to authenticate devices and protect

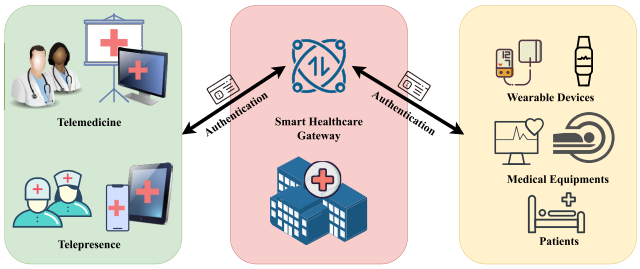

Fig. 1. System architecture.

them from physical threats in insecure environments. In other words, the PUF(·) can act as a digital fingerprint of the device, with the following security properties.

1) *Uniqueness:* For any two different PUFs, $\text{PUF}_1(\cdot)$ and $\text{PUF}_2(\cdot)$, given the same input C, the outputs $R_1 \leftarrow \text{PUF}_1(C)$ and $R_2 \leftarrow \text{PUF}_2(C)$ are different.
2) *Reproducibility:* The response is the same for any given PUF, given the same input, multiple times.
3) *Physical Unclonability:* Due to the physical structure and technical reasons, a specific PUF embedded in a device cannot be cloned.
4) *One-Wayness:* PUFs are analogous to one-way functions in cryptography, that is, given a response R and a particular PUF(·), it is difficult to recover the corresponding challenge C.

C. Fuzzy Extractor

A fuzzy extractor is a method for converting a noisy nonuniform input into a uniform random string that can be reliably reproduced [23]. It is important to note that "fuzzy" in this context means that the fixed values required for encryption are extracted from values close to, but not equal to, the original key, without compromising the required security. The fuzzy extractor consists of a generating function $Gen(\cdot)$ and a recurrence function $Rep(\cdot)$, described in detail as follows.

1) *Generate Function $Gen(\cdot)$:* Given the input biometrics bio of users, the function generates the secret data δ and the common auxiliary parameter τ, where $(\delta, \tau) = Gen(bio)$.
2) *Reproduction Function $Rep(\cdot)$:* Given the user input bio, the function reproduces the secret data δ using the auxiliary parameter τ, where $\delta = Rep(bio, \tau)$.

D. System Architecture

Our proposed protocol focuses on describing the communication between users (i.e., doctors), medical gateways, and medical sensors in a smart healthcare scenario. Fig. 1 illustrates the system architecture of the protocol, where the user typically uses a resource-constrained handheld device (e.g., smartphone and tablet), the sensor nodes are mainly low-power implantable devices, and the medical gateway typically has powerful computational capabilities to perform complex computations.

E. Threat Model

In this article, the classical Dolev–Yao (DY) model [24] is used to evaluate the security of protocols. The model assumes that the cryptographic primitives used in the protocol are secure and that an attacker can intercept, tamper with, delete, store, and replay any message from the open channel. An attacker can only decrypt or sign a message if he has the correct key. An attacker can only forge new messages from the keys and messages in his possession [25].

F. Common Security Attributes

In this section, we describe common security properties for authentication protocols in detail.

1) Resistant to Offline Password Guessing Attacks: An offline password guessing attack occurs in an offline environment, where an attacker uses brute-force cracking tools, password dictionaries, and similar methods to offline deduce the legitimate user's password. In contrast to online guessing attacks, offline attacks have the characteristic of being able to make an unlimited number of attempts. When calculating and storing the hash value of a password, the addition of a secret salt value can increase the difficulty for an attacker to guess the password, as the attacker must now simultaneously guess both the password and the secret salt value.

2) Resistant to Session Key Compromise Attacks: A session key compromise attack refers to a situation where an attacker is able to obtain the session key. The security of the session key is a crucial prerequisite for secure communication between entities. When the attacker successfully determines the key, they can use it to decrypt encrypted data without the sender's knowledge.

3) Anonymity and Untraceability: Anonymity, in the context of information security and privacy, signifies the condition wherein an attacker is precluded from ascertaining the authentic identity of a communication participant solely through the examination of the transmitted message [26]. Untraceability, on the other hand, denotes the state in which an attacker encounters insurmountable challenges in determining both the specific user responsible for sending a message and any potential linkage between multiple messages, thus obfuscating any discernible connection among them [27].

4) Resistant to Cloning and Physical Attacks: Physical and cloning attacks encompass scenarios where the attackers endeavors to expropriate the confidential information of an authorized node by physical means, including techniques, such as side-channel analysis, power consumption analysis, and related methods. Afterward, the attacker attempts to establish a fraudulent communication node, replete with spurious credentials and identity attributes, with the aim of infiltrating the network under the guise of an authorized node. PUFs are commonly used in authentication protocols to resist cloning attacks [28].

5) Resistant to Impersonation Attacks: Impersonation attacks refer to situations in which attackers have the ability to assume the identity of legitimate users. Attackers may carry out this attack through various means, including eavesdropping, message manipulation, replicating and replaying

messages, or exploiting vulnerabilities introduced by physical and cloning attacks.

In addition, we also make the following assumptions about the attacker's ability.

1) During the brief device authentication process, the transient (erased or overwritten when in use) intermediate variables involved in the session state calculation are safe.
2) The server can provide expensive tamper-proof storage to protect any secrets and is powerful enough to perform complex calculations.
3) In contrast, end devices have limited storage capacity to protect the secrets stored in their nonvolatile memory. As a result, they are vulnerable to capture.

III. Deficiencies and Insecurity of Masud et al.'s Scheme

The issues with the Masud et al.'s protocol [15] have been analyzed in our previous work [19]. To save space, we omit a review of Masud et al.'s protocol and its attacks, and only briefly present our analysis results and improvement suggestions. Afterward, we will provide our complete improved protocol based on these improvement suggestions. Any interested reader is referred to [15] and [19] for further details of the original protocol and our attacks against it.

A. Two Implementation Issues

The original protocol exhibited deficiencies in its implementation by neglecting to transmit the temporary identity of sensor nodes S_{TID} at the commencement of the protocol, as well as failing to update the temporary identity of users D_{TID}^{new} upon the completion of authentication. As a result, the protocol's proper functioning was compromised. To address these limitations, the revised protocol is introduced, incorporating the inclusion of the *SID* field in the initial message and ensuring the timely update of DID^{new} and SID^{new} following a successful authentication process.

