IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 February 2024, accepted 3 March 2024, date of publication 5 March 2024, date of current version 12 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3373879

RESEARCH ARTICLE

Enhanced Lightweight Medical Sensor Networks Authentication Scheme Based on Blockchain

TAEWOONG KANG, NARYUN WOO, AND JIHYEON RYU
School of Computer and Information Engineering, Kwangwoon University, Seoul 01897, Republic of Korea

Corresponding author: Jihyeon Ryu (jhryu@kw.ac.kr)

This work was supported in part by the Research Grant of Kwangwoon University in 2023, and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) under Grant RS-2023-00239728.

ABSTRACT In the rapidly evolving environment of wireless medical sensor networks (WMSN) and the internet of medical things (IoMT), remote medical support has seen unprecedented advancements. It is essential that the data relayed from the sensors must be trustworthy and unaltered, and that the sensors themselves are genuine. Wireless networks, however, have inherent vulnerabilities. In addition, since WMSN is directly linked to patients' lives, its continuous availability is crucial. Considerable efforts have been made to maintain the integrity and authenticity of such data. However, many studies have failed to address the problem of a single point of failure (SPOF). This issue has been particularly detrimental to patients who require ongoing management. To address this issue and ensure the protection of the authenticity and integrity of patient data, we suggest the implementation of an authentication scheme based on blockchain technology. In 2022, Yu et al. introduced a blockchain-integrated authentication and key generation scheme for WMSN using Physical Unclonable Functions (PUFs), effectively addressing the SPOF problem by conducting mutual authentication through smart contracts without relying on centralized servers. Our research found that this scheme inadvertently shared critical parameters, including challenge-response pairs and important private keys, with the blockchain network, making it vulnerable to various breaches. We present an enhanced protocol designed to mitigate these security challenges. By limiting the data interaction with smart contracts and ensuring only relevant parties access crucial parameters, our approach reduces the risk of public information disclosure on the blockchain. This not only mitigates the SPOF issue but also efficiently helps in prevention of physical attacks. We prove that our proposed system prevents known security vulnerabilities through informal and formal analysis using the Scyther, Proverif, and BAN logic. Furthermore, the proposed scheme offers 67.37% reduction in computation costs and 3.67% in communication costs, presenting an efficient and secure solution for WMSN in the IoMT landscape.

INDEX TERMS User authentication, fuzzy extractor, physical unclonable functions, wireless medical sensor networks, blockchain network.

I. INTRODUCTION

The rapid development of the Internet of Things (IoT) and 5G wireless networks has led to an increase in research related to the Internet of Medical Things (IoMT) and wireless medical sensor networks (WMSN) [1]. These technological advancements have allowed for a plethora of innovations in the medical field. WMSN-based systems now enable functionalities that were either challenging or impossible for traditional medical systems [2]. For instance, they facilitate continuous patient monitoring, swift emergency responses, and more efficient treatments for chronic diseases such as diabetes [3].

One of the major advantages of these systems is their ability to offer remote medical support. This allows patients to receive care and monitoring without the necessity of being physically present at medical facility. As depicted in Fig. 1, patient information is transmitted to medical professionals

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad S. Khan.

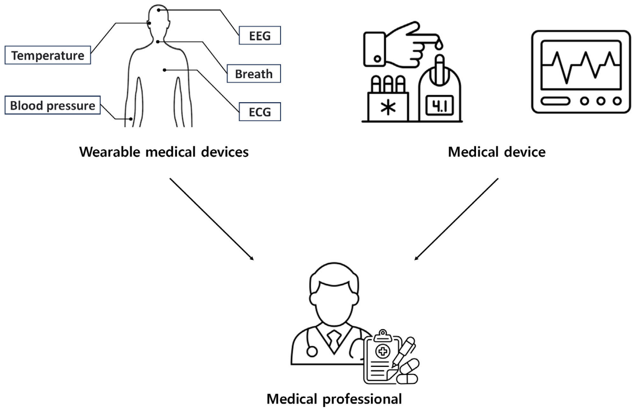


FIGURE 1. Utilization of IoMT.

via wireless networks from wearable or embedded medical device. This is especially useful for individuals living in remote areas or those with mobility issues [1], [2], [3], [4]. However, due to the inherent lack of physical boundaries in wireless networks, the signals being transmitted are vulnerable to various potential threats, including interception, tampering, and the injection of malicious payloads by attackers. In scenarios where attackers transmit inaccurate information to medical professionals or intentionally overload the server leading to its failure, such incidents can escalate into severe problems directly impacting patient safety. Within a medical environment, user authentication is paramount. It must meticulously manage sensitive patient data, ensuring confidentiality and security. This is necessary for ensuring sensitivity of personal information and electronic health records, as well as for facilitating accurate and collaborative assessments of patient conditions among different medical professionals, thereby aiding in collaborative decision-making.

To bolster patient privacy protection, researchers have created medical environments leveraging WSNs. These IoMT setups generally consist of four main components: (1) A medical professional interface, (2) sensor nodes that measure vital metrics such as heart rate, blood pressure, body temperature, etc., (3) a gateway node that transmits data from the sensor nodes to a central server, and (4) a central server that collects and analyzes the information. Given the inherent characteristics of WSNs, where data packets might be easily captured and modified, ensuring the confidentiality and integrity of WSNs is crucial.

To address security concerns in the WMSN environment, various security schemes based on centralized server approaches have been introduced [5], [6], [8], [9], [10], [11]. Each of these schemes aimed to meet specific security and performance requirements. In 2016, Li et al. [5] introduced a scheme centered on verifying the authenticity of users in the WMSN environment. However, Das et al. [6] identified vulnerabilities in Li's scheme, specifically related to privileged-insider attacks and sensor node capture attacks. As a remedy, Das et al. proposed a new scheme that utilized smart cards. Subsequently, Wu et al. [7] proposed a two-factor authentication scheme tailored for WMSN. In addition, Li et al. [8] proposed a three-factor user authentication protocol based on elliptic curve cryptography (ECC). This protocol was designed to effectively defend against potential threats such as unauthorized mobile device access and denial-of-service (DOS) attacks. However, the approaches presented in this way were vulnerable to physical attacks such as tampering or replacement attacks on the sensors. Therefore, to address these issues, Alladi et al. [10] proposed a two-way authentication protocol leveraging techniques such as Physical Unclonable Function (PUF). The research landscape in the WMSN environment is ever-evolving, evidenced by Li et al.'s scheme [11] and Fotouhi et al.'s scheme [9], which simplifies the protocol by exclusively using hash functions combined with XOR operations. While many studies [5], [6], [8], [9], [10], [11] have been conducted, recurring theme is the prevalent reliance on a trustworthy centralized gateway node (GWN) to verify data integrity. This dependence on a centralized management system introduces a critical vulnerability. Should this central system malfunction for any reason, it may cause the entire medical monitoring system to collapse, potentially jeopardizing the health of many patients, especially those heavily dependent on these systems. Such centralized systems inherently carry the limitation of being a SPOF and often fail to address the associated challenges.

In environments where high availability is essential, such as WMSN, blockchain technology is effective in solving SPOF problems and simultaneously ensures data integrity. Due to these characteristics, it is being widely utilized in fields where availability and integrity are important [12]. Over the past few years, numerous studies have been conducted to address SPOF problem by integrating blockchain technology into WMSN. Specifically, Wang et al. [13] introduced a scheme combining blockchain technology and PUF, effectively solving the SPOF issue and enhancing resistance to physical attacks. However, in 2022, Yu et al. [14] identified certain vulnerabilities in this approach, such as susceptibility to the man-in-the-middle (MITM) attack and session key disclosure attack. As a countermeasure, they proposed an alternative scheme. Nevertheless, the revised scheme [14] exhibited flaws, such as the sensor capture attack and potential disclosure of secretive parameters to the smart contract (SC). In this scheme [14], we have discovered that important secret intermediary parameters, such as the private key of gateway and the challenge-response pairs of communication participants, are being shared with the smart contract. This exposes essential secret parameters, crucial for secure key generation, on the blockchain network. Additionally, we found that the inappropriate concatenation of data is causing information leakage, which is manifested due to the properties of XOR operations. Therefore, there is a concern that information about the identity of user or nonce may not be properly concealed and could be intercepted in transit. Our proposed scheme completely solves the sensor capture attack (physical capture attack) and the disclosure of secretive parameters to smart contracts (stolen verifier attack). Moreover, it proposes solutions for a total of eight attack methods, including replay attack, insider attack, man-in-the-middle attack, offline password guessing attack, perfect forward secrecy, and impersonation attack. Our paper proposes an enhanced authentication scheme for WMSN, leveraging the capabilities of blockchain technology. Our primary contributions can be summarized as follows:

- Instead of storing the entire PUF pair in both the GWN's local database and the SC, we employ the PUF solely for private key generation. In addition to impeding attackers from deciphering the properties of PUF pairs, it also serves as a deterrent against the concentration of PUF pairs in any database, thereby discouraging the interest of potential attackers.
- To prevent data leakage caused by XOR operations between data of incompatible lengths, we utilize two hash functions. In our proposed scheme the hash function is designed to generate output with a length consistently equal to or greater than the data undergoing XOR operations. This configuration effectively safeguards against the potential disclosure of information through these operations, ensuring the security of the data handling process.
- To minimize the information passed to the SC, the GWN, rather than the SC, executes the computations. The SC's role is confined to comparing values and executing predetermined events. This approach prevents the transfer of crucial secret parameters to the SC and their subsequent disclosure on the blockchain network, thereby offering a robust defense against potential threats, such as attacks from stolen verifiers, that arise from information disclosure.
- Our scheme underwent rigorous validation processes. We conducted both informal and formal security analyses using ProVerif, Scyther, and BAN logic. These evaluations confirm the enhanced security performance of our scheme compared to its counterparts. Additionally, we performed a comparative analysis of communication and computational costs with state-of-the-art research. Our scheme demonstrates an exemplary 67.37% reduction in computational costs and a 3.67% reduction in communication costs.

By addressing these issues, our proposed scheme successfully eliminates the SPOF problem and effective defense mechanism against potential attacks such as the stolen verifier attack.

The remainder of this paper is structured as follows: Section II summarizes the key concepts, the system model, and the attack model. Section III demonstrates the workings of our proposed scheme. Section IV demonstrates the security credentials of our scheme, showcasing both informal and formal analyses. Section V offers a comparative study of the computational and communication costs of the proposed scheme in relation to previous research. Last, we conclude our research and present the future work in Section VI.

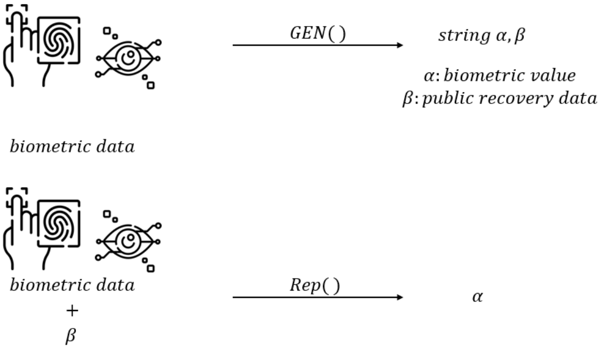


FIGURE 2. Method of fuzzy extractor.

II. PRELIMINARIES

In this section, we introduce the concepts of the fuzzy extractor and the PUFs, both of which underpin the scheme we propose. We will also discuss the conditions and potential attack scenarios for which the scheme has been designed. The detailed content is as follows:

A. FUZZY EXTRACTOR

There can be various methods to recognize a person. For example, facial recognition [15] or identifying their usual emotions to check if their patterns match their original patterns [16]. Among these, we commonly use the method known as a fuzzy extractor.

We employ fuzzy extractors to extract the user's biometric information [17], [18], [19]. Given the inherent challenges in consistently inputting biometric data, such as fingerprints and iris patterns, as identical values every time, this approach becomes essential. As explained in figure 2, fuzzy extractors are typically divided into two primary components:

1) GEN(): This probabilistic generation procedure uses biometric data to extract an α value and generate public recovery data β. This procedure is denoted as $\mathrm{GEN}(BIO) = \{\alpha, \beta\}$. While α maintains a fixed bit length of m, β does not reveal any information about α.
2) REP(): This deterministic process involves retrieving the α value using the public recovery data β from the input BIO'. This procedure can be represented as $\mathrm{REP}(BIO', \beta) = \alpha$. This equation holds true only when the Hamming distance between BIO and BIO' stays within an acceptable error range.

B. PHYSICAL UNCLONABLE FUNCTION

PUFs generate different response patterns for individual devices by exploiting the unique physical characteristics of integrated circuits (IC) [20]. When a user provides a challenge input C, the PUF generates a response value R, denoted as $R = PUF(C)$. Intriguingly, the R varies even with the same input C if the device itself is different. This characteristic implies that a single device will consistently provide the same response for a given challenge. However, when the same challenge is applied to a different device, the response varies. This is because it is impossible to manufacture IC identically which is why it is sometimes figuratively referred to as an 'electronic fingerprint'. Given these characteristics, PUFs are emerging as a robust security tool. One of the defining advantages of PUFs is their resistance to digital hacking. Even if an attacker learns the challenge from numerous challenge-response pairs, they cannot interpret the PUF's unique characteristics. As a result, they cannot know the actual response value. Moreover, the current state of technology makes it impossible to replicate an IC perfectly. Thus, even if a malicious entity acquires the challenge, they cannot reproduce the authentic response. Through this, our proposed system enables each communication entity to utilize unique and robust individual keys. Even if physically compromised, attackers cannot access information about the private keys of communication participants. Therefore, this enhances the resistance of our proposed authentication scheme to physical attacks. However, PUFs exhibit sensitivity to environmental factors such as temperature and humidity, which can potentially introduce errors in the output. In the context of our proposed scheme, we operate on the premise of utilizing an ideal PUF, one that is resilient to such environmental variables.

C. BLOCKCHAIN NETWORK

Blockchain technology, as the name suggests, is composed of blocks linked in a chain [21]. Each block is connected using the hash value of the previous block, and these blocks contain data structures for storing transaction records. When a new block is created to record a new transaction, this information is propagated to all nodes in the network, ensuring that all participants maintain the same ledger. This process embodies the principles of distributed ledger technology. The method used to achieve agreement among all participants on the state of the distributed ledger is termed the consensus mechanism. This includes the proof of work method, which involves solving problems to create and add blocks, and the proof of stake method, where validators are selected based on their stake and are responsible for creating and validating blocks.