B. Session Key Disclosure and Traceability Attacks

The susceptibility of the original scheme to session key disclosure attacks and traceability attacks stems from the attacker's capability to acquire confidential data from sensor nodes through side-channel attacks. To address this fundamental flaw, we opt to integrate a PUF into the protocol in order to reinforce its security measures.

C. Offline Password Guessing Attacks

In order to enhance security and mitigate the risks associated with offline password guessing, we adopt a biometric authentication approach, replacing traditional passwords. Additionally, we incorporate secret salts into the system, further increasing the complexity and difficulty of offline password guessing attacks.

IV. Our Enhanced Scheme

Based on Masud et al.'s protocol, we propose a more secure and efficient anonymous authentication exchange protocol. The protocol enables proper agreement of session keys between users (i.e., doctors), medical gateways, and medical sensors in smart healthcare scenarios, which in turn ensures secure communication. It has three phases named *the user registration phase*, *the sensor node registration phase*, and *the authentication and key exchange phase*. The symbolic descriptions in the protocol are shown in Table I.

TABLE I
Notation Table

Symbol	Details
ID_i	Identity of the i^{th} user
BIO_i	Biometrics of the i^{th} user
DID_i	Temporary identity of the i^{th} user
SN_n	Identification code for the n^{th} sensor
SID_n	Temporary identity of the n^{th} sensor
r, b	Random number
$\langle C_i, R_i \rangle$	The *Challenge-Response* pair of PUF
$Gen(\cdot)$	Fuzzy extractor obfuscation function
$Rep(\cdot)$	Fuzzy extractor reproduction function
SK	Session key

A. User Registration Phase

Users (doctors) must register their devices with the medical gateway in order to obtain real-time patient health data. The steps for user registration are as follows.

Step 1: The user enters his identity information ID_i and biometric BIO_i, and the smart device generates $(k_i, hid_i) = Gen(BIO_i)$ through a fuzzy extractor and sends the registration request $Req : \{ID_i, k_i\}$ to the medical gateway through a secure channel.

Step 2: After receiving a registration request, the medical gateway traverses its registration list and determines whether the user is a new user. The gateway then generates a random number b_i and a pseudonym $DID_i = b_i \oplus ID_i$ for the user. The gateway stores it in its registry while sending DID_i to the user.

Step 3: Upon receipt of the message, the user device generates a random number r_i of short length (e.g., 8 bits) and calculates $CPW_i = h(k_i \| ID_i \| r_i)$ and stores ID_i, CPW_i, hid_i, and DID_i in its memory. After completing the registration, the user binds the registered sensor nodes by other means and stores their related information. At this point, the user device contains $\langle ID_i, CPW_i, hid_i, DID_i, SID_n \rangle$, etc.

B. Sensor Node Registration Phase

Step 1: The sensor node identifies its own unique identity SN_n, and sends it to the gateway on a secure channel.

Step 2: Once the registration request has been received, the gateway iterates through its registration list and determines whether or not it is a registered node. It then generates a random number b_n and calculates $SID_n = b_n \oplus SN_n$, while generating a challenge C_n and sending the message $\{SID_n, C_n\}$ to the sensor node over a secure channel.

Step 3: The sensor node generates the response $R_n \leftarrow PUF(C_n)$ via the PUF function. The gateway stores the SID_n in its memory and sends the response R_n to the gateway.

Step 4: The gateway records related information $\langle SN_n, SID_n, (C_n, R_n), b_n\rangle$ in its registration list.

C. Authentication and Key Exchange Phase

As participants communicate primarily through unreliable channels, it is necessary for participants to authenticate each other and negotiate session keys. This section describes in detail the process of participant authentication and key exchange, as shown in Fig. 2.

$U \rightarrow GW$: A fuzzy extractor recovers $k_i = Rep(hid_i, BIO_i)$ from the user's identity ID_i and biometric BIO_i. The smart device continuously tries a random number r_i of shorter length, computes $CPW_i' = h(k_i \| ID_i \| r_i)$, and checks whether $CPW_i' = CPW_i$ holds within a finite number of times (no more than $2^{|r_i|}$). If it is not the case, the device stops the execution of the protocol; otherwise, it proceeds to the next steps. The device generates the random number b_i^{new} and calculates $N_i = b_i^{\text{new}} \oplus h(k_i)$. It then generates the verification message $\alpha = h(b_i^{\text{new}} \| k_i \| DID_i \| SID_n)$. The user device sends the message $M_1 : \{N_i, \alpha, DID_i, SID_n\}$ through the open channel.

$GW \rightarrow SN$: After receiving a message from the user U_i, the gateway retrieves its registry and locates the registration information k_i, ID_i and $\langle C_n, R_n\rangle$, SN_n for DID_i and SID_n. It then computes $b_i^{\text{new}\prime} = N_i \oplus h(k_i)$, $\alpha' = h(b_i^{\text{new}\prime} \| k_i \| DID_i \| SID_n)$ and verifies whether $\alpha' = \alpha$ holds or not. After verification, the gateway generates a random number b_n^{new} and the session key SK. Then, it computes $SID_n^{\text{new}} = SN_n \oplus b_n^{\text{new}}$, $SK_n = (SK \| SID_n^{\text{new}}) \oplus h(R_n)$ and generates the verification message $\beta = h(SK \| R_n \| SID_n \| SID_n^{\text{new}})$. The gateway sends the message $M_2 : \{SK_n, \beta, C_n\}$ through the open channel.

$SN \rightarrow GW$: After receiving the challenge C_n, the sensor node outputs the response $R_n \leftarrow PUF(C_n)$ and calculates $(SK' \| SID_n^{\text{new}\prime}) = SK_n \oplus h(R_n)$ and $\beta' = h(SK' \| R_n \| SID_n \| SID_n^{\text{new}\prime})$. If $\beta' = \beta$ does not hold, the sensor node terminates the protocol and deletes the session key SK' and the new pseudonym $SID_n^{\text{new}\prime}$; conversely, the sensor node accepts the session key SK', stores the new pseudonym $SID_n^{\text{new}\prime}$ and continues with the protocol. The sensor node generates the authentication message $\gamma = h(SID_n^{\text{new}\prime} \| SK')$ and sends the confirmation message $M_3 : \{\gamma\}$.

$GW \rightarrow U$: The gateway calculates $\gamma' = h(SID_n^{\text{new}} \| SK)$. If $\gamma' = \gamma$ does not hold, the gateway terminates the protocol; if it does, the gateway replaces and continues the protocol with the registration record for that sensor. Gateway computes $DID_i^{\text{new}} = ID_i \oplus b_i^{\text{new}}$ and $SK_i = (SID_n^{\text{new}} \| SK \| DID_i^{\text{new}}) \oplus h(k_i)$. Also generate the verification message $\lambda = h(SK \| DID_i \| k_i \| DID_i^{\text{new}} \| SID_n^{\text{new}})$ and send the message $M_4 : \{SK_i, \lambda\}$. Then, the gateway adds DID_i^{new} to the registration record of that user.

U: User calculates $(SID_n^{\text{new}\prime} \| SK' \| DID_i^{\text{new}\prime}) = SK_i \oplus h(k_i)$ and $\lambda' = h(SK' \| DID_i \| k_i \| DID_i^{\text{new}\prime} \| SID_n^{\text{new}\prime})$. If $\lambda' = \lambda$ does not hold, the user terminates the protocol and rejects SK' and $DID_i^{\text{new}\prime}$; conversely, the user accepts SK' and replaces DID_i and SID_n stored in memory with $DID_i^{\text{new}\prime}$ and $SID_n^{\text{new}\prime}$.

Note that during the functioning of the protocol, the participants send temporary identities over an open channel, and the updating of the temporary identities ensures that the protocol satisfies the security property of untraceability.

V. Security Analysis of Proposed Protocol

In this section, we will analyze the security of the improved protocol in detail, using both heuristic methods and formal methods, specifically the BAN logic [29] and the ProVerif tool [30].

Heuristic analysis examines protocol security in the most intuitive and direct way, particularly by scrutinizing all fundamental security properties. In contrast, BAN logic focuses on the analysis of mutual authentication within the protocol. Finally, the ProVerif tool provides a rigorous verification of security properties by leveraging π-calculus. This verification process strengthens the validation of the analyses conducted by the first two methods.

A. Heuristic Analysis

1) Resistant to Offline Password Guessing Attacks: In order o make it more difficult for an attacker to guess the password, the improved protocol is designed using biometrics, and a secret salt is added to the calculation of CPW_i. The user's smart device must generate an additional short (e.g., 8-bit) random number r_i to compute $CPW_i = h(k_i \| ID_i \| r_i)$. Note that the secret salt value r_i is not stored in the device memory and cannot be obtained by an attacker through a side-channel attack. When a legitimate user logs in to the device, he or she must try r_i a limited number of times, and because of the short length of the salt, it is very easy for a legitimate user who knows the password to pass authentication. An attacker who does not know the password has to guess both the password and the secret salt value, making it significantly more difficult to guess. Therefore, the new protocol maximizes resistance against offline password guessing attacks.

2) Resistant to Session Key Compromise Attacks: In Masud et al.'s scheme, an attacker can obtain temporary secret data through a side-channel attack and consequently compute the final session key. Therefore, the new protocol is specifically designed to address the threat posed by side-channel attacks. On the user side, a fuzzy extractor is added, and the secret data k_i is no longer statically stored but dynamically generated by the user entering the correct password. On the sensor node side, the new protocol utilizes a PUF to resist side-channel attacks. As a result, the new protocol ensures the security of the session key.

3) Anonymity and Untraceability: During the protocol registration phase, the real identity information of users and sensor nodes is hidden through $DID_i = ID_i \oplus b_i$ and $SID_n = SN_n \oplus b_n$. Moreover, during the protocol execution, pseudonyms are sent over the open channel, and without knowing b_i and b_n, an attacker eavesdropping on the communication cannot determine the communication participants corresponding to DID_i and SID_n. On the other hand, the pseudonyms

User	Gateway	Sensor Node
Input ID_i, BIO_i		
$k_i = Rep(hid, BIO_i)$		
$CPW_i' = h(k_i \| ID_i \| r_i)$		
Verify $CPW_i' \stackrel{?}{=} CPW_i$		
Generate b_i^{new}		
$N_i = b_i^{new} \oplus h(k_i)$		
$\alpha = h(b_i^{new} \| k_i \| DID_i \| SID_n)$		
$\xrightarrow[M_1]{\{N_i, \alpha, DID_i, SID_n\}}$		
	Locate DID_i, SID_n	
	Retrieve $k_i, ID_i, \langle C_n, R_n \rangle, SN_n$	
	$b_i^{new\prime} = N_i \oplus h(k_i)$	
	$\alpha' = h(b_i^{new\prime} \| k_i \| DID_i \| SID_n)$	
	Verify $\alpha' \stackrel{?}{=} \alpha$	
	Generate b_n^{new}, SK	
	$SID_n^{new} = SN_n \oplus b_n^{new}$	
	$SK_n = (SK \| SID_n^{new}) \oplus h(R_n)$	
	$\beta = h(SK \| R_n \| SID_n \| SID_n^{new})$	
	$\xrightarrow[M_2]{\{SK_n, \beta, C_n\}}$	
		$R_n \leftarrow PUF(C_n)$
		$(SK' \| SID_n^{new\prime}) = SK_n \oplus h(R_n)$
		$\beta' = h(SK' \| R_n \| SID_n \| SID_n^{new\prime})$
		Verify $\beta' \stackrel{?}{=} \beta$
		Accept SK' and store $SID_n^{new\prime}$
		$\gamma = h(SID_n^{new\prime} \| SK')$
	$\xleftarrow[M_3]{\{\gamma\}}$	
	$\gamma' = h(SID_n^{new} \| SK)$	
	Verify $\gamma' \stackrel{?}{=} \gamma$	
	Alt. SID_n as SID_n^{new}	
	$DID_i^{new} = ID_i \oplus b_i^{new}$	
	$SK_i = (SID_n^{new} \| SK \| DID_i^{new}) \oplus h(k_i)$	
	$\lambda = h(SK \| DID_i \| k_i \| DID_i^{new} \| SID_n^{new})$	
$\xleftarrow[M_4]{\{SK_i, \lambda\}}$		
$(SID_n^{new\prime} \| SK' \| DID_i^{new\prime}) = SK_i \oplus h(k_i)$		
$\lambda' = h(SK' \| DID_i \| k_i \| DID_I^{new\prime} \| SID_n^{new\prime})$		
Accept SK'		
Alt. DID_i, SID_n as $DID_i^{new\prime}, SID_n^{new\prime}$		

Fig. 2. Authentication and key exchange phase in the proposed protocol.

of the sensor nodes are updated after each protocol run, so an attacker cannot track the communication of participants through DID_i and SID_n. In conclusion, the new protocol provides both anonymity and untraceability.