Due to the characteristics of the blockchain's distributed ledger, all participants can transparently verify changes in the network. Even if an attacker attempts to modify the ledger on a specific node, other nodes in the network will not recognize this change, making arbitrary data manipulation difficult. In the case of external attacks that alter the ledger of a node, the integrity and availability of the data are maintained through copies of the ledger held by other nodes.

More advanced systems like Ethereum [22] offer smart contract technology, allowing contracts to be programmed to execute automatically when certain conditions are met. When using smart contracts, transaction details are transparently recorded on the blockchain network. Therefore, sensitive information such as personal data must be stored off-chain or processed in an encrypted form to maintain confidentiality.

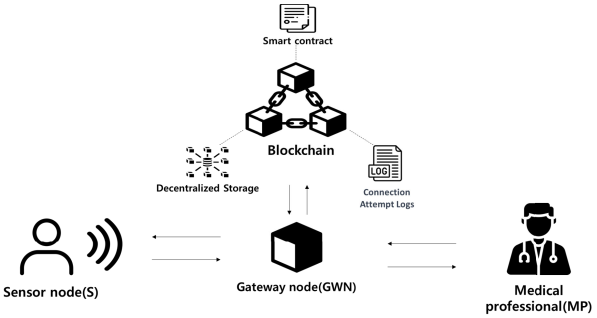


FIGURE 3. Network model of proposed scheme.

Thus, blockchain technology presents a trade-off between transparency and confidentiality. To balance these, we have designed the authentication process to include smart contracts. In this process, smart contracts verify authentication confirmation messages received from gateway nodes and record both the process and its outcome on the blockchain network to ensure transparency in the authentication process. During this process, necessary information is stored off-chain, and the information transmitted to smart contracts is encrypted to prevent the exposure of critical data on the blockchain network. If the number of authentication failures recorded on the blockchain network reaches the predetermined threshold, users are automatically blocked. This prevents malicious users from overwhelming the server. The inherent nature of smart contracts restricts participation in the authentication process to contracted parties, further enhancing the robustness of the protocol we have proposed. After authentication, established session keys are used to encrypt data, and this information can be automatically exchanged using smart contracts. This process is also transparently disclosed on the blockchain network. Such a method enhances user convenience and protects the WMSN environment from threats like DoS attacks. By applying blockchain in WMSN, as described above, it ensures transparency in the authentication process and resolves the issue of SPOF that originates from centralized server based systems. Additionally, it can enhance user convenience by utilizing smart contracts.

D. SYSTEM MODEL

This section introduces the configuration of the network in preparation for introducing our scheme. Drawing from the network models used in [23] and [24], we have configured our network model as illustrated in Fig. 3. This incorporates the blockchain network with SC via the GWN and is designed to facilitate communication between sensor nodes and medical professionals. It is assumed that all participant devices are equipped with a PUF by default.

1) Sensor Nodes: These nodes collect patient data and transmit it to medical professionals. Upon fulfilling certain predefined conditions in SC, the sensor nodes

autonomously transmit patient information to the concerned medical professionals. In this case, because the information transmitted to the blockchain network could potentially be made public, it is imperative that sensor nodes are designed to refrain from sending sensitive information to the SC. Prior to this information transfer, establishing a session key between the medical professionals and the sensor nodes becomes a critical step.

2) Gateway Nodes: GWNs essentially serve to facilitate the establishment of a blockchain network. They aid in generating session keys and mediating mutual authentication between medical professionals and sensor nodes. Crucially, not all data from the GWNs is shared with the blockchain network; only pertinent information for each stage is transmitted. Users with the capability to initiate a blockchain network can opt for a new network and a fitting consensus algorithm through these GWNs. However, if they are concerned about vulnerabilities inherent in small-scale blockchain networks, such as a single-point collisions or DOS attacks, they have the flexibility to deploy the network on established platforms such as Ethereum.

3) Medical Professionals: Before a medical professional can engage with the system, they must register on the blockchain network via the GWN, providing their password and biometric information. Post-registration, it becomes essential for them to engender a session key with the sensor node to enable information exchange. In the confines of our scheme, the session key is both generated and maintained during the login session. Thus, to receive information continuously, the medical professional must remain logged in.

E. ATTACK MODEL

Drawing inspiration from commonly used models such as those in [25] and [26], our attack model is structured based on the Dolev–Yao attack model [27]. The capabilities of attackers are defined as follows for the analysis of security properties. It is worth noting that we are not detailing the practical means by which an attacker might implement these capabilities.

1) The attackers have the ability to manipulate all messages transmitted across public channels. This includes the capability to intercept, modify, resend, and delete messages. However, messages sent over private channels remain outside the attacker's influence.

2) The attackers can illicitly acquire a legitimate user's MD and can potentially access the confidential user information stored therein, leveraging techniques such as differential power analysis attacks [28].

3) The attackers are incapable of discerning private keys of adequate length possessed by communication participants. Additionally, they cannot acquire hash function collisions within polynomial time constraints.

TABLE 1. **Notation.**

Notation	Description
MP_i	i-th Medical Professional
ID_{MP_i}	Identity of MP_i
MID_{MP_i}	Masked Identity of MP_i
PW	Password of MP_i
MPW	Masked Password of MP_i
BIO_i	Biometrics of MP_i
p_i	Secret Key of MP_i
S_k	k-th Sensor Node
ID_{S_k}	Identity of S_k
MID_{S_k}	Masked Identity of S_k
x_k	Secret Key of MP_i
GWN_j	j-th Gateway Node
ID_{GWN_j}	Identity of GWN_j
g_j	Secret Key of ID_{GWN_j}
PG	PUF Response using g_j as Challenge
KEY	Session Key between MP_i and S_k
SK, RN, SCN	Random Values
T_x	x-th Time Stamp
SC	Smart Contract
$H(), h()$	Two Distinct Hash Functions
$\oplus$	XOR Operation
$\|\|$	Concatenate Operation

4) While attackers might deploy physical attacks on sensors to extract secret information stored within [29] our scheme operates under the assumption that the PUF integrated into the sensor remains inscrutable.

III. PROPOSED SCHEME

In this section, we delineate our proposed scheme. The key notations integral to our scheme are collated in Table 1. Our scheme consists of several phases: the initialize phase, S_k registration phase, MP_i login and authentication phase, password and biometric information change phase, and finally, the data transfer phase. The initialize phase outlines the setup procedure for the GWN and SC. The registration phase involves the GWN receives information from communication participants, subsequently laying down a mutual private key. During the login and authentication phase, participants are mutually authenticated, establishing a session key for communication between S_k and MP_i. The password and biometric information change phase facilitates MP_i in enhancing their security. Lastly, the data transfer phase elaborates on the method for disseminating patient information. The subsequent content provides a detailed exposition of each phase.

A. INITIALIZE PHASE

In the initial phase, the protocol producer must configure a consensus mechanism. While the blockchain's consensus mechanism can be tailored to the situation, it is worth noting that if one cannot achieve an ideal stake division when creating a new blockchain network, utilizing the proof of stake is not advised. Consequently, the proof of work approach is recommended for this protocol. Furthermore, the producer must build SC to facilitate the registration and authentication processes of S_k and MP_i. This contract should be configured to invoke individual functions automatically based on certain conditions, such as the presence of timestamps or the size of the received data. Moreover, when constructing blocks for MP_i and S_k, it is essential to integrate a counter to track each participant's authentication failures. Setting a limit on the number of failures that would result in the block's suspension is also crucial. Lastly, given that off-chain information transmitted to a SC can be made public to blockchain participants, sensitive data should either be excluded from the SC or be encrypted prior to its inclusion.

B. S_K REGISTRATION PHASE

This phase precedes the login and authentication process. Here, S_k goes through respective registration steps with SC and formulates a confidential value that is jointly held with GWN_j. In the proposed protocol, ID_{S_k} is perceived as public information stored within the blockchain. All transactions in this registration stage occur through private communication channels. The intricate steps of the registration in our scheme are depicted in Fig. 4.

1) S_K first chooses the secret random value; x_k then generates X_k, such that $X_k = \mathrm{PUF}(x_k)$. S_K computes $MID_{S_k} = h(ID_{S_k} \parallel X_k)$ and then transmits $\{ID_{S_k}, MID_{S_k}, X_k\}$ to GWN_j.
2) After getting $\{ID_{S_k}, MID_{S_k}, X_k\}$, GWN_j computes $MID'_{S_K} = h(ID_{S_k} \parallel X_k)$ and generates a random value SCN_0. After that, GWN_j calls $SRegister$ function within SC with $\{h(MID'_{S_K} \parallel SCN_0), h(MID_{S_K} \parallel SCN_0), ID_{S_k}\}$. Then, SC checks for the duplication of ID_{S_k}. If ID_{S_k} is duplicated, it terminates the registration process; if not, SC checks $h(MID_{S_K}||SCN_0) \stackrel{?}{=} h(MID'_{S_K}||SCN_0)$. If this equation is valid, SC registers S_k with ID_{S_k} on the blockchain. Otherwise, SC terminates this process. After that GWN_k generates a random value SK_{S_k} and secret parameter PG, such that $PG = \mathrm{PUF}(g_j)$.
3) GWN_j computes shared private value $GS_1^{S_k} = h(g \parallel MID_{S_k} \parallel SK_{S_k})$. Then, using $GS_1^{S_k}$, GWN_j computes $GS_2^{S_k} = GS_1^{S_k} \oplus h(X_k \parallel MID_{S_k})$, $GS_3^{S_k} = GS_1^{S_k} \oplus PG \oplus h(g \parallel ID_{GWN_j})$, and $GS_4^{S_k} = h(GS_1^{S_k} \parallel X_k \parallel MID_{S_k})$. After that, GWN_j transmits $\{GS_2^{S_k}, GS_4^{S_k}\}$ to S_k. Then, GWN_j stores $\{GS_3^{S_k}, MID_{S_k}\}$ with $PUF(ID_{S_k})$ as key in local database.
4) After S_k obtains $\{GS_2^{S_k}, GS_4^{S_k}\}$, S_k computes $GS_1^{S'_k} = GS_2^{S_k} \oplus h(X_k \parallel MID_{S_k})$ and checks whether $GS_4^{S_k} \stackrel{?}{=} h(GS_1^{S'_k} \parallel X_k \parallel MID_{S_k})$; if this equation is valid, S_k stores $GS_2^{S_k}$ and x_k. Otherwise, S_k terminates the process by sending $GS_1^{S_k}$ as a response.

C. MP_I REGISTRATION PHASE

This phase outlines the registration process specifically designed for medical professionals, and it necessitates their physical presence at the center. The entirety of this registration process is conducted over private communication channels. The comprehensive steps of the registration for our proposed scheme are illustrated in Fig. 4.

1) First, MP_i inputs ID_{MP_i}, PW and imprints its own biometric information BIO_i in MD. Then, MP_i computes $GEN(BIO_i) = \{\alpha_i, \beta_i\}$. After that, MD chooses a random value p_i and generates P_i, such that $P_i = \mathrm{PUF}(p_i)$
2) MD computes $MID_{MP_i} = h(ID_{MP_i} \parallel P_i)$ and $MPW = h(PW \parallel \alpha_i)$. Then, MP_i transmits $\{ID_{MP_i}, P_i, MID_{MP_i}, MPW\}$ to GWN_j.
3) When GWN_j gets $\{ID_{MP_i}, P_i, MID_{MP_i}, MPW\}$, it computes $MID'_{MP_i} = h(ID_{MP_i} \parallel P_i)$ and generates a random value SCN_1. Then, GWN_j calls the $MPReister$ function within SC with $\{h(MID_{MP_i} \parallel SCN_1), h(MID'_{MP_i} \parallel SCN_1), ID_{MP_i}\}$. Then, SC checks for the duplication of ID_{MP_i}. If ID_{MP_i} is duplicated, it terminates the registration process; if not, SC checks $h(MID'_{MP_i} \parallel SCN_1) \stackrel{?}{=} h(MID_{MP_i} \parallel SCN_1)$ if the condition is not met, terminates the process; otherwise, SC registers MD with ID_{MP_i} on the blockchain. Then, GWN_j generates random value SK_{MP_i} and computes $PG = PUF(g_j)$, $GM_0^{MP_i} = PUF(h(ID_{MP_i} \parallel MPW \parallel P_i))$,$GM_1^{MP_i} = h(g_j \parallel MID_{MP_i} \parallel SK_{MP_i})$, $GM_2^{MP_i} = GM_1^{MP_i} \oplus MID_{MP_i} \oplus MPW$, $GM_3^{MP_i} = h(P_i \parallel MID_{MP_i} \parallel MPW \parallel GM_1^{MP_i})$, $GM_4^{MP_i} = GM_1^{MP_i} \oplus PG \oplus h(g_j \parallel ID_{GWN})$. After that, GWN_j transmits $\{GM_2^{MP_i}, GM_3^{MP_i}\}$ and stores $\{GM_4^{MP_i}, MID_{MP_i}\}$ with $GM_0^{MP_i}$ as key in local database.
4) After MP_i gets $\{GM_2^{MP_i}, GM_3^{MP_i}\}$, MP_i computes $GM_1^{MP'_i} = GM_2^{MP_i} \oplus MID_{MP_i} \oplus MPW$. Using $GM_1^{MP'_i}$, $GM_3^{MP_i} \stackrel{?}{=} h(P_i \parallel MID_{MP_i} \parallel MPW \parallel GM_1^{MP'_i})$ is checked. If this equation is valid, MP_i stores $\{GM_2^{MP_i}, GM_3^{MP_i}, p_i\}$ in own MD. Otherwise, the process is terminated by sending $GM_1^{MP_i}$ as a response.

D. LOGIN AND AUTHENTICATION PHASE

This phase is executed over a public communication channel. After undergoing the login procedure, MP_i makes a request to GWN_j to access information related to S_k. Subsequently, during the authentication procedure, both MP_i and S_k can establish a session key based on the information they exchange. For every login and authentication process, the previously used session key is discarded, giving way to the creation of a new session key. A comprehensive outline of the login and authentication procedures of our scheme is depicted in Fig. 5.

1) First, MP_i inputs ID_{MP_i}, PW_i and imprints its own biometric information BIO'_i in MD. After that, MD computes $REP(BIO'_i, \beta_i) = \alpha_i$, $P_i = PUF(p_i)$, $MPW = h(PW_i \parallel \alpha_i)$ and $MID_{MP_i} = h(ID_{MP_i} \parallel P_i)$. Using MID_{MP_i}, MPW, it computes $GM_1^{MP_i}$, such that $GM_1^{MP_i} = GM_2^{MP_i} \oplus MID_{MP_i} \oplus MPW$. Next, MD computes $GM_3^{MP'_i} = h(P_i \parallel MID_{MP_i} \parallel MPW \parallel$

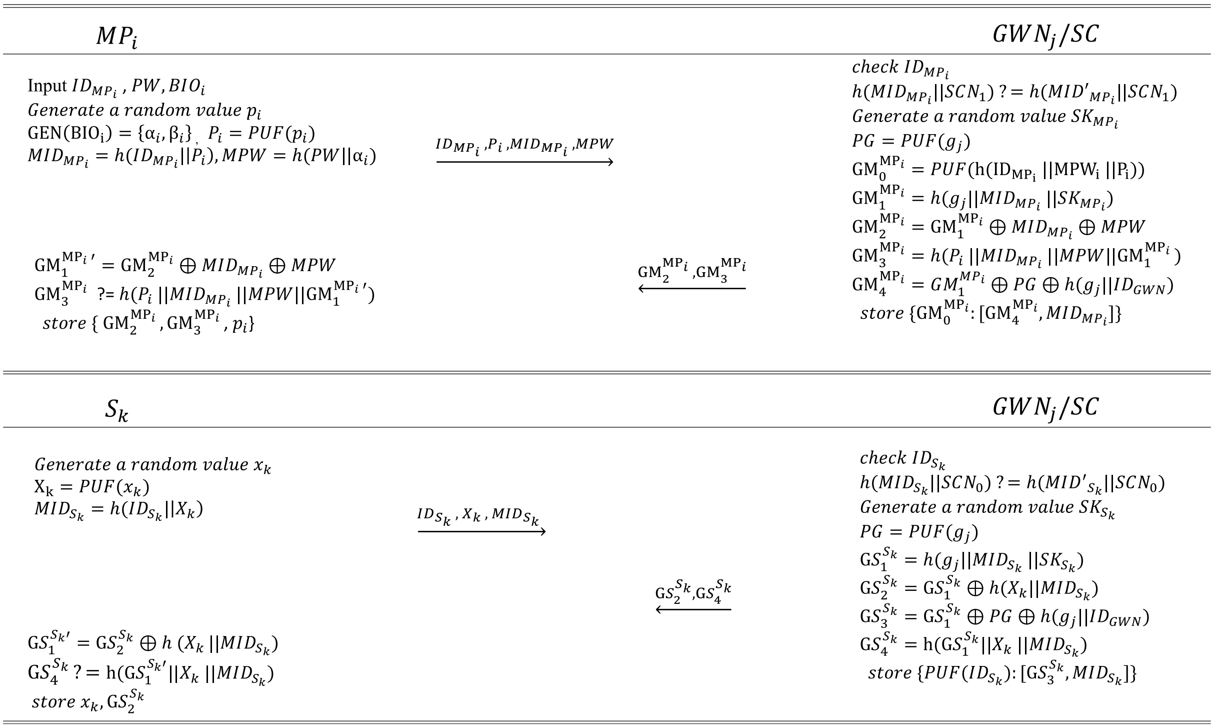

MP_i
GWN_j/SC
Input ID_{MP_i}, PW, BIO_i
Generate a random value p_i
$GEN(BIO_i) = \{\alpha_i, \beta_i\}$, $P_i = PUF(p_i)$
$MID_{MP_i} = h(ID_{MP_i}||P_i)$, $MPW = h(PW||\alpha_i)$
$ID_{MP_i}, P_i, MID_{MP_i}, MPW$
check ID_{MP_i}
$h(MID_{MP_i}||SCN_1) ?= h(MID'_{MP_i}||SCN_1)$
Generate a random value SK_{MP_i}
$PG = PUF(g_j)$
$GM_0^{MP_i} = PUF(h(ID_{MP_i}||MPW_i||P_i))$
$GM_1^{MP_i} = h(g_j||MID_{MP_i}||SK_{MP_i})$
$GM_2^{MP_i} = GM_1^{MP_i} \oplus MID_{MP_i} \oplus MPW$
$GM_3^{MP_i} = h(P_i||MID_{MP_i}||MPW||GM_1^{MP_i})$
$GM_4^{MP_i} = GM_1^{MP_i} \oplus PG \oplus h(g_j||ID_{GWN})$
store $\{GM_0^{MP_i}: [GM_4^{MP_i}, MID_{MP_i}]\}$
$GM_2^{MP_i}, GM_3^{MP_i}$
$GM_1^{MP_i}{}' = GM_2^{MP_i} \oplus MID_{MP_i} \oplus MPW$
$GM_3^{MP_i} ?= h(P_i||MID_{MP_i}||MPW||GM_1^{MP_i}{}')$
store $\{GM_2^{MP_i}, GM_3^{MP_i}, p_i\}$
S_k
GWN_j/SC
Generate a random value x_k
$X_k = PUF(x_k)$
$MID_{S_k} = h(ID_{S_k}||X_k)$
ID_{S_k}, X_k, MID_{S_k}
check ID_{S_k}
$h(MID_{S_k}||SCN_0) ?= h(MID'_{S_k}||SCN_0)$
Generate a random value SK_{S_k}
$PG = PUF(g_j)$
$GS_1^{S_k} = h(g_j||MID_{S_k}||SK_{S_k})$
$GS_2^{S_k} = GS_1^{S_k} \oplus h(X_k||MID_{S_k})$
$GS_3^{S_k} = GS_1^{S_k} \oplus PG \oplus h(g_j||ID_{GWN})$
$GS_4^{S_k} = h(GS_1^{S_k}||X_k||MID_{S_k})$
store $\{PUF(ID_{S_k}): [GS_3^{S_k}, MID_{S_k}]\}$
$GS_2^{S_k}, GS_4^{S_k}$
$GS_1^{S_k}{}' = GS_2^{S_k} \oplus h(X_k||MID_{S_k})$
$GS_4^{S_k} ?= h(GS_1^{S_k}{}'||X_k||MID_{S_k})$
store $x_k, GS_2^{S_k}$

FIGURE 4. Registration phase.

```
Algorithm 1 Validation Assistance Steps of $SC$
Input: two values to compare, $A$ and $A'$
if blockchain address of $GWN_j$ is incorrect then
    block the address that was requested and terminate process
    if the counter equals the predefined value then
        block the matching section and terminate process
    end if
end if
if $A$ equal $A'$ then
    do next step
else
    Increase the counter and retry the designated process
end if
```

$GM_1^{MP_i}$) and checks $GM_3^{MP'_i} \overset{?}{=} GM_3^{MP_i}$. If this equation holds, MP_i successfully logs into MD. If not, MD rejects the process.

2) MP_i selects ID_{S_k} and MD generates a random value RN_{MP_i}. Next, MD computes $M_0 = h(ID_{MP_i} \parallel MPW_i \parallel P_i)$, $M_1 = (ID_{S_k} \parallel RN_{MP_i}) \oplus H(GM_1^{MP_i} \parallel MID_{MP_i} \parallel T_1)$ and $M_2 = h(GM_1^{MP_i} \parallel MID_{MP_i} \parallel RN_{MP_i} \parallel T_1)$. After that, MP_i transmits the request $C_1 = \{M_0, M_1, M_2, T_1\}$ to GWN_j.

3) Upon receiving the message C_1, GWN_j checks the freshness of T_1. If T_1 is invalid, it rejects the process. If not, GWN_j checks for the existence of $GM_0^{MP'_i}$, such that $GM_0^{MP'_i} = PUF(M_0)$. If this value exists, $GM_4^{MP_i}$ and MID_{MP_i} are derived using $GM_0^{MP_i}$. Note that we store $\{GM_4^{MP_i}, MID_{MP_i}\}$ with $GM_0^{MP_i}$ as key in the local database. Then, GWN_j computes $GM_1^{MP_i} = GM_4^{MP_i} \oplus PG \oplus h(g_j \parallel ID_{GWN})$, $(ID_{S_k} \parallel RN_{MP_i}) = M_1 \oplus H(GM_1^{MP_i} \parallel MID_{MP_i} \parallel T_1)$, $M'_2 = h(GM_1^{MP_i} \parallel MID_{MP_i} \parallel T_1)$. After that, GWN_j calls the *MPAuthentication* function and builds a transaction with M_2 and M'_2 to SC. Then, SC checks the address of GWN_j in the blockchain network and confirms the counter of MP_i. If the address of GWN_j is invalid, or if the counter has reached the predetermined threshold (e.g., five times), SC blocks the matching section. After that SC compares M_2 and M'_2. If they do not match, it increments the counter, and MP_i attempts the authentication process once again.

4) If they do match, GWN_j generates a random value RN_{GWN_j} and computes $M_3 = RN_{GWN_i} \oplus h(GM_1^{MP_i} \parallel MID_{MP_i} \parallel RN_{MP_i} \parallel T_2)$, $M_4 = h(GM_1^{MP_i} \parallel MID_{MP_i} \parallel RN_{GWN_i} \parallel T_2)$ and transmits $C_2 = \{M_3, M_4, T_2\}$ to MP_i.

5) After receiving the message, C_2, MP_i checks the freshness of T_2. If T_2 can accept, MP_i calculates $RN_{GWN_j} = M_3 \oplus h(GM_1^{MP_i} \parallel MID_{MP_i} \parallel RN_{MP_i} \parallel T_2)$, $M'_4 = h(GM_1^{MP_i} \parallel MID_{MP_i} \parallel RN_{GWN_i} \parallel T_2)$ and checks $M'_4 \overset{?}{=} M_4$ to verify the validity of the message. If this equation holds true, it can be considered that GWN_j is legitimate.

MP_i $\quad GWN_j/SC$ $\quad S_k$

$input\ ID_{MP_i}, PW_{MP_i}, BIO_i$

$\alpha_i = REP(BIO_i, \beta_i)$

$\mathrm{P_i} = \mathrm{PUF(p_i)},\ \mathrm{MPW} = h(PW||\alpha),\ \mathrm{MID_{MP_i}} = \mathrm{h(ID_{MP_i}||P_i)}$

$\mathrm{G}M_1^{\mathrm{MP}_i} = \mathrm{G}M_2^{\mathrm{MP}_i} \oplus MID_{MP_i} \oplus MPW$

$check\ GM_3 ?= h(M_i||MID_{MP_i}||MPW||\mathrm{G}M_1^{\mathrm{MP}_i})$

$Generate\ a\ random\ value\ RN_{MP_i}$

$select\ ID_{S_k}$

$M_0 = h\left(ID_{MP_i}||MPW||P_i\right)$

$M_1 = (ID_{S_k}||RN_{MP_i}) \oplus H(\mathrm{GM}_1^{\mathrm{MP}_i}||MID_{MP_I}||T_1)$

$M_2 = h(\mathrm{G}M_1^{\mathrm{MP}_i}||MID_{MP_i}||RN_{MP_i}||T_1)$

$\xrightarrow{C_1=\{M_0,M_1,M_2,T_1\}}$

$check\ T_1$

$compute\ \mathrm{GM_0^{MP_i}} = \mathrm{PUF(M_0)}$

$find\ \mathrm{GM}_0^{\mathrm{MP}_i}\ and\ derive\ \mathrm{GM}_4^{\mathrm{MP}_i}\ and\ MID_{MP_i}$

$\mathrm{GM}_1^{\mathrm{MP}_i} = \mathrm{GM}_4^{\mathrm{MP}_i} \oplus PG \oplus h(g_j||ID_{GWN})$

$(ID_{S_k}||RN_{MP_i}) = M_1 \oplus H(GM_1^{MP_i}||MID_{MP_i}||T_1)$

$check\ M_2 ?= h(\mathrm{G}M_1^{\mathrm{MP}_i}||MID_{MP_i}||T_1)$

$Generate\ a\ random\ value\ RN_{GWN_j}$

$M_3 = RN_{GWN} \oplus \mathrm{h}(\mathrm{G}M_1^{\mathrm{MP}_i}||\mathrm{MID_{MP_i}}||\mathrm{RN_{MP_i}}||\mathrm{T_2})$

$M_4 = \mathrm{h}(\mathrm{G}M_1^{\mathrm{MP}_i}||\mathrm{MID_{MP_i}}||\mathrm{RN_{GWN}}||\mathrm{RN_{MP_i}}||\mathrm{T_2})$

$\xleftarrow{C_2=\{M_3,M_4,T_2\}}$

$check\ T_2$

$RN_{GWN} = M_3 \oplus \mathrm{h}(\mathrm{G}M_1^{\mathrm{MP}_i}||\mathrm{MID_{MP_i}}||\mathrm{RN_{MP_i}}||\mathrm{T_2})$

$check\ M_4 ?= \mathrm{h}(\mathrm{G}M_1^{\mathrm{MP}_i}||\mathrm{MID_{MP_i}}||\mathrm{RN_{GWN}}||\mathrm{RN_{MP_i}}||\mathrm{T_2})$

$find\ PUF\left(ID_{S_k}\right) and\ derive\ \mathrm{G}S_3^{S_k}, MID_{S_k}$

$compute\ \mathrm{G}S_1^{S_k} = \mathrm{G}S_3^{S_k} \oplus PG \oplus h(g_j||ID_{GWN})$

$M_5 = (RN_{GWN}||RN_{MP_i}) \oplus h(\mathrm{G}S_1^{S_k}||MID_{S_k}||T_3)$

$M_6 = MID_{MP_i} \oplus h(\mathrm{G}S_1^{S_k}||MID_{S_k}||RN_{GWN}||T_3)$

$M_7 = \mathrm{h}(\mathrm{MID_{MP_i}}||\mathrm{RN_{MP_i}}||\mathrm{RN_{GWN_j}}||\mathrm{G}S_1^{S_k}||\mathrm{MID}_{S_k}||\mathrm{T_3})$

$\xrightarrow{C_3=\{M_5,M_6,M_7,T_3\}}$

$check\ T_3$

$compute\ \mathrm{G}S_1^{S_k} = \mathrm{G}S_2^{S_k} \oplus h(X_k||MID_{S_k})$

$(RN_{GWN_j}||RN_{MP_i}) = M_5 \oplus h(\mathrm{G}S_1^{S_k}||MID_{S_k}||T_3)$

$MID_{MP_i} = M_6 \oplus h(\mathrm{G}S_1^{S_k}||MID_{S_k}||RN_{GWN}||T_3)$

$M_7 ?= \mathrm{h}(\mathrm{MID_{MP_i}}||\mathrm{RN_{MP_i}}||\mathrm{RN_{GWN}}||\mathrm{G}S_1^{S_k}||\mathrm{MID_{SD_j}}||\mathrm{T_3})$