4) Resistant to Cloning and Physical Attacks: Suppose an attacker could attempt to tamper with the memory of the sensor node SN_n or perform a side-channel attack to obtain the data stored in the memory [31]. However, such an attempt would change the function of the PUF and the adversary would obtain a PUF that could not produce any output, thus rendering the adversary's attempt meaningless. It is because of the nonreplicable nature of the PUF that makes this protocol resistant to cloning and physical attacks.

5) Resistant to Impersonation Attacks: Assuming that an attacker $\mathcal{A}$ intercepts a message $M_1 : \{N_i, \alpha, DID_i, SID_n\}$ sent by a user U_i to the gateway and then tries to impersonate U_i to resend the tampered message, it must recalculate k_i. However, k_i needs to be reproduced from the fuzzy extractor using the ID_i, BIO_i of the user and other information. The aforementioned analysis of anonymity and resistance to cloning attacks shows that the adversary has no access to these secret values and, therefore, the proposed protocol is resistant to impersonation attacks. In a similar manner, it can be inferred that the attacker also cannot successfully impersonate the sensor node.

B. BAN Logic-Based Authentication Proof

BAN logic is a logic rule proposed by Burrows et al. [29]. It is primarily used to analyze message interaction protocols and to assist the verifier in determining whether a message is trusted or eavesdropped. BAN logic plays an extensive and active role in the formal analysis of authentication protocols. In this section, we will use BAN logic to

TABLE II
BAN LOGIC NOTATIONS

Symbol	Details
$P\| \equiv X$	P believes a statement X
$P \triangleleft X$	P has received a message containing X
$P\| \sim X$	P once sent a message containing X
$\#X$	X is fresh
$P\| \Rightarrow X$	P has jurisdiction over X
$P \overset{K}{\longleftrightarrow} Q$	K is a good key shared to communicate between P and Q
$P \overset{X}{=\!=} Q$	X is a shared secret between P and Q
$\langle X \rangle_K$	A message connected by X and K , where K is a secret

TABLE III
BAN LOGIC RULES

Rule	Details	Expressions
R_1	Message Meaning Rule	$\frac{P\|\equiv Q \overset{K}{\longleftrightarrow} P, P \triangleleft \{X\}_K}{P\|\equiv Q\|\sim X}$
R_2	Jurisdiction Rule	$\frac{P\|\equiv Q\|\Rightarrow X, P\|\equiv Q\|\equiv X}{P\|\equiv X}$
R_3	Nonce Verification Rule	$\frac{P\|\equiv \#(X), P\|\equiv Q\|\sim X}{P\|\equiv Q\|\equiv X}$
R_4	Freshness Rule	$\frac{P\|\equiv \#(X)}{P\|\equiv \#(X,Y)}$
R_5	Belief Rules	$\frac{P\|\equiv (X), P\|\equiv (Y)}{P\|\equiv (X,Y)}$, $\frac{P\|\equiv Q\|\equiv (X,Y)}{P\|\equiv Q\|\equiv X}$

formally prove that the proposed protocols can achieve mutual authentication.

BAN logic semantics defines three types of objects. Communication subjects, i.e., the actual participants of the protocol, e.g., terminals, servers, etc. Logical statements, i.e., the expression of specific logic in specific symbols. And shared secrets, i.e., secret data agreed in advance between participants. The basic BAN logic symbols and their corresponding meanings are shown in Table II. The basic rules of BAN logic reasoning are shown in Table III.

1) The idealized form of the proposed protocol is as follows.

Message 1: $U_i \rightarrow GW : \langle N_i, \alpha \rangle_{U_i \overset{k_i}{\leftrightarrow} GW}, DID_i, SID_i$.

Message 2: $GW \rightarrow SN_n : \langle SK_n, \beta \rangle_{SN_n \overset{R_n}{\leftrightarrow} GW}, C_n$.

Message 3: $SN_n \rightarrow GW : \langle \gamma \rangle_{SN_n \overset{R_n}{\leftrightarrow} GW}$.

Message 4: $GW \rightarrow U_i : \langle SK_i, \lambda \rangle_{U_i \overset{k_i}{\leftrightarrow} GW}$.

2) It is important to prove the following goals.

GOAL1: $U_i| \equiv (U_i \leftrightarrow^{SK} GW)$.
GOAL2: $GW| \equiv U_i| \equiv (U_i \leftrightarrow^{SK} GW)$.
GOAL3: $U_i| \equiv GW| \equiv (U_i \leftrightarrow^{SK} GW)$.
GOAL4: $SN_n| \equiv (SN_n \leftrightarrow^{SK} GW)$.
GOAL5: $GW| \equiv SN_n| \equiv (SN_n \leftrightarrow^{SK} GW)$.
GOAL6: $SN_n| \equiv GW| \equiv (SN_n \leftrightarrow^{SK} GW)$.

3) The initial state of our protocol is defined as follows.

A1: $U_i| \equiv \#(b_i^{\text{new}})$.
A2: $GW| \equiv \#(b_i^{\text{new}}, b_n^{\text{new}}, SK)$.
A3: $SN_n| \equiv \#(b_n^{\text{new}})$.
A4: $U_i| \equiv (U_i \leftrightarrow^{k_i} GW)$.
A5: $GW| \equiv (U_i \leftrightarrow^{k_i} GW)$.
A6: $U_i| \equiv GW| \Rightarrow (U_i \leftrightarrow^{SK} GW)$.
A7: $SN_n| \equiv (SN_n \leftrightarrow^{R_n} GW)$.
A8: $GW| \equiv (SN_n \leftrightarrow^{R_n} GW)$.
A9: $SN_n| \equiv GW| \Rightarrow (SN_n \leftrightarrow^{SK} GW)$.
A10: $GW| \equiv (U_i \leftrightarrow^{SK} GW)$.
A11: $GW| \equiv (SN_n \leftrightarrow^{SK} GW)$.