$Generate\ a\ random\ value\ RN_{S_k}$

$KEY = H(MID_{MP_i}||MID_{S_k}||RN_{S_k}||RN_{MP_i})$

$M_8 = RN_{S_k} \oplus \mathrm{h}(\mathrm{MID_{S_k}}||\mathrm{G}S_1^{S_k}||\mathrm{RN_{GWN}}||\mathrm{T_4})$

$M_9 = \mathrm{h}(\mathrm{MID_{S_k}}||\mathrm{RN}_{S_k}||\mathrm{RN}_{\mathrm{GWN}_j}||\mathrm{G}S_1^{S_k}||\mathrm{T_4})$

$M_{10} = h(MID_{S_k}||KEY||RN_{GWN_j})$

$\xleftarrow{C_4=\{M_8,M_9,\ M_{10}\ T_4\}}$

$check\ T_4$

$RN_{S_k} = M_8 \oplus \mathrm{h}(\mathrm{MID_{S_k}}||\mathrm{G}S_1^{S_k}||\mathrm{RN_{GWN}}||\mathrm{T_4})$

$check\ M_9 ?= \mathrm{h}(\mathrm{MID_{S_k}}||\mathrm{RN_{MP_i}}||\mathrm{RN_{GWN}}||\mathrm{G}S_1^{S_k}||\mathrm{T_4})$

$M_{11} = (MID_{S_k}||RN_{S_k}) \oplus H(RN_{GWN_j}||RN_{MP_i}||\mathrm{GM}_1^{\mathrm{MP}_i}||T_5)$

$\xleftarrow{C_5=\{M_{10},M_{11},T_5\}}$

$check\ T_5$

$(MID_{S_k}||RN_{S_k}) = M_{11} \oplus H(RN_{GWN}||RN_{MP_i}||\mathrm{GM}_1^{\mathrm{MP}_i}||T_5)$

$KEY = H(MID_{MP_i}||MID_{S_k}||RN_{S_k}||RN_{MP_i})$

$check\ M_{10} ?= h(MID_{S_k}||KEY||RN_{GWN_j})$

FIGURE 5. Login and authentication phase.

6) Simultaneously, upon sending a message to MP_i, GWN_j derives $\{GS_3^{S_k}, MID_{S_k}\}$ using $PUF(ID_{S_k})$ and computes $GS_1^{S_k} = GS_3^{S_k} \oplus PG \oplus h(g_j \parallel ID_{GWN_j})$, $M_5 = (RN_{GWN_j} \parallel RN_{MP_i}) \oplus h(GS_1^{S_k} \parallel MID_{S_k} \parallel T_3)$, $M_6 = MID_{MP_i} \oplus h(GS_1^{S_k} \parallel MID_{S_k} \parallel RN_{GWN_j} \parallel T_3)$, $M_7 = h(MID_{MP_i} \parallel RN_{MP_i} \parallel RN_{GWN_j} \parallel GS_1^{S_k} \parallel$

$MID_{S_k} \| T_3)$. Then, GWN_j transmits the message $C_3 = \{M_5, M_6, M_7, T_3\}$ to S_k.

7) After receiving the message C_3, S_k checks freshness of T_3. If T_3 can accept, S_k computes $X_k = PUF(x_k)$, $MID_{S_k} = h(ID_{S_k} \| X_k)$, $GS_1^{S_k} = GS_2^{S_k} \oplus h(X_k \| MID_{S_k})$. After that, S_k (using secret parameter $GS_1^{S_k}$) computes $(RN_{GWN_j} \| RN_{MP_i}) = M_5 \oplus h(GS_1^{S_k} \| MID_{S_k} \| T_3)$, $MID_{MP_i} = M_6 \oplus h(GS_1^{S_k} \| MID_{S_k} \| RN_{GWN_j} \| T_3)$, $M'_7 = h(MID_{MP_i} \| RN_{MP_i} \| RN_{GWN_j} \| GS_1^{S_k} \| MID_{S_k} \| T_3)$. After that, S_k checks $M_7 \overset{?}{=} M'_7$; if this equation holds, S_k generates a random value RN_{S_k} and computes $KEY = H(MID_{MP_i} \| MID_{S_k} \| RN_{S_k} \| RN_{MP_i})$, $M_8 = RN_{S_k} \oplus h(MID_{S_k} \| GS_1^{S_k} \| RN_{GWN_j} \| T_4)$, $M_9 = h(MID_{S_k} \| RN_{MP_i} \| RN_{GWN_j} \| GS_1^{S_k} \| T_4)$, $M_{10} = h(MID_{S_k} \| KEY \| RN_{GWN_j})$. Then, S_k transmits the message $C_4 = \{M_8, M_9, M_{10}, T_4\}$ to GWN_j.

8) When GWN_j receives the message C_4, GWN_j checks whether the T_4 is valid. If it is valid, GWN_j computes $RN_{S_k} = M_8 \oplus h(MID_{S_k} \| GS_1^{S_k} \| RN_{GWN_j} \| T_4)$ and $M'_9 = h(MID_{S_k} \| RN_{MP_i} \| RN_{GWN_j} \| GS_1^{S_k} \| T_4)$. After that GWN_j calls *SAuthentication* function and transmits $\{M_9, M'_9\}$ to SC. If SC receives these values, it checks the address of GWN_j in the blockchain network and confirms the counter of S_k. If the address of the GWN_j is invalid, or if the counter has reached the predetermined threshold (e.g., five times), SC blocks the matching section. Next, SC compares M_9 and M'_9. If they do not match, it increments the counter, and S_k attempts the process of generating a random value again.

9) If they do match, GWN_j computes $M_{11} = (MID_{S_k} \| RN_{S_k} \oplus H(RN_{GWN_j} \| RN_{MP_i} \| GM_1^{MP_i} \| T_5)$ and transmits the message $C_5 = \{M_{10}, M_{11}, T_5\}$ to MP_i.

10) Upon receiving the message C_5, MP_i checks the freshness of T_5. If T_5 is valid, MP_i computes $(MID_{S_k} \| RN_{S_k}) = H(RN_{GWN_j} \| RN_{MP_i} \| GM_1^{MP_i} \| T_5)$ and $KEY = H(MID_{MP_i} \| MID_{S_k} \| RN_{S_k} \| RN_{MP_i})$. Then, MP_i checks $M_{10} \overset{?}{=} h(MID_{S_k} \| KEY \| RN_{GWN_j})$. If this equation is correct, MP_i can use this KEY as a session key while receiving information from S_k.

E. PASSWORD AND BIOMETRIC INFORMATION CHANGE PHASE

If a registered MP_i wants to change their password and biometric information for safer use, they can change it via the following process.

1) MP_i inputs its own ID_{MP_i}, PW, and BIO'_i. Then, MD computes $REP(BIO'_i, \beta_i) = \alpha_i$, $P_i = PUF(p_i)$, $MPW_i = h(PW \| \alpha_i)$, and $MID_{MP_i} = h(ID_{MP_i} \| P_i)$. Using MID_{MP_i}, MPW, MD computes $GM_1^{MP_i}$, such that $GM_1^{MP_i} = GM_2^{MP_i} \oplus MID_{MP_i} \oplus MPW$, $GM_3^{MP'_i} = h(P_i \| MID_{MP_i} \| MPW_i \| GM_1^{MP_i})$ and checks $GM_3^{MP'_i} \overset{?}{=} GM_3^{MP_i}$. If this equation holds, MP_i can change the password and biometric information.

2) MP_i inputs new password PW^{new} and imprints BIO_i^{new} through a secure channel. Then, MD computes $GEN(BIO_i) = \{\alpha_i^{new}, \beta_i^{new}\}$, $MPW_i^{new} = h(PW_i^{new} \| \alpha_i^{new})$, $newGM_2^{MP_i} = GM_1^{MP_i} \oplus MID_{MP_i} \oplus MPW^{new}$, and $newGM_3^{MP_i} = h(P_i \| MID_{MP_i} \| MPW \| GM_1^{MP_i})$. Then, MD replaces $GM_2^{MP_i}$ and $GM_3^{MP_i}$ with $newGM_2^{MP_i}$ and $newGM_3^{MP_i}$. If the value has been successfully changed, then this process is completed.

F. DATA TRANSFER PHASE

If MP_i wants to automatically acquire additional data from a specific sensor node under specific conditions, they have the option to incorporate an auxiliary function within SC. This facilitates the automatic retrieval of information from the said sensor node under specific situations. Subsequently, the legitimacy of the SC written by MP_i is verified by the blockchain participants. When the contract is executed, only the essential information required for the contract is transferred from S_k to SC. Therefore, it is incumbent upon MP_i to write the SC to prevent the patient's sensitive information from being disclosed. Moreover, whenever S_k is prompted to send out the patient's personal information upon the request of MP_i or by the execution of a SC, it is paramount to first encrypt this data. The encryption is to be performed using the session key, which is previously established during the authentication process through symmetric encryption methods such as AES. This encrypted data is then dispatched to MP_i. It is noteworthy that a shared session key between MP_i and S_k exists as long as MP_i remains logged in. In addition, if MP_i is not logged in, it is crucial that the SC is implemented such that it cannot be executed.

G. SENSOR ADDITION PHASE

In instances where the employed device encounters technical glitches or if there is a need to replace it for any reason, a new sensor node can be registered to replace the existing one. In this case, since the existing sensor ID remains registered in within the blockchain network, it is vital that the ID of the new sensor node is distinguishable from the ID of the sensor it succeeds.

IV. SECURITY ANALYSIS

In this section, we conduct a security analysis that is broadly divided into two parts: informal security analysis and formal security analysis. Further details are as follows:

A. INFORMAL SECURITY ANALYSIS

In this section, we undertake an informal security analysis to demonstrate how our proposed protocol is secure from various types of attacks and satisfies key security characteristics. Table 2 provides a comparison of the security features our scheme offers against those in previous studies. This comparison clearly indicates that our scheme fulfills

TABLE 2. Comparison of security features.

Security Features	Proposed scheme	Wu et al. [7]	Fotouhi et al. [9]	Wang et al. [13]	Yu et al. [14]
Resistance to Replay Attack	O	X	O	O	O
Resistance to Physical Capture Attack	O	O	X	X	X
Resistance to Insider Attack	O	O	O	X	X
Resistance to Man-In-The-Middle Attack	O	O	O	X	X
Resistance to Offline Guessing Attack	O	X	O	O	O
Resistance to Stolen Verifier Attack	O	O	X	X	X
Resistance to Impersonation Attack	O	X	O	O	O
Resistance to SPOF	O	X	X	O	O
Perfect Forward Secrecy	O	O	O	O	O
Confidentiality	O	O	O	X	X
Mutual Authentication	O	O	O	X	X

several security characteristics that previous research did not sufficiently cover.

1) *Confidentiality*: Given that the communication process over public channels and the transmission of information to the SC are publicly visible, there is a necessity to protect participants' data. In our scheme, we have obscured participant details and secret keys through the application of nonce, XOR, and hash functions. Therefore, our protocol ensures data confidentiality.

2) *Mutual Authentication*: In our approach, GWN_j interprets M_1 and confirms the value of M_2 from the SC to authenticate MP_i. MP_i interprets M_3 and verifies M_4 to authenticate GWN_j. Through these processes, mutual authentication is achieved between GWN_j and MP_i. Similarly, S_k interprets M_5 and M_6 then verify M_7 to authenticate GWN_j. Subsequently, GWN_j interprets M_8 and computes M_9 to ascertain mutual authentication with GWN_j. By comparing M_{10}, both MP_i and S_k can authenticate each other if the KEY value generated by S_k matches the received information. This procedure signifies the ability of our protocol to facilitate mutual authentication among the involved participant.

3) *Replay Attack*: Malevolent actors might try to capture and resend messages from public channels to mimic valid participants or cause network traffic. However, our design incorporates a timestamp T_x ensuring processes only proceed when the timestamp value is valid. This inclusion defends against replay attacks.

4) *Insider Attack*: During the registration phase, the user transfers $MPW = h(PW \parallel \alpha_i)$ to the GWN_j. This means a malicious GWN_i cannot ascertain the user's password. With MD registered on the blockchain, impersonation of MP_i is feasible. This design element strengthens our scheme against insider attacks.

5) *Man-In-The-Middle Attack*: Attackers can intercept communications to acquire information, resend them, inject other information, or delete information. However, our design facilitates the generation of shared secret key $GM_1^{MP_i}$ and $GS_1^{S_k}$ between communicating participant using random values $SK_{MP_i} and SK_{S_k}$. Without knowledge of these values, attackers are unable to obtain information, or pretend to be a legitimate user Therefore, the scheme we proposed is secure against MITM attacks.

6) *Offline Password Guessing Attack*: Assuming an attacker obtains MP_i's MD, they might execute an offline password guessing attack to obtain MP_i's secret information, $GM_1^{MP_i}$. However, because the scheme uses a masked password $MPW{=}h(PW \parallel \alpha_i)$ derived from user biometrics rather than the direct password, it is computationally infeasible for the attacker. Even if an attacker deciphers the MPW, they cannot impersonate a legitimate user without comprehending the PUF pair and consequently deriving the essential element $GM_1^{MP_i}$. Therefore, our scheme stands robust against offline password guessing attacks.

7) *Physical Capture Attack*: An attacker might attempt physical capture attack to extract information from the sensor node S_K. However, since we utilize PUF to generate X_k, extracting the full value of x_k becomes arduous even if the attacker manages to obtain a value of x_k, due to PUF's inherent properties As a result, the attacker cannot retrieve the essential value $GS_1^{S_k}$, necessary for session key generation, hindering them from discerning the session key generated between MP_i and S_k. The distinct operational properties of each PUF also make interpreting it and determining the session key highly inefficient. Our proposed scheme remains secure against physical capture attacks.