4) The proof process is as follows.

Proof:

S1: According to Message 1, we get $GW \triangleleft (\langle N_i, \alpha \rangle_{U_i \overset{k_i}{\leftrightarrow} GW}, DID_i, SID_i)$.
S2: According to A5 and R1, we can get $GW| \equiv U_i| \sim (N_i, \alpha, DID_i, SID_n)$.
S3: According to A3 and R4, we get $GW| \equiv \#(N_i, \alpha, DID_i, SID_n)$.
S4: According to S2, S3, and R3, we get $GW| \equiv U_i| \equiv (N_i, \alpha, DID_i, SID_n)$.
S5: According to S4 and A5, we can deduce that $GW| \equiv U_i| \equiv (U_i \leftrightarrow^{SK} GW)$. (**GOAL2**).
S6: According to Message 2, we can get $SN_n \triangleleft (\langle SK_n, \beta \rangle_{SN_n \overset{R_n}{\leftrightarrow} GW}, C_n)$.
S7: According to A7 and R1, we can get $SN_n| \equiv GW| \sim (SK_n, \beta, C_n)$.
S8: According to A3 and R4, we can get $SN_n| \equiv \#(SK_n, \beta, C_n)$.
S9: According to S7, S8, and R3, we get $SN_n| \equiv GW| \equiv (SK_n, \beta, C_n)$.
S10: According to S9 and A8, we can deduce that $SN_n| \equiv GW| \equiv (SN_n \leftrightarrow^{SK} GW)$. (**GOAL6**).
S11: According to S9, S10, A9, and R2, we can get $SN_n| \equiv (SN_n \leftrightarrow^{SK} GW)$. (**GOAL4**).
S12: According to Message 3, we can get $GW \triangleleft (\langle \gamma \rangle_{SN_n \overset{R_n}{\leftrightarrow} GW})$.
S13: According to A8 and R1, we can get $GW| \equiv SN_n| \sim (\gamma)$.
S14: According to A2 and R4, we can get $GW| \equiv \#(\gamma)$.
S15: According to S13, S14, and R3, we get $GW| \equiv SN_n| \equiv (\gamma)$.
S16: According to S15 and R5, we get $GW| \equiv SN_n| \equiv (SN_n \leftrightarrow^{SK} GW)$. (**GOAL5**).
S17: According to Message 4, we get $U_i \triangleleft (\langle SK_i, \lambda \rangle_{U_i \overset{k_i}{\leftrightarrow} GW})$.
S18: According to A4 and R1, we get $U_i| \equiv GW| \sim (SK_i, \lambda)$.
S19: According to A1 and R4, we get $U_i| \equiv \#(SK_i, \lambda)$.
S20: According to S18, S19, and R3, we get $U_i| \equiv GW| \equiv (SK_i, \lambda)$.
S21: According to S20 and A4, we can deduce that $U_i| \equiv GW| \equiv (U_i \leftrightarrow^{SK} GW)$. (**GOAL3**).
S22: According to S20, S21, A6, and R2, we get $U_i| \equiv (U_i \leftrightarrow^{SK} GW)$. (**GOAL1**). ■

C. Formal Security Proof Using ProVerif

ProVerif [32] is a π-algorithm-based automated cryptographic protocol verification tool developed by Bruno Blanchet using the Prolog language. During more than 20 years of development, ProVerif has been widely used for the formal verification of cryptographic protocols [30], [33], [34], [35].

Our ProVerif validation code is divided into three sections: 1) a declaration section; 2) a query section; and 3) a protocol flow section. The declaration section defines the variable name and type for each variable. We use two channel types, Private Channel *sch* for communication participants to pass sensitive messages during the registration phase, and Public Channel *ch* for communication participants to pass messages publicly. The parameters and functions required for the validation process are also defined. The code of the declaration section is shown in Fig. 3.

```
(*---------channel--------*)
free sch:channel[private]. (* Secure channel *)
free ch:channel. (* Public channel  *)
(*----Ui's parameters------*)
free IDui:bitstring[private].
free PWui:bitstring[private].
free SKui:bitstring[private].
(*----------SN's parameters--------------*)
free SNn:bitstring[private].
free Rn:bitstring[private].
free SKsn:bitstring[private].
(*-----------GW's parameters-------------*)
free bsn:bitstring[private].*)
free SKgw:bitstring[private].
free xRn:bitstring[private].
(*--------------Fuctions-----------------------*)
fun Hash(bitstring):bitstring.
(*string concatenation*)
fun Concat(bitstring,bitstring):bitstring.
(* XOR operation *)
fun XORFun(bitstring, bitstring) : bitstring.
(*fuzzy extractor GEN func.*)
fun Gen1(bitstring):bitstring.
fun Gen2(bitstring):bitstring.
(*fuzzy extractor REC func.*)
fun REC(bitstring,bitstring):bitstring.
(*PUF function*) fun PUF(bitstring):bitstring.
equation forall m:bitstring,n:bitstring;
   XORFun(XORFun(m,n),n)=m.
equation forall a:bitstring;
   REC(a,Gen1(a))=Gen2(a).
equation forall b:bitstring;
   REC(b,Gen2(b))=Gen1(b).
reduc forall m1:bitstring, m2:bitstring;
   SeparateFirst(Concat(m1,m2)) = m1.
reduc forall m11:bitstring, m22:bitstring;
   SeparateSecond(Concat(m11,m22)) = m22.
(*-----------events--------------*)
event Uistart.
event Uiend.
event GWstart.
event GWend.
event Snstart.
event Snend.
```

Fig. 3. Declaration statements in ProVerif codes.

As shown in Fig. 4, we define a set of query statements to verify the key security and mutual authenticity of each participant, as well as the identity of the user and the security of the biometric password. Where `noninterf IDui` is the query used to verify the strong anonymity of the user, and `weaksecret PWui` is the query used to verify the resistance against the offline password guessing attack.