8) *Stolen Verifier Attack*: In the event of a stolen verifier attack, in order to impersonate a legitimate user participating in the communication, assume an accesses the local database of GWN_j and steals information such as $\{GM_0^{MP_i}: [GM_4^{MP_i}, MID_{MP_i}]\}$, $\{PUF(ID_{S_k}) : [GS_3^{S_k}, MID_{S_k}]\}$. However, it is extremely difficult to generate essential information for each communication, such as $GM_1^{MP_i}$ and $GS_1^{S_k}$, from this stolen information. The protective measures in our scheme, GWN's secret keys g, PG, and XOR, as well as hash functions keep these values safe. Additionally, to specify the MP_i, an attacker must interpret the GWN_j's PUF pair, which varies based on the IC's intrinsic physical properties, making the task

formidable. Our design effectively defends against the stolen verifier attack.

9) *Perfect Forward Secrecy*: Our protocol mandates the generation of a fresh session *KEY* for every authentication. If a past session key gets compromised, it cannot decrypt the information encrypted with the current session key. Moreover, deriving the session key requires knowledge of values $GM_1^{MP_i}$ and $GS_1^{S_k}$. Obtaining these, along with nonce details RN_{MP_i}, RN_{S_k} and masked *ID* values MID_{MP_i}, MID_{S_k}, which are masked using the value generated through *PUF* is extremely difficult, ensuring the scheme's perfect forward secrecy.

10) *Impersonation Attack*: During registration, our design sets up the secret keys $GM_1^{MP_i}$ and $GS_1^{S_k}$ shared between GWN_j and MP_i as well as GWN_j and S_k. To launch an impersonation attack, these keys are imperative. Given their composition—incorporating random value *SK*, masking identifier *MID*, and the secret key of GWN_j, *g*, so it is difficult to deduce them, even if the information exchanged in the public channel is stolen. This makes our protocol resolute against impersonation attacks.

11) *Scalability*: Our proposed scheme can maintain stable performance even as the number of medical sensors increases. When a new medical sensor, S_k, is registered in the system, it is managed through blockchain network, allowing many medical devices to be easily registered and managed. However, the increase in sensors involves a rise in the time required to derive and validate parameters needed for the authentication process. To address this, our proposed scheme utilizes specific values such as $GM_0^{MP_i}$, $PUF(ID_{S_k})$ to efficiently derive authentication parameters, aiming to minimize the communication waiting time that may occur with network expansion. Furthermore, considering the various loads that can occur in GWN_j, the proposed scheme can use the blockchain network to distribute the workload among multiple servers. If one server becomes overloaded, another server takes over the task, thus ensuring a response to various workload situations.

B. FORMAL SECURITY ANALYSIS USING BAN LOGIC

In this section, we employ BAN logic, a widely recognized security analysis technique used in numerous studies such as [30] and [31], to conduct an exhaustive evaluation of our proposed scheme. By leveraging the thorough framework of BAN logic, which encompasses analysis rules, goals, assumptions, and derivations, we are able to rigorously assess the scheme's robustness against various security vulnerabilities. Notations for BAN logic are listed in table 3.

1) BAN LOGIC RULES

The mentioned rules (R1)~(R5) are foundational concepts used to analysis our protocols. These rules aim to capture various aspects of trust, data integrity, and authority in our distributed system.

TABLE 3. Notation for BAN logic.

Notation	Description
A, B	Two participants
M, N, P	Parameters
KEY	Session key
T	Timestamp
$A \mid\equiv B$	A believes B
$A \lhd M$	A sees M
$A \mid\sim B$	A sent B
$A \Rightarrow M$	A controls M
$\sharp(M)$	Fresh M
$A \overset{KEY}{\longleftrightarrow} B$	A and B have shared key KEY
$\xrightarrow{P} A$	P is public key of A
$(M)_{KEY}$	Parameter M is hashed/encrypted by K

- (R1) Message Meaning Rule (MMR)

$$A \mid\equiv A \overset{KEY}{\longleftrightarrow} B, A \lhd \{M\}_{KEY}$$

- (R2) Freshness Rule (FR)

$$\frac{A \mid\equiv \sharp(M)}{A \mid\equiv \sharp(M, N)}$$

- (R3) Nonce Verification Rule (NVR)

$$\frac{A \mid\equiv \sharp(M), A \mid\equiv B \mid\sim M}{A \mid\equiv B \mid\sim M}$$

- (R4) Belief Rule (BR)

$$\frac{A \mid\equiv (M, N)}{A \mid\equiv M}$$

- (R5) Jurisdiction Rule (JR)

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

2) BAN LOGIC GOALS

According to the analytical procedures of BAN logic, our protocol has successfully achieved mutual authentication and secure session key establishment, and will therefore satisfy the following goals.

- (G1) $MD \mid\equiv (ID_{MP_i}, PW, BIO_i)$
- (G2) $MD \mid\equiv (GM_2^{MP_i}, GM_3^{MP_i})$
- (G3) $GWN_j \mid\equiv RN_{MP_i}$
- (G4) $SC \mid\equiv (M_2)$
- (G5) $MP_i \mid\equiv (RN_{GWN_j})$
- (G6) $MP_i \mid\equiv (M_4)$
- (G7) $S_k \mid\equiv (RN_{GWN_j}, RN_{MP_i})$
- (G8) $S_k \mid\equiv (M_7)$
- (G9) $GWN_j \mid\equiv (RN_{S_k})$
- (G10) $GWN_j \mid\equiv (M_9)$
- (G11) $MP_i \mid\equiv (MID_{S_k}, RN_{S_k})$
- (G12) $MP_i \mid\equiv (KEY)$

3) IDEALIZED FORM

The idealized message form of the protocol we have proposed is as follows.

- (M1) $MP_i \rightarrow GWN_j : \{M_0, M_1, M_2, T_1\}$
- (M2) $GWN_j \rightarrow MP_i : \{M_3, M_4, T_2\}$
- (M3) $GWN_j \rightarrow S_k : \{M_5, M_6, M_7, T_3\}$
- (M4) $S_k \rightarrow GWN_j : \{M_8, M_9, M_{10}, T_4\}$
- (M5) $GWN_j \rightarrow MP_i : \{M_{10}, M_{11}, T_5\}$

4) ASSUMPTIONS

Based on the environment of our protocol, we have established the following basic assumptions to exhaustively analyze our scheme.

- (A1) $GWN_j \mid\equiv \sharp(T_1)$
- (A2) $MP_i \mid\equiv \sharp(T_2)$
- (A3) $S_k \mid\equiv \sharp(T_3)$
- (A4) $GWN_j \mid\equiv \sharp(T_4)$
- (A5) $MP_i \mid\equiv \sharp(T_5)$
- (A6) $MP_i \mid\equiv \sharp(\beta_i)$
- (A7) $MP_i \mid\equiv \sharp(RN_{MP_i})$
- (A8) $GWN_j \mid\equiv \sharp(RN_{GWN_j})$
- (A9) $S_k \mid\equiv \sharp(RN_{S_k})$
- (A10) $S_K \mid\equiv \sharp(x_k)$
- (A11) $MD \mid\equiv MD \underset{\longleftrightarrow}{p_i} MP_i$
- (A12) $GWN_j \mid\equiv \{GWN_j \underset{\longleftrightarrow}{MID_{MP_i}} MP_i\}$
- (A13) $S_k \mid\equiv S_k \underset{\longleftrightarrow}{GS_2^{S_k}} GWN_j$
- (A14) $GWN_j \mid\equiv \xrightarrow{ID_{S_k}} S_k$
- (A15) $MP_i \mid\equiv GWN_j \Rightarrow RN_{GWN_j}$
- (A16) $MP_i \mid\equiv S_k \Rightarrow RN_{S_k}$
- (A17) $S_k \mid\equiv MP_i \Rightarrow RN_{MP_i}$
- (A18) $S_k \mid\equiv GWN_j \Rightarrow RN_{GWN_j}$

5) DERIVATION

Based on the aforementioned assumptions and the fundamental premises of BAN logic, we analyze the idealized form of messages in the proposed scheme and provide the following main proof procedures.

- (D1) $MD \triangleleft (ID_{MP_i}, PW, BIO_i)$
- (D2) $MD \mid\equiv MP_i \mid\equiv (ID_{MP_i}, PW, BIO_i)$

$$by\ (D1), (A11), (A6)$$

- (D3) $MD \mid\equiv (ID_{MP_i}, PW, BIO_i)$

$$by\ (D2)$$

- (D4) $MD \mid\equiv (GM_2^{MP_i}, GM_3^{MP_i})$

$$by\ (D3)$$

From derivation (D1)∼(D4), we achieve (G1), (G2) and MP_i is able to log in MD.

- (D5) $GWN_j \triangleleft (M_0, M_1, M_2, T_1)$

$$by\ M1$$

- (D6) $GWN_j \mid\equiv MP_i \mid\equiv (M_0, M_1, M_2, T_1)$

$$by\ (D5), (A1)$$

- (D7) $GWN_j \mid\equiv (GM_4^{MP_i}, MID_{MP_i})$

$$by\ (D6)$$

- (D8) $GWN_j \mid\equiv RN_{MP_i}$

$$by\ (D7)$$

- (D9) $SC \mid\equiv M_2$

$$by\ (D8), (A12)$$

From derivation (D5)∼(D9), we achieve (G3), (G4) and GWN_j believes that MP_i is legitimate.

- (D10) $MP_i \triangleleft (M_3, M_4, T_2)$
- (D11) $MP_i \mid\equiv GWN_j \mid\equiv (M_3, M_4, T_2)$

$$by\ (D10), (A2)$$

- (D12) $MP_i \mid\equiv M_3$

$$by\ (D11), (A7)$$

- (D13) $MP_i \mid\equiv RN_{GWN_j}$

$$by\ (D12), (A15)$$

- (D14) $MP_i \mid\equiv M_4$

$$by\ (D13)$$

From derivation (D10)∼(D14), we achieve (G5), (G6) and MP_i believes that GWN_j is legitimate.

- (D15) $S_k \triangleleft (M_5, M_6, M_7, T_3)$
- (D16) $S_k \mid\equiv GWN_j \mid\equiv (M_5, M_6, M_7, T_3)$

$$by\ (D15), (A3)$$

- (D17) $S_k \mid\equiv GWN_j \mid\equiv (RN_{GWN_j}, RN_{MP_i})$

$$by\ (D16), (A10), (A13)$$

- (D18) $S_k \mid\equiv (RN_{GWN_j}, RN_{MP_i})$

$$by\ (D17), (A17), (A18)$$

- (D19) $S_k \mid\equiv M_7$

$$by\ (D18)$$

From derivation (D15)∼(D19), we achieve (G7), (G8) and S_k believes that GWN_j is legitimate.

- (D20) $GWN_j \triangleleft (M_8, M_9, M_{10}, T_4)$
- (D21) $GWN_j \mid\equiv S_k \mid\equiv (M_8, M_9, M_{10}, T_4)$

$$by\ (D20), (A4)$$

- (D22) $GWN_j \mid\equiv S_k \mid\equiv RN_{S_k}$

$$by\ (D21), (A8), (A14)$$

- (D23) $GWN_j \mid\equiv M_9$

$$by\ (D22)$$

From derivation (D20)~(D23), we achieve (G9), (G10) and GWN_j believes that S_k is legitimate.

- (D24) $MP_i \lhd (M_{10}, M_{11}, T_5)$
- (D25) $MP_i \mid\equiv GWN_j \mid\equiv (M_{10}, M_{11}, T_5)$

  $$by\ (D24), (A5)$$

- (D26) $MP_i \mid\equiv (MID_{S_k}, RN_{S_k})$

  $$by\ (D25), (A7), (A16)$$

- (D27) $MP_i \mid\equiv S_k \mid\equiv KEY$

  $$by\ (D26), (A9)$$

- (D28) $MP_i \mid\equiv KEY$

  $$by\ (D27)$$

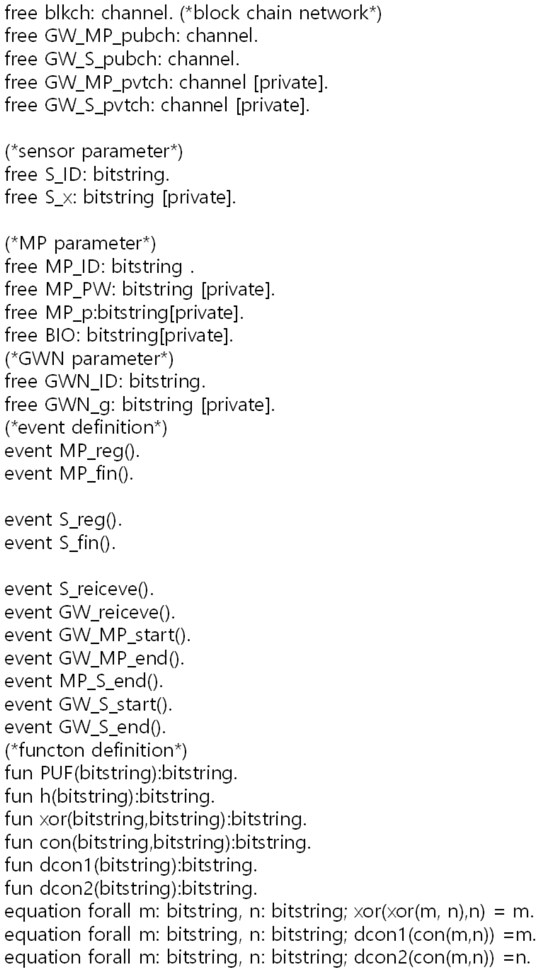
From derivation (D24)~(D28), we achieve (G11), (G12) and MP_i is able to secretly share session key KEY with S_k.

In summary, our protocol not only ensures the secure generation of a session key between MP_i and S_k but also demonstrates the capability to mutually authenticate participants.

C. FORMAL SECURITY ANALYSIS USING ProVerif

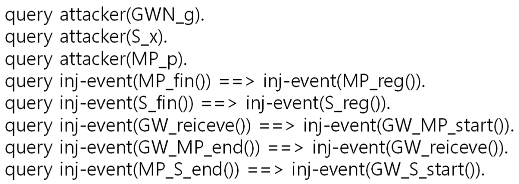
This section employs ProVerif, a widely recognized formal analysis tool featured in studies such as [32] and [33], to conduct an exhaustive evaluation. ProVerif [34] provides an environment that facilitates operations such as XOR, concatenation, and hash functions. It empowers the verification of our proposed protocol against unlimited data, deeming ProVerif an apt choice for validating our proposed approach. Through this, we evaluate the assurance of mutual authentication among communication participants and the confidentiality of critical parameters in our proposed scheme.