We have coded the protocol flow section according to the protocol execution steps, which are not listed here due to space constraints. The output in Fig. 5 shows that our proposed

```
(*------------queries-------------*)
query attacker(SKui).
query attacker(SKsn).
query attacker(SKgw).
query attacker(PWui).
query attacker(IDui).
query inj-event(GWend)==>inj-event(Uistart).
query inj-event(GWend)==>inj-event(Snstart).
query inj-event(Uiend)==>inj-event(GWstart).
query inj-event(Snend)==>inj-event(GWstart).
query inj-event(Uiend)==>inj-event(Snstart).
query inj-event(Snend)==>inj-event(Uistart).
noninterf IDui.
weaksecret PWui.
```

Fig. 4. Query statements in ProVerif codes.

```
------------------------------------------------
Verification summary:
Query not attacker(SKui[]) is true.
Query not attacker(SKsn[]) is true.
Query not attacker(SKgw[]) is true.
Query not attacker(PWui[]) is true.
Query not attacker(IDui[]) is true.
Query inj-event(GWend)==>inj-event(Uistart) is true.
Query inj-event(GWend)==>inj-event(Snstart) is true.
Query inj-event(Uiend)==>inj-event(GWstart) is true.
Query inj-event(Snend)==>inj-event(GWstart) is true.
Query inj-event(Uiend)==>inj-event(Snstart) is true.
Query inj-event(Snend)==>inj-event(Uistart) is true.
Non-interference IDui is true.
Weak secret PWui is true.
------------------------------------------------
```

Fig. 5. Result of ProVerif.

protocol satisfies mutual authentication and guarantees the security of keys, user identities, and biometric passwords.

VI. Comparison of Security and Performance

This section compares the proposed scheme in terms of security, computational, and communication overheads with the tripartite authentication schemes of Masud et al. [15], Sharma and Kalra [36], Shihab and AlTawy [18], Kim et al. [17], and Kwon et al. [16] in IoMT scenario. Among them, all of the relevant schemes are lightweight, using only hash functions and XOR operations to design the protocol, where [16], [17], and [18] are improvements on Masud et al.'s scheme [15] and also use biometrics. In short, in the above protocol, the communication entities concerned, i.e., the gateway, the user, and the sensor, are all involved in the authentication and key exchange process.

A. Comparison of Security

Table IV shows the results of comparing the security of the same type of protocol. Where ✓ indicates that the protocol has this security attribute, while ✗ indicates that the protocol does not have this security attribute. The SPs in the table represent the various security attributes compared.

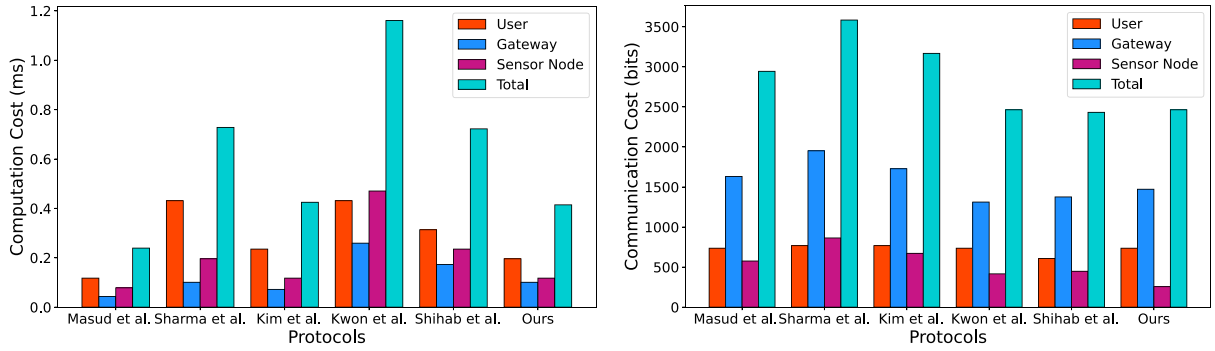

Fig. 6. Performance comparison. (a) Comparison of computation cost. (b) Comparison of communication cost.

TABLE IV
COMPARISON OF SECURITY WITH RELEVANT PROTOCOLS

Security Property	Protocol					
	[15]	[36]	[17]	[16]	[18]	Ours
SP1	✓	✓	✓	✓	✓	✓
SP2	✗	✗	✗	✓	✓	✓
SP3	✗	✗	✗	✓	✗	✓
SP4	✗	✗	✗	✓	✓	✓
SP5	✗	✗	✗	✓	✓	✓
SP6	✓	✓	✓	✓	✓	✓
SP7	✓	✓	✓	✓	✓	✓
SP8	✗	✓	✗	✓	✓	✓
SP9	✓	✗	✗	✓	✗	✓

SP1 represents Mutual authentication, SP2 represents anonymity and untraceability, SP3 represents resistance to cloning and physical attacks, SP4 represents resistance to offline password guessing attacks, SP5 represents resistance to impersonation attacks, SP6 represents resistance to replay attacks, SP7 represents resistance to DoS attacks, SP8 represents session key security and SP9 represents resistance to desynchronization attacks.

From the comparison results, it can be seen that our protocol is resistant to physical or cloning attacks because it uses PUFs. It should be noted that, given the balance between efficiency and stronger security, we do not use DH key exchange and therefore do not have forward security.

B. Comparison of Computation Cost

The computational overhead includes all cryptographic operations to complete authentication and key exchange. Since the compared schemes all involve only hashing operations, we only calculate the overhead of the hashing operations. Note that we do not consider operations, such as XOR and concatenation, which have negligible computational overhead. In this section, we use a TSMC 7-nm Snapdragon 870 CPU, Android 12, 8G RAM device to simulate a resource-constrained device, and an AMD Ryzen5 3500U 2.10-GHz CPU, Ubuntu 12.04, 16G RAM device to simulate a nonresource-constrained device. We implement and calculate the computational cost of the hash operations in the above protocol. After 10 000 executions, we had an average execution time of 0.0392 ms (for resource-constrained devices) and 0.0144 ms (for nonresource-constrained devices), respectively, as shown in Table V. Table VI and Fig. 6(a) give the comparative results for the relevant protocols in terms of computational overhead.