First, we describe the functions and parameters showcased in Fig. 6 implemented in our scheme. The ProVerif code seen in Fig. 6 helps determine adversary capabilities and equivalence verifications. Fig. 7 defines the "query attacker()" and "query inj-event() ==> inj-event()". The "query attacker()" inspects if the attacker, leveraging messages acquired from public channels, can compute the private keys of communication participants. Conversely, the query "inj-event() ==> inj-event()" ensures the proposed protocol's correct operation by scrutinizing the event sequence. Fig. 8 shows the process handled by S_k, where S_k conducts the registration and authentication phases. Fig. 10 depicts the process handled by MP_i. Here, MP_i goes through the registration, authentication, and session key verification phases. Both Figs.9 and 11 illustrate the process undertaken by GWN_j. Given that GWN_j processes the registration and authentication of both S_k and MP_i centrally, it handles two registration phases and three authentication phases. We can see that an "event()" has been set up in the process for each communication participant to check mutual authentication. If the 'event()' we have set operates according to a predetermined sequence, then 'inj-event() ==> inj-event()' becomes true, and we can determine whether the protocol we have proposed is capable of performing mutual authentication.



```
free blkch: channel. (*block chain network*)
free GW_MP_pubch: channel.
free GW_S_pubch: channel.
free GW_MP_pvtch: channel [private].
free GW_S_pvtch: channel [private].

(*sensor parameter*)
free S_ID: bitstring.
free S_x: bitstring [private].

(*MP parameter*)
free MP_ID: bitstring .
free MP_PW: bitstring [private].
free MP_p:bitstring[private].
free BIO: bitstring[private].
(*GWN parameter*)
free GWN_ID: bitstring.
free GWN_g: bitstring [private].
(*event definition*)
event MP_reg().
event MP_fin().

event S_reg().
event S_fin().

event S_reiceve().
event GW_reiceve().
event GW_MP_start().
event GW_MP_end().
event MP_S_end().
event GW_S_start().
event GW_S_end().
(*functon definition*)
fun PUF(bitstring):bitstring.
fun h(bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring.
fun con(bitstring,bitstring):bitstring.
fun dcon1(bitstring):bitstring.
fun dcon2(bitstring):bitstring.
equation forall m: bitstring, n: bitstring; xor(xor(m, n),n) = m.
equation forall m: bitstring, n: bitstring; dcon1(con(m,n)) =m.
equation forall m: bitstring, n: bitstring; dcon2(con(m,n)) =n.
```

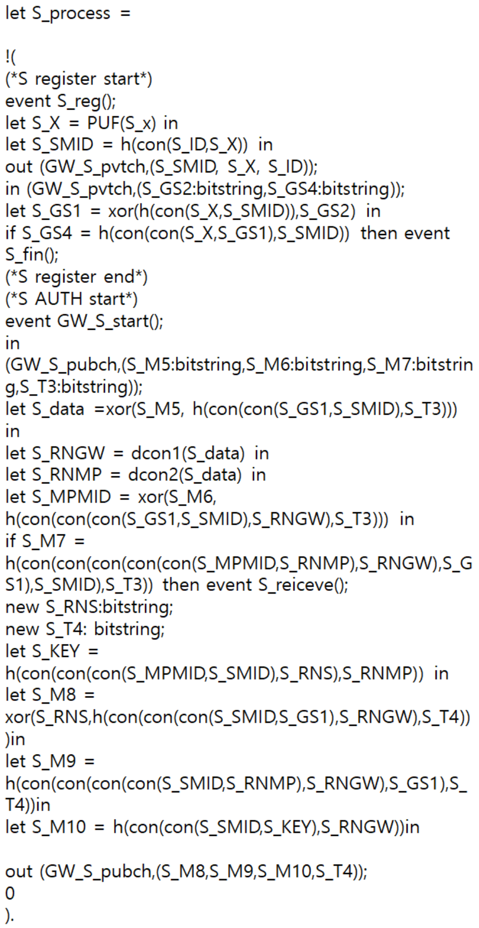
FIGURE 6. ProVerif code for definition.



```
query attacker(GWN_g).
query attacker(S_x).
query attacker(MP_p).
query inj-event(MP_fin()) ==> inj-event(MP_reg()).
query inj-event(S_fin()) ==> inj-event(S_reg()).
query inj-event(GW_reiceve()) ==> inj-event(GW_MP_start()).
query inj-event(GW_MP_end()) ==> inj-event(GW_reiceve()).
query inj-event(MP_S_end()) ==> inj-event(GW_S_start()).
```

FIGURE 7. ProVerif code for event.

The verification results are presented in Fig. 12. Information related to the query not attacker() indicates that attempts to breach our set private key were unsuccessful. Conversely, data related to the query inj-event() signifies that our verification has been appropriately implemented in the proposed protocol. In summary, the protocol we proposed successfully met the security requirements against known attacks when tested based on the Dolev-Yao attack model, and it also enabled the establishment of a secure session key. Additionally, we demonstrated the feasibility of mutual authentication among the communication



```
let S_process =

!(
(*S register start*)
event S_reg();
let S_X = PUF(S_x) in
let S_SMID = h(con(S_ID,S_X)) in
out (GW_S_pvtch,(S_SMID, S_X, S_ID));
in (GW_S_pvtch,(S_GS2:bitstring,S_GS4:bitstring));
let S_GS1 = xor(h(con(S_X,S_SMID)),S_GS2) in
if S_GS4 = h(con(con(S_X,S_GS1),S_SMID)) then event
S_fin();
(*S register end*)
(*S AUTH start*)
event GW_S_start();
in
(GW_S_pubch,(S_M5:bitstring,S_M6:bitstring,S_M7:bitstrin
g,S_T3:bitstring));
let S_data =xor(S_M5, h(con(con(S_GS1,S_SMID),S_T3)))
in
let S_RNGW = dcon1(S_data) in
let S_RNMP = dcon2(S_data) in
let S_MPMID = xor(S_M6,
h(con(con(con(S_GS1,S_SMID),S_RNGW),S_T3))) in
if S_M7 =
h(con(con(con(con(con(S_MPMID,S_RNMP),S_RNGW),S_G
S1),S_SMID),S_T3)) then event S_reiceve();
new S_RNS:bitstring;
new S_T4: bitstring;
let S_KEY =
h(con(con(con(S_MPMID,S_SMID),S_RNS),S_RNMP)) in
let S_M8 =
xor(S_RNS,h(con(con(con(S_SMID,S_GS1),S_RNGW),S_T4))
)in
let S_M9 =
h(con(con(con(con(S_SMID,S_RNMP),S_RNGW),S_GS1),S_
T4))in
let S_M10 = h(con(con(S_SMID,S_KEY),S_RNGW))in

out (GW_S_pubch,(S_M8,S_M9,S_M10,S_T4));
0
).
```

FIGURE 8. **ProVerif code for S_k in registration and authentication phase.**

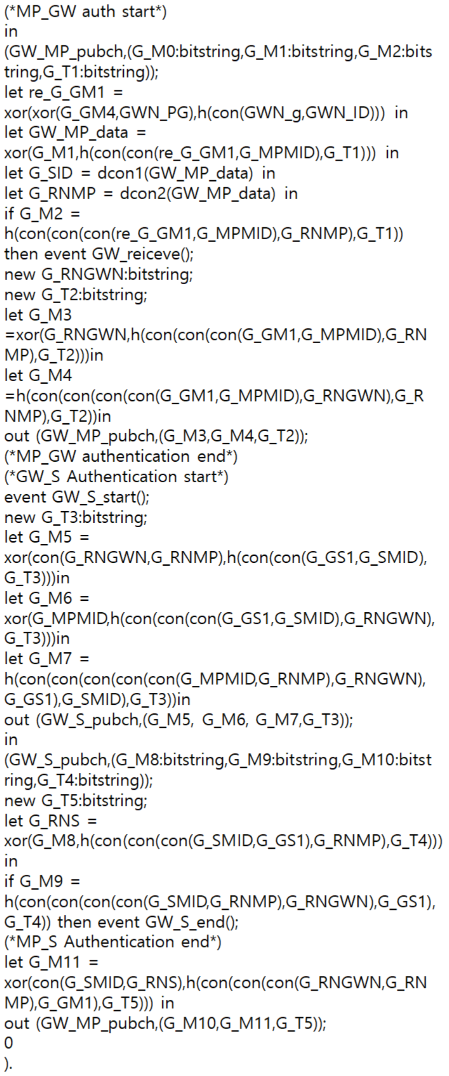
participants, thereby showcasing the robustness of the scheme we proposed.

D. FORMAL SECURITY ANALYSIS USING SCYTHER

We employ an additional protocol verification tool named Scyther to evaluate internal attacks that were not successfully verified using Proverif. Scyther [35] is a widely used verification tool, supporting both the Dolev-Yao model and CK model for attack simulations. This capability will be instrumental in rigorously assessing our protocol's resilience against internal security breaches.

Scyther operates based on the Security Protocol Description Language(SPDL) and offers four authentication claims - *Alive*(aliveness), *Weakagree*(weak agreement), *Niagree*(non-injective agreement), and *Nisynch*(non-injective synchronization). Furthermore, it also provides a *Secret* claim for verifying the confidentiality of data.

Alive indicates that both communication participants can proceed with the communication. The *Weakagree* is established when both communication participants are aware of the communication. *Niagree* is established when there is an agreement on the data exchanged between the participants. If this exchanged message follows a specific order, then *Nisynch* is established [36]. Through these four types of authentication claims, we can perform a check to determine if our proposed protocol can operate properly, and we can examine the confidentiality of key variables using the *Secret* claim. We have collectively analyzed the security of our proposed protocol by validating all four authentication claims provided by Scyther for each communication session, along with verifying the confidentiality of crucial secret variables that might attract the interest of an attacker, as well as the confidentiality of the session key. The simulation result obtained through Scyther v1.1.3 is presented in Fig. 13. According to Fig. 13, we have fulfilled all four authentication claims, and we can confirm that no attacks occurred under the Scyther simulation. This demonstrates that we can satisfy security properties even in various scenarios and establish the session key securely.

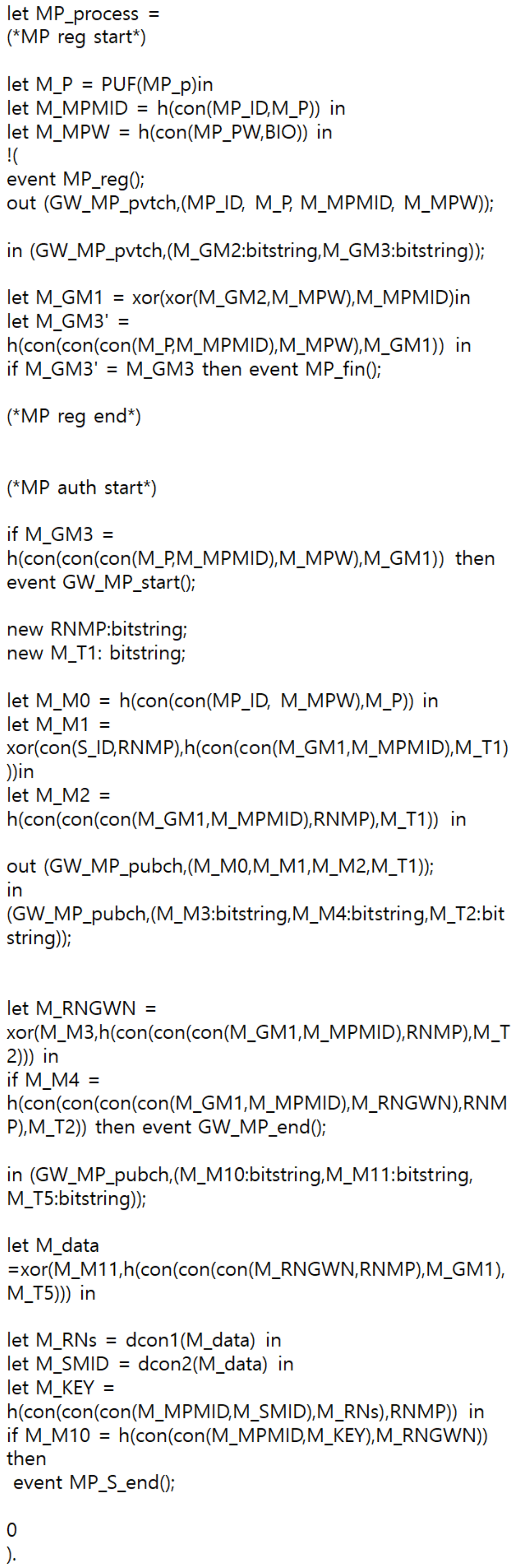


```
(*MP_GW auth start*)
in
(GW_MP_pubch,(G_M0:bitstring,G_M1:bitstring,G_M2:bits
tring,G_T1:bitstring));
let re_G_GM1 =
xor(xor(G_GM4,GWN_PG),h(con(GWN_g,GWN_ID))) in
let GW_MP_data =
xor(G_M1,h(con(con(re_G_GM1,G_MPMID),G_T1))) in
let G_SID = dcon1(GW_MP_data) in
let G_RNMP = dcon2(GW_MP_data) in
if G_M2 =
h(con(con(con(re_G_GM1,G_MPMID),G_RNMP),G_T1))
then event GW_reiceve();
new G_RNGWN:bitstring;
new G_T2:bitstring;
let G_M3
=xor(G_RNGWN,h(con(con(con(G_GM1,G_MPMID),G_RN
MP),G_T2)))in
let G_M4
=h(con(con(con(con(G_GM1,G_MPMID),G_RNGWN),G_R
NMP),G_T2))in
out (GW_MP_pubch,(G_M3,G_M4,G_T2));
(*MP_GW authentication end*)
(*GW_S Authentication start*)
event GW_S_start();
new G_T3:bitstring;
let G_M5 =
xor(con(G_RNGWN,G_RNMP),h(con(con(G_GS1,G_SMID),
G_T3)))in
let G_M6 =
xor(G_MPMID,h(con(con(con(G_GS1,G_SMID),G_RNGWN),
G_T3)))in
let G_M7 =
h(con(con(con(con(con(G_MPMID,G_RNMP),G_RNGWN),
G_GS1),G_SMID),G_T3))in
out (GW_S_pubch,(G_M5, G_M6, G_M7,G_T3));
in
(GW_S_pubch,(G_M8:bitstring,G_M9:bitstring,G_M10:bitst
ring,G_T4:bitstring));
new G_T5:bitstring;
let G_RNS =
xor(G_M8,h(con(con(con(G_SMID,G_GS1),G_RNMP),G_T4)))
in
if G_M9 =
h(con(con(con(con(G_SMID,G_RNMP),G_RNGWN),G_GS1),
G_T4)) then event GW_S_end();
(*MP_S Authentication end*)
let G_M11 =
xor(con(G_SMID,G_RNS),h(con(con(con(G_RNGWN,G_RN
MP),G_GM1),G_T5))) in
out (GW_MP_pubch,(G_M10,G_M11,G_T5));
0
).
```

FIGURE 9. **ProVerif code for GWN_j in authentication phase.**

V. PERFORMANCE ANALYSIS

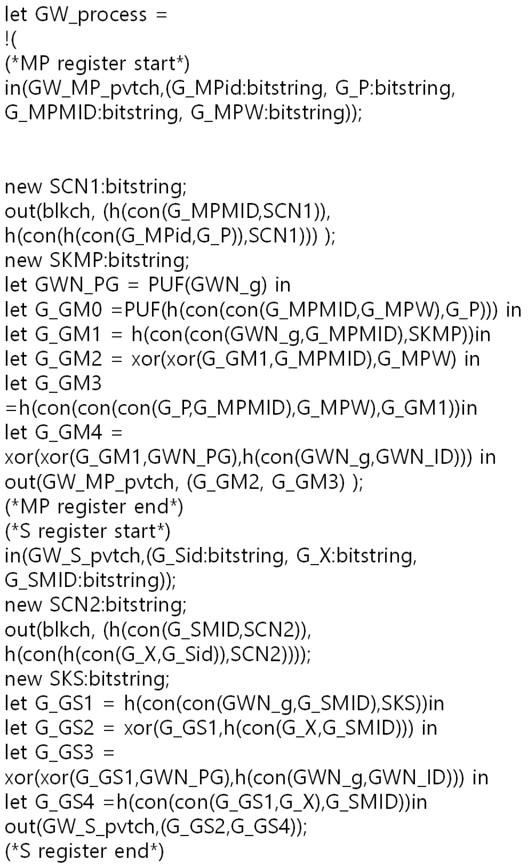
In this section, we compared our proposed scheme with prior research in the field. The comparison is twofold: focusing



```
let MP_process =
(*MP reg start*)

let M_P = PUF(MP_p)in
let M_MPMID = h(con(MP_ID,M_P)) in
let M_MPW = h(con(MP_PW,BIO)) in
!(
event MP_reg();
out (GW_MP_pvtch,(MP_ID, M_P, M_MPMID, M_MPW));

in (GW_MP_pvtch,(M_GM2:bitstring,M_GM3:bitstring));

let M_GM1 = xor(xor(M_GM2,M_MPW),M_MPMID)in
let M_GM3' =
h(con(con(con(M_P,M_MPMID),M_MPW),M_GM1)) in
if M_GM3' = M_GM3 then event MP_fin();

(*MP reg end*)


(*MP auth start*)

if M_GM3 =
h(con(con(con(M_P,M_MPMID),M_MPW),M_GM1)) then
event GW_MP_start();

new RNMP:bitstring;
new M_T1: bitstring;

let M_M0 = h(con(con(MP_ID, M_MPW),M_P)) in
let M_M1 =
xor(con(S_ID,RNMP),h(con(con(M_GM1,M_MPMID),M_T1)
))in
let M_M2 =
h(con(con(con(M_GM1,M_MPMID),RNMP),M_T1)) in

out (GW_MP_pubch,(M_M0,M_M1,M_M2,M_T1));
in
(GW_MP_pubch,(M_M3:bitstring,M_M4:bitstring,M_T2:bit
string));


let M_RNGWN =
xor(M_M3,h(con(con(con(M_GM1,M_MPMID),RNMP),M_T
2))) in
if M_M4 =
h(con(con(con(con(M_GM1,M_MPMID),M_RNGWN),RNM
P),M_T2)) then event GW_MP_end();

in (GW_MP_pubch,(M_M10:bitstring,M_M11:bitstring,
M_T5:bitstring));

let M_data
=xor(M_M11,h(con(con(con(M_RNGWN,RNMP),M_GM1),
M_T5))) in

let M_RNs = dcon1(M_data) in
let M_SMID = dcon2(M_data) in
let M_KEY =
h(con(con(con(M_MPMID,M_SMID),M_RNs),RNMP)) in
if M_M10 = h(con(con(M_MPMID,M_KEY),M_RNGWN))
then
 event MP_S_end();

0
).
```

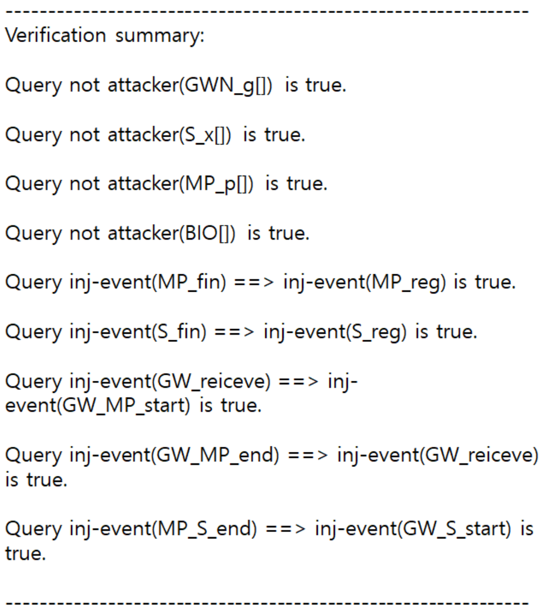
FIGURE 10. ProVerif code for MP_i in registration and authentication phase.

first on the computational cost and subsequently on the communication cost.



```
let GW_process =
!(
(*MP register start*)
in(GW_MP_pvtch,(G_MPid:bitstring, G_P:bitstring,
G_MPMID:bitstring, G_MPW:bitstring));


new SCN1:bitstring;
out(blkch, (h(con(G_MPMID,SCN1)),
h(con(h(con(G_MPid,G_P)),SCN1))) );
new SKMP:bitstring;
let GWN_PG = PUF(GWN_g) in
let G_GM0 =PUF(h(con(con(G_MPMID,G_MPW),G_P))) in
let G_GM1 = h(con(con(GWN_g,G_MPMID),SKMP))in
let G_GM2 = xor(xor(G_GM1,G_MPMID),G_MPW) in
let G_GM3
=h(con(con(con(G_P,G_MPMID),G_MPW),G_GM1))in
let G_GM4 =
xor(xor(G_GM1,GWN_PG),h(con(GWN_g,GWN_ID))) in
out(GW_MP_pvtch, (G_GM2, G_GM3) );
(*MP register end*)
(*S register start*)
in(GW_S_pvtch,(G_Sid:bitstring, G_X:bitstring,
G_SMID:bitstring));
new SCN2:bitstring;
out(blkch, (h(con(G_SMID,SCN2)),
h(con(h(con(G_X,G_Sid)),SCN2))));
new SKS:bitstring;
let G_GS1 = h(con(con(GWN_g,G_SMID),SKS))in
let G_GS2 = xor(G_GS1,h(con(G_X,G_SMID))) in
let G_GS3 =
xor(xor(G_GS1,GWN_PG),h(con(GWN_g,GWN_ID))) in
let G_GS4 =h(con(con(G_GS1,G_X),G_SMID))in
out(GW_S_pvtch,(G_GS2,G_GS4));
(*S register end*)
```

FIGURE 11. ProVerif code for GWN_j in registration phase.

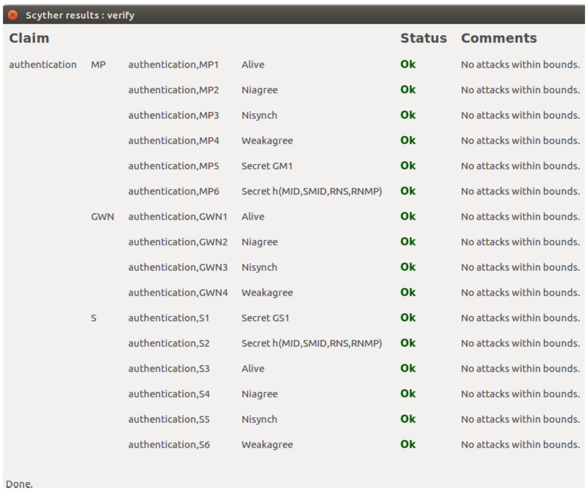


```
--------------------------------------------------------------
Verification summary:

Query not attacker(GWN_g[]) is true.

Query not attacker(S_x[]) is true.

Query not attacker(MP_p[]) is true.

Query not attacker(BIO[]) is true.

Query inj-event(MP_fin) ==> inj-event(MP_reg) is true.

Query inj-event(S_fin) ==> inj-event(S_reg) is true.

Query inj-event(GW_reiceve) ==> inj-
event(GW_MP_start) is true.

Query inj-event(GW_MP_end) ==> inj-event(GW_reiceve)
is true.

Query inj-event(MP_S_end) ==> inj-event(GW_S_start) is
true.

--------------------------------------------------------------
```

FIGURE 12. Results of event code.

A. COMPUTATION COST

We primarily analyze the computational costs of our protocol against previous schemes tailored for the WMSN environment. Given that the execution time of XOR operations



```
Scyther results : verify
Claim                                                                  Status  Comments
authentication  MP   authentication,MP1   Alive                            Ok      No attacks within bounds.
                     authentication,MP2   Niagree                          Ok      No attacks within bounds.
                     authentication,MP3   Nisynch                          Ok      No attacks within bounds.
                     authentication,MP4   Weakagree                        Ok      No attacks within bounds.
                     authentication,MP5   Secret GM1                       Ok      No attacks within bounds.
                     authentication,MP6   Secret h(MID,SMID,RNS,RNMP)      Ok      No attacks within bounds.
                GWN  authentication,GWN1  Alive                            Ok      No attacks within bounds.
                     authentication,GWN2  Niagree                          Ok      No attacks within bounds.
                     authentication,GWN3  Nisynch                          Ok      No attacks within bounds.
                     authentication,GWN4  Weakagree                        Ok      No attacks within bounds.
                S    authentication,S1    Secret GS1                       Ok      No attacks within bounds.
                     authentication,S2    Secret h(MID,SMID,RNS,RNMP)      Ok      No attacks within bounds.
                     authentication,S3    Alive                            Ok      No attacks within bounds.
                     authentication,S4    Niagree                          Ok      No attacks within bounds.
                     authentication,S5    Nisynch                          Ok      No attacks within bounds.
                     authentication,S6    Weakagree                        Ok      No attacks within bounds.
Done.
```

FIGURE 13. Validation result using Scyther.

TABLE 4. Computational cost comparison.

Scheme	MP	GWN	S	Total costs
Wu et al. [7]	$11T_h$	$17T_h$	$6T_h$	$34T_h$
Li et al. [8]	$8T_h$+$3T_{pm}$	$8T_h$+$1T_{pm}$	$4T_h$+$2T_{pm}$	$20T_h$+$6T_{pm}$
Fotouhi et al. [9]	$12T_h$	$22T_h$	$7T_h$	$41T_h$
Wang et al. [13]	$6T_h$	$4T_h$	$3T_h$	$13T_h$
Yu et al. [14]	$9T_h$	$9T_h$	$7T_h$	$25T_h$
Our Scheme	$8T_h$+$3T_H$	$10T_h$+$2T_H$	$8T_h$+$1T_H$	$26T_h$+$6T_H$

is negligible compared to hash functions, our comparison focuses on the communication cost of SHA-1 T_h, SHA-256 T_H, and ECC point multiplication T_{pm}.

Experiments were conducted in a setting featuring a CPU: Intel Core i7-8700 3.20 GHz, memory: 48 GB, and an OS: Win10 64-bit. During these tests, the execution times for the SHA-1, SHA-256 algorithms and point multiplication of ECC, as implemented in Python Cryptography library, were gauged. Our observations determined that SHA-1 and SHA-256 clocked in at 0.0009ms and 0.0010ms, respectively, while point multiplication took 0.628ms. Consequently, the operational duration for our scheme in such an environment is estimated to be 0.0294ms. As delineated in Table 4 and figure 14, our protocol demonstrated a reduction in the overall computational cost by 3.92% and 20.32% in comparison to [9] respectively. It also showed a significant 92.55% decrease when compared to [8]. Conversely, when juxtaposed against [13] and [14], a spike in computational demands was discernible. Excluding [8] that involves T_{pm}, which is known to be much slower than conventional one-way hash functions, our scheme shows an average increase in computational load of 19.29% for MP_i and 58.45% for S_k, while displaying a 5.98% decrease for GWN_j. As a result, when excluding the scheme of Li et al. [8], there was an overall average increase of 15.63% in computational costs. However, in comparison to all other schemes, the computational load for MP_i decreased by an average of 77.80%, that for S_k decreased by 72.64%, and for GWN_j, it decreased by 52.91%. Collating these data points, there was an overall benefit from a 67.37% reduction in computation cost.

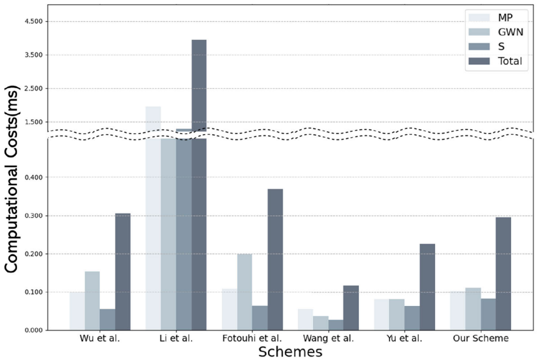


FIGURE 14. Comparison of computational cost.

Despite the evident escalation in computational overhead compared to [13] and [14], as underscored in Table 4, it is crucial to note that our system has utilized these additional computations to enhance its defense against potential threats. Moreover, when compared to recent studies, it has demonstrated superior performance, thereby ensuring its suitability for WMSN environments.