TABLE V
HASH FUNCTION EXECUTION TIME

	CPU	OS	RAM	Platform	Time(ms)
RD[1]	Snapdragon 870	Android 12	8G	Java	0.0392
URD[2]	AMD R5 3500U	Ubuntu 12.04	16G	Java	0.0144

[1]Resource-restricted Devices, [2]Unrestricted Resources Devices

TABLE VI
COMPARISON OF COMPUTATION COST

Protocol	Computation Cost (ms)			
	User	Gateway	Sensor Node	Total
[15]	$3T_h \approx 0.1176$	$3T_h \approx 0.0432$	$2T_h \approx 0.0784$	0.2392
[36]	$11T_h \approx 0.4312$	$7T_h \approx 0.1008$	$5T_h \approx 0.1960$	0.7280
[17]	$6T_h \approx 0.2352$	$5T_h \approx 0.0720$	$3T_h \approx 0.1176$	0.4248
[16]	$11T_h \approx 0.4312$	$18T_h \approx 0.2592$	$12T_h \approx 0.4704$	1.1608
[18]	$8T_h \approx 0.3136$	$12T_h \approx 0.1728$	$6T_h \approx 0.2352$	0.7216
Ours	$5T_h \approx 0.1960$	$7T_h \approx 0.1008$	$3T_h \approx 0.1176$	0.4144

TABLE VII
COMPARISON OF COMMUNICATION COST

Protocol	Communication Cost (bits)			
	User	Gateway	Sensor Node	Total
[15]	736	1632	576	2944
[36]	768	1952	864	3584
[17]	768	1728	672	3168
[16]	736	1312	416	2464
[18]	608	1376	448	2432
Ours	736	1472	256	2464

C. Comparison of Communication Cost

The communication overhead is the amount of data exchanged or transmitted by the participants in completing the authentication process. The following assumptions are made uniformly in this section: the length of the random number, the identity, the challenge, and the response is 160 bits, the length of the timestamp is 32 bits, and the output of the hash function is 256 bits. In the proposed protocol, message $M_1\{N_i, \alpha, DID_i, SID_n\}$ sends $160 + 256 + 160 + 160 = 736$ bits, message $M_2\{SK_n, \beta, C_n\}$ sends $320 + 256 + 160 = 736$ bits, message $M_3\{\gamma\}$ sends 256 bits, and message $M_4\{SK_i, \lambda\}$ sends $480 + 256 = 736$ bits. Table VII and Fig. 6(b) give the required communication overhead and the cumulative overhead for the user, gateway, and sensor parties in the proposed protocol and the associated protocol, respectively.

Generally speaking, the results of the comparison show that our proposed protocol is computationally and communicationally efficient while maintaining a strong guarantee of security.

VII. Conclusion

This article presents an improved authenticated key exchange protocol designed specifically for an IoMT application scenario, ensuring privacy and secure communication among doctors, patients, and medical gateways. The proposed protocol employs PUFs to withstand physical threats while leveraging biometrics, fuzzy extractors, and secret salt values to significantly heighten the level of difficulty for password guessing attacks. To establish the security of the protocol, a comprehensive array of formal and heuristic security analysis methods is employed, validating its robustness. Comparative evaluations against other relevant protocols demonstrate that our proposed solution achieves an optimal balance between security and efficiency in resource-constrained IoMT scenarios.

Acknowledgment

The authors would like to thank the anonymous referees for their constructive comments.

References

[1] G. Hatzivasilis, O. Soultatos, S. Ioannidis, C. Verikoukis, G. Demetriou, and C. Tsatsoulis, "Review of security and privacy for the Internet of Medical Things (IoMT)," in *Proc. 15th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, 2019, pp. 457–464.

[2] M. N. Bhuiyan, M. M. Rahman, M. M. Billah, and D. Saha, "Internet of things (IoT): A review of its enabling technologies in healthcare applications, standards protocols, security, and market opportunities," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10474–10498, Jul. 2021.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[4] A. Arfaoui, A. Kribeche, and S.-M. Senouci, "Context-aware anonymous authentication protocols in the Internet of Things dedicated to e-health applications," *Comput. Netw.*, vol. 159, pp. 23–36, Aug. 2019.

[5] Z. Mahmood, H. Ning, A. Ullah, and X. Yao, "Secure authentication and prescription safety protocol for telecare health services using ubiquitous IoT," *Appl. Sci.*, vol. 7, no. 10, p. 1069, 2017.

[6] M. A. Iqbal and M. Bayoumi, "Secure end-to-end key establishment protocol for resource-constrained Healthcare sensors in the context of IoT," in *Proc. 2016 Int. Conf. High Perform. Comput. Simul. (HPCS)*, 2016, pp. 523–530.

[7] K. Park et al., "LAKS-NVT: Provably secure and lightweight authentication and key agreement scheme without verification table in medical Internet of Things," *IEEE Access*, vol. 8, pp. 119387–119404, Jun. 2020.

[8] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and X. Li, "Cryptanalysis and enhancement of anonymity preserving remote user mutual authentication and session key agreement scheme for E-health care systems," *J. Med. Syst.*, vol. 39, no. 11, pp. 1–21, 2015.

[9] K. Fan, W. Jiang, H. Li, and Y. Yang, "Lightweight RFID protocol for medical privacy protection in IoT," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1656–1665, Apr. 2018.

[10] S. F. Aghili, H. Mala, P. Kaliyar, and M. Conti, "SecLAP: Secure and lightweight RFID authentication protocol for medical IoT," *Future Gener. Comput. Syst.*, vol. 101, pp. 621–634, Dec. 2019.

[11] J. Kang, K. Fan, K. Zhang, X. Cheng, H. Li, and Y. Yang, "An ultra light weight and secure RFID batch authentication scheme for IoMT," *Comput. Commun.*, vol. 167, pp. 48–54, Feb. 2021.

[12] V. Yanambaka, S. Mohanty, E. Kougianos, D. Puthal, and L. Rachakonda, "PMsec: PUF-based energy-efficient authentication of devices in the Internet of Medical Things (IoMT)," in *Proc. IEEE Int. Symp. Smart Electron. Syst. (iSES) (Formerly iNiS)*, 2019, pp. 320–321.

[13] K.-L. Tsai, Y.-L. Huang, F.-Y. Leu, and I. You, "TTP based high-efficient multi-key exchange protocol," *IEEE Access*, vol. 4, pp. 6261–6271, Sep. 2016.

[14] F. Wu et al., "A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks," *Future Gener. Comput. Syst.*, vol. 82, pp. 727–737, May 2018.