B. COMMUNICATION COST

In this section, the emphasis shifts toward analyzing the communication cost associated with our devised scheme, comparing it against those of previous methods within analogous WMSN contexts. Key components are as follows: the point on ECC is 320 bits, the hash function H is designated as 256 bits, h is 160 bits, the ID of MP is set at 160 bits, the ID of GWN and S is set at 80 bits, random values measure 80 bits, and timestamps are 32 bits. Within our framework, MP_i undertakes a singular transmission, represented as $C1 = \{M_0, M_1, M_2, T_1\}$, which equals 608bits. Concurrently, S_k enacts one transmission denoted by $C_4 = \{M_8, M_9, M_{10}, T_4\}$, amounting to 512 bits. Meanwhile, GWN_j dispatches three separate transmissions: $C_2 = \{M_3, M_4, T_2\}$, $C_3 = \{M_5, M_6, M_7, T_3\}$, and $C_4 = \{M_{10}, M_{11}, T_4\}$ aggregating to 1312 bits — a breakdown of $352 + 512 + 448$ bits.

In Wu et al. [7], MP_i transmits a total of 800 bits, comprising segments of 160, 80, 80, 160, 160, and 160 bits. GWN_j transmits a total of 1360 bits, segmented into 80, 160, 160, and 160 bits in the first part, and 160, 160, 160, 160, and 160 bits in the second part. S_k transmits a total of 320 bits, specifically comprising two segments of 160 bits each. In Li et al. [8], MP_i transmits a total of 800 bits, divided into segments of 320, 160, 160 and 160 bits. GWN_j transmits a total of 1280 bits, segmented into 320, 160, and 160 bits in the first part, and 320, 160 and 160 bits in the second part. S_k transmits a total of 640 bits, comprising segments of 320, 160, and 160 bits. In Fotouhi et al. [9], MP_i transmits a total of 720 bits, divided into segments of 160, 80, 160, 160, and 160 bits. GWN_j transmits a total of 1680 bits, segmented into 80, 160, 160, 160, and 160 bits in the first part, and 160, 160, 160, 160, and 160 bits in the second part. S_k transmits a total of 480 bits, comprising segments of 160, 160, and 160 bits.

TABLE 5. **Communication cost comparison.**

Scheme	MP	GWN	S	Total costs
Wu et al. [7]	800 bits	1360 bits	320 bits	2480 bits
Li et al. [8]	800 bits	1280 bits	640 bits	2720 bits
Fotouhi et al. [9]	720 bits	1680 bits	480 bits	2880 bits
Wang et al. [13]	656 bits	1024 bits	496 bits	2176 bits
Yu et al. [14]	576 bits	1216 bits	576 bits	2368 bits
Our Scheme	608 bits	1312 bits	512 bits	2432 bits

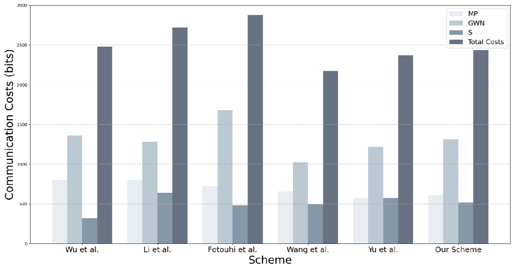


FIGURE 15. **Comparison of communication cost.**

In Wang et al. [13], MP_i transmits a total of 656 bits, broken down into segments of 160, 160, 160, 80, 32, and 64 bits. GWN_j transmits a total of 1024 bits, segmented into 160, 80, and 32 bits in the first part, 160, 160, 80, and 32 bits in the second part, and 80, 160, 160, and 32 bits in the third part. S_k transmits a total of 496 bits, comprising segments of 160, 160, 80, 32, and 64 bits. In Yu et al. [14], MP_i transmits a total of 576 bits, comprising segments of 240, 160, 80, 64, and 32 bits. GWN_j transmits a total of 1216 bits, segmented into 160, 160, and 32 bits in the first part, 240, 160, 80, and 32 bits in the second part, and 160, 160, and 32 bits in the third part.S_k transmits a total of 576 bits, specifically comprising segments of 160, 160, 160, 64, and 32 bits.

As can be seen from Table 5 and Fig. 15, the aggregate communication cost averages out to 2524.8 bits, with our scheme clocking in slightly lower at 2432 bits. This effectively means that our model operates with a leaner communication overhead, shaving off approximately 3.67%. When compared in detail with each of the previous studies, our scheme achieved a 1.93% reduction in communication costs compared to [7], a 10.58% reduction compared to [8], and a 15.55% reduction compared to [9]. On the other hand, there was an increase of 11.76% in communication costs compared to [13], and a 2.70% increase compared to [14]. However, once we factor in the exclusion of the message transfer RN_{MP_i} from the Yu et al. [14] protocol, our scheme's advantage in minimizing communication cost becomes even more pronounced. Furthermore, we have ensured the communication cost for S_k remains at an balanced tier, which is instrumental in dictating the longevity of the sensors in the system. Given these attributes, our protocol as an optimal fit for the WMSN landscapes.

VI. CONCLUSION

The rapid development in IoT and 5G network technologies has influenced the advancement of IoMT and WMSN, allowing in unprecedented capabilities for remote medical support, not experienced in traditional medical systems. However, in these advancements, the vulnerability of patient information became a significant concern, raising both personal and legal challenges, prompting a wave of research aimed at plugging these gaps. Consequently, numerous studies were undertaken to address these concerns.

Yu et al. [14] proposed a protocol that harnessed the powers of both blockchain and PUF. While their approach adeptly addressed a multitude of issues, including the SPOF problem inherent in traditional centralized authentication protocols, it has its limitations.

With critical information channeled to the SC and becoming accessible on the blockchain network, the protocol was susceptible to a gamut of security breaches, such as stolen verifier attacks, physical capture attacks, insider attacks, and MITM attacks. This vulnerability situation prompted us to create a secured authentication protocol tailored for WMSN based on blockchain.

Our findings underscores the efficacy of our proposed scheme showcasing its ability to strike a balance in computational cost compared to other related schemes and reduces the communication cost by 1.81%. As evident from Table 2, our scheme satisfies many security requirements outshining competing schemes such as those delineated in [7], [9], [13], and [14]. In essence, our approach not only provides superior security but also enhances stability. As a result, medical professionals can offer medical support to patients more reliably, and patients, in turn, can receive medical support without the concern of personal information leakage or disruptions in the medical system. While our present methodology exhibits significant advantages, it faces limitations such as the difficulty in modifying smart contracts once they are set and the potential for bottlenecks as the number of nodes increases. We propose future work to focus on improving computational efficiency and updatable functionalities in SC. Specifically, based on the study conducted by [37], which integrates blockchain technology with a focus on continuous healthcare data in real mobile computing environments, we will present more specific plans for integrating a blockchain network into the actual environment. Throughout this process, our efforts will be directed towards eliminating factors that hinder the user experience. These advancements are poised to further establish our scheme's stature as an indispensable protocol within the WMSN environment.

REFERENCES

[1] W. Meng, Y. Cai, L. T. Yang, and W.-Y. Chiu, "Hybrid emotion-aware monitoring system based on brainwaves for Internet of Medical Things," *IEEE Internet Things J.*, vol. 8, no. 21, pp. 16014–16022, Nov. 2021.

[2] H. Rahangdale, N. Chavhan, and P. Ade, "A review on WMSN (wireless medical sensor networks) for health monitoring systems," *ECS Trans.*, vol. 107, no. 1, pp. 1973–1980, Apr. 2022.

[3] J. J. Rodrigues Barata, R. Munoz, R. D. De Carvalho Silva, J. J. P. C. Rodrigues, and V. H. C. De Albuquerque, "Internet of Things based on electronic and mobile health systems for blood glucose continuous monitoring and management," *IEEE Access*, vol. 7, pp. 175116–175125, 2019.

[4] M. K. Kagita, N. Thilakarathne, T. R. Gadekallu, and P. K. R. Maddikunta, "A review on security and privacy of Internet of Medical Things," in *Intelligent Internet of Things for Healthcare and Industry*. Cham, Switzerland: Springer, 2022, pp. 171–187.

[5] X. Li, J. Niu, S. Kumari, J. Liao, W. Liang, and M. K. Khan, "A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity," *Secur. Commun. Netw.*, vol. 9, no. 15, pp. 2643–2655, Oct. 2016.

[6] A. K. Das, A. K. Sutrala, V. Odelu, and A. Goswami, "A secure smartcard-based anonymous user authentication scheme for healthcare applications using wireless medical sensor networks," *Wireless Pers. Commun.*, vol. 94, no. 3, pp. 1899–1933, Jun. 2017.

[7] F. Wu, X. Li, A. K. Sangaiah, L. Xu, S. Kumari, L. Wu, and J. Shen, "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.

[8] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Chen, "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems," *IEEE Syst. J.*, vol. 14, no. 1, pp. 39–50, Mar. 2020.

[9] M. Fotouhi, M. Bayat, A. K. Das, H. A. N. Far, S. M. Pournaghi, and M. A. Doostari, "A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107333.

[10] T. Alladi, V. Chamola, and Naren, "HARCI: A two-way authentication protocol for three entity healthcare IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 2, pp. 361–369, Feb. 2021.

[11] J. Li, Z. Su, D. Guo, K. R. Choo, and Y. Ji, "PSL-MAAKA: Provably secure and lightweight mutual authentication and key agreement protocol for fully public channels in Internet of Medical Things," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13183–13195, Sep. 2021.

[12] L. Ting, M. Khan, A. Sharma, and M. D. Ansari, "A secure framework for IoT-based smart climate agriculture system: Toward blockchain and edge computing," *J. Intell. Syst.*, vol. 31, no. 1, pp. 221–236, Feb. 2022.

[13] W. Wang, Q. Chen, Z. Yin, G. Srivastava, T. R. Gadekallu, F. Alsolami, and C. Su, "Blockchain and PUF-based lightweight authentication protocol for wireless medical sensor networks," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 8883–8891, Jun. 2022.

[14] S. Yu and Y. Park, "A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 20214–20228, Oct. 2022.

[15] M. Khan, S. Hariharasitaraman, S. Joshi, V. Jain, M. Ramanan, A. SampathKumar, and A. A. Elngar, "A deep learning approach for facial emotions recognition using principal component analysis and neural network techniques," *Photogramm. Rec.*, vol. 37, no. 180, pp. 435–452, Dec. 2022.

[16] M. Khan and A. Malviya, "Big data approach for sentiment analysis of Twitter data using Hadoop framework and deep learning," in *Proc. Int. Conf. Emerg. Trends Inf. Technol. Eng.*, Feb. 2020, pp. 1–5.

[17] G. Xu, F. Wang, M. Zhang, and J. Peng, "Efficient and provably secure anonymous user authentication scheme for patient monitoring using wireless medical sensor networks," *IEEE Access*, vol. 8, pp. 47282–47294, 2020.

[18] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2884–2895, Aug. 2018.

[19] 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. Cryptograph. Techn.*, Interlaken, Switzerland, 2004, pp. 523–540.

[20] Y. Gao, S. F. Al-Sarawi, and D. Abbott, "Physical unclonable functions," *Nat. Electron.*, vol. 3, pp. 81–91, Feb. 2020.

[21] K. Wüst and A. Gervais, "Do you need a blockchain?" in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 45–54.

[22] V. Buterin, "Next-generation smart contract and decentralized application platform," Ethereum, Ethereum White Paper, 2014.

[23] A. Sammoud, M. A. Chalouf, O. Hamdi, N. Montavont, and A. Bouallegue, "A secure and lightweight three-factor authentication and key generation scheme for direct communication between healthcare professionals and patient's WMSN," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2020, pp. 1–6.

[24] 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.

[25] J. Ryu, J. Oh, D. Kwon, S. Son, J. Lee, Y. Park, and Y. Park, "Secure ECC-based three-factor mutual authentication protocol for telecare medical information system," *IEEE Access*, vol. 10, pp. 11511–11526, 2022.

[26] J. Ryu, D. Kang, and D. Won, "Improved secure and efficient Chebyshev chaotic map-based user authentication scheme," *IEEE Access*, vol. 10, pp. 15891–15910, 2022.

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

[28] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Annu. Int. Cryptol. Conf.*, 1999, pp. 388–397.

[29] D. G. Padmavathi and M. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *Int. J. Comput. Sci. Inf. Secur.*, vol. 4, no. 1, pp. 117–125, 2009.

[30] S. Itoo, A. A. Khan, M. Ahmad, and M. J. Idrisi, "A secure and privacy-preserving lightweight authentication and key exchange algorithm for smart agriculture monitoring system," *IEEE Access*, vol. 11, pp. 56875–56890, 2023.

[31] Y. Chen and J. Chen, "An efficient and privacy-preserving mutual authentication with key agreement scheme for telecare medicine information system," *Peer-Peer Netw. Appl.*, vol. 15, no. 1, pp. 516–528, Jan. 2022.

[32] M. N. Aman, K. C. Chua, and B. Sikdar, "Mutual authentication in IoT systems using physical unclonable functions," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1327–1340, Oct. 2017.

[33] R. Akkaoui, "Blockchain for the management of Internet of Things devices in the medical industry," *IEEE Trans. Eng. Manag.*, vol. 70, no. 8, pp. 2707–2718, Jan. 2021.

[34] B. Blanchet, B. Smyth, and V. Cheval. (2015). *ProVerif 1.90: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial*. [Online]. Available: http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf

[35] C. J. Cremers, "The Scyther tool: Verification, falsification, and analysis of security protocols: Tool paper," in *Proc. 20th Int. Conf. Comput. Aided Verification (CAV)*, Jul. 2008, pp. 414–418.

[36] N. Anand and M. A. Saifulla, "EN-LAKP: Lightweight authentication and key agreement protocol for emerging networks," *IEEE Access*, vol. 11, pp. 28645–28657, 2023.

[37] P. Chinnasamy, A. Albakri, M. Khan, A. A. Raja, A. Kiran, and J. C. Babu, "Smart contract-enabled secure sharing of health data for a mobile cloud-based e-health system," *Appl. Sci.*, vol. 13, no. 6, p. 3970, Mar. 2023.

TAEWOONG KANG is currently pursuing the bachelor's degree in computer information engineering (major) with Kwangwoon University. His research interests include system security user authentication and machine learning.

NARYUN WOO is currently pursuing the bachelor's degree in computer information engineering (major) with Kwangwoon University. Her research interests include cryptography, cyber security, and digital forensics.

JIHYEON RYU received the B.S. degree in mathematics and computer science and the Ph.D. degree in cyber security from Sungkyunkwan University, South Korea. She is an Assistant Professor with the School of Computer and Information Engineering, Kwangwoon University. Her research interests include cyber security, machine learning, and user authentication.