[15] M. Masud, G. S. Gaba, K. Choudhary, M. S. Hossain, M. F. Alhamid, and G. Muhammad, "Lightweight and anonymity-preserving user authentication scheme for IoT-based Healthcare," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2649–2656, Feb. 2022.

[16] D. Kwon, Y. Park, and Y. Park, "Provably secure three-factor-based mutual authentication scheme with PUF for wireless medical sensor networks," *Sensors*, vol. 21, no. 18, p. 6039, 2021.

[17] K. Kim, J. Ryu, Y. Lee, and D. Won, "An improved lightweight user authentication scheme for the Internet of Medical Things," *Sensors*, vol. 23, no. 3, p. 1122, 2023.

[18] S. Shihab and R. AlTawy, "Lightweight authentication scheme for Healthcare with robustness to Desynchronization attacks," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 18140–18153, Oct. 2023.

[19] S. Wang, X. Zhou, K. Wen, B. Weng, and P. Zeng, "Security analysis of a user authentication scheme for IoT-based healthcare," *IEEE Internet Things J.*, vol. 10, no. 7, pp. 6527–6530, Apr. 2023.

[20] D. R. Brown, "Generic groups, collision resistance, and ECDSA," *Des., Codes Crypt.*, vol. 35, no. 1, pp. 119–152, 2005.

[21] S. S. Sahoo, S. Mohanty, and B. Majhi, "Improved biometric-based mutual authentication and key agreement scheme using ECC," *Wireless Pers. Commun.*, vol. 111, no. 2, pp. 991–1017, 2020.

[22] G. S. Rawat, K. Singh, N. I. Arshad, K. Hadidi, and A. Ahmadian, "A lightweight authentication scheme with privacy preservation for vehicular networks," *Comput. Elect. Eng.*, vol. 100, May 2022, Art. no. 108016.

[23] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Int. Conf. Theory Appl. Crypt. Techn.*, 2004, pp. 523–540.

[24] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[25] Q. Chen, K. Su, C. Liu, and Y. Xiao, "Automatic verification of Web service protocols for epistemic specifications under Dolev-Yao model," in *Proc. Int. Conf. Serv. Sci.*, 2010, pp. 49–54.

[26] I. A. Tsoukalas and P. D. Siozos, "Privacy and anonymity in the information society—Challenges for the European union," *Sci. World J.*, vol. 11, pp. 458–462, Mar. 2011.

[27] Y. K. Lee, L. Batina, and I. Verbauwhede, "EC-RAC (ECDLP based randomized access control): Provably secure RFID authentication protocol," in *Proc. IEEE Int. Conf. RFID*, 2008, pp. 97–104.

[28] D. Pointcheval, "Topics in cryptology—CT-RSA 2006" in *Proc. Crypt. Track RSA Conf.*, vol. 3860, San Jose, CA, USA, 2006, pp. 192–225.

[29] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst. (TOCS)*, vol. 8, no. 1, pp. 18–36, 1990.

[30] N. Kobeissi, K. Bhargavan, and B. Blanchet, "Automated verification for secure messaging protocols and their implementations: A symbolic and computational approach," in *Proc. 2017 IEEE Eur. Symp. Security Privacy (EuroS&P)*, 2017, pp. 435–450.

[31] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. 19th Annu. Int. Crypt. Conf.*, Santa Barbara, CA, USA, 1999, pp. 388–397.

[32] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "*ProVerif 2.00: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*," INRIA, Paris, France, 2018.

[33] K. Bhargavan, B. Blanchet, and N. Kobeissi, "Verified models and reference implementations for the TLS 1.3 standard candidate," in *Proc. IEEE Symp. Security Privacy (SP)*, 2017, pp. 483–502.

[34] N. Kobeissi, G. Nicolas, and K. Bhargavan, "Noise explorer: Fully automated modeling and verification for arbitrary noise protocols," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS P)*, 2019, pp. 356–370.

[35] V. Cortier, D. Galindo, and M. Turuani, "A formal analysis of the Neuchâtel e-voting protocol," in *Proc. IEEE Eur. Symp. Security Privacy (EuroS P)*, 2018, pp. 430–442.
[36] G. Sharma and S. Kalra, "A lightweight user authentication scheme for cloud-IoT based healthcare services," *Iranian J. Sci. Technol. Trans. Elect. Eng.*, vol. 43, no. 1 pp. 619–636, 2019.

Xin Zhou received the M.S. degree from the School of Information Science and Technology, Hangzhou Normal University, Hangzhou, China. He is currently pursuing the Ph.D. degree with the School of Mathematics and Statistics, Fujian Normal University, Fuzhou, China.

His research interests include authenticated key exchange protocols, secure multiparty computation, blockchain, and public-key cryptography.

Shengbao Wang received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2008.

He is currently an Associate Professor with the Key Laboratory of Cryptography Technology of Zhejiang Province and the School of Information Science and Technology, Hangzhou Normal University, Hangzhou, China. His research interests lie in public-key cryptography, especially focus on public-key encryption, and authenticated key agreement protocols.

Kang Wen is currently pursuing the M.S. degree with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou, China.

His research interests include authenticated key exchange protocols, blockchain, and public-key cryptography.

Bin Hu received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2009.

He is currently an Associate Professor with the Key Laboratory of Cryptography Technology of Zhejiang Province and the School of Information Science and Technology, Hangzhou Normal University, Hangzhou. His research interests lie in cryptographic protocol design and cryptography engineering.

Xiao Tan received the Ph.D. degree from the School of Computer Science, City University of Hong Kong, Hong Kong, China, in 2014.

He is currently a Lecturer with the Key Laboratory of Cryptography Technology of Zhejiang Province and the School of Information Science and Technology, Hangzhou Normal University, Hangzhou, China. His research interests include applied cryptography, fair exchange protocols, cloud security, and blockchain security.

Qi Xie received the Ph.D. degree in applied mathematics from Zhejiang University, Hangzhou, China, in 2005.

He was a Visiting Scholar with the Department of Computer Science, University of Birmingham, Birmingham, U.K., from 2009 to 2010 and the Department of Computer Science, City University of Hong Kong, Hong Kong, in 2012. He is a Professor with Hangzhou Normal University, Hangzhou, and the Key Laboratory of Cryptography of Zhejiang Province. His research interests include applied cryptography, including digital signatures, authentication, and key agreement protocols.