electronics

Article

ID-Based Deniable Authentication Protocol with Key Agreement and Time-Bound Properties for 6G-Based WBAN Healthcare Environments

Chien-Lung Hsu [1,2,3,4,5,*], Anh-Tuan Nguyen [2] and Guan-Lin Cheng [1]

[1] Department of Information Management, Chang Gung University, Taoyuan City 33302, Taiwan; tiawan5678@gmail.com
[2] Graduate Institute of Management, Chang Gung University, Taoyuan City 33302, Taiwan; d1040010@cgu.edu.tw
[3] Department of Visual Communication Design, Ming-Chi University of Technology, New Taipei City 24301, Taiwan
[4] Administration, Chang Gung Memorial Hospital, Taoyuan City 33305, Taiwan
[5] Healthy Aging Research Center, Chang Gung University, Taoyuan City 33302, Taiwan
* Correspondence: clhsu@mail.cgu.edu.tw

Abstract: The advent of 6G technology is expected to bring a paradigm shift in the field of wireless communication. With its faster data transfer rates and lower latency, 6G could be an ideal solution for the challenges faced by Wireless Body Area Networks (WBANs) in terms of efficient data bandwidth and edge computing. Smart healthcare systems with 6G-based WBANs might provide more efficient and higher-quality healthcare services. However, 6G-based WBAN healthcare systems might face potential security and safety challenges from cybersecurity threats. This paper will propose an ID-based deniable authentication protocol with key agreement and time-bound properties for 6G-based WBAN healthcare environments by considering user privacy, secure communications, authentication, authorization, and scalability of 6G-based WBANs. As compared with previously proposed protocols, the proposed protocol will achieve the following security requirements: mutual authentication, key agreement for secure communication, deniability, time-bound access privilege control, and identity-based public key management for scalable wearable devices and 6G-based WBAN Service Providers. We proved the claimed security requirements of the proposed protocol by using AVISPA simulation and discussed its computational complexities. As compared with previous works, the proposed protocol can gain better contributions in terms of security requirements and performance evaluations for 6G-based WBAN healthcare environments.

Keywords: ID-based; deniability; authentication; mutual key agreement; time-bound; 6G; WBAN; healthcare environment

Citation: Hsu, C.-L.; Nguyen, A.-T.; Cheng, G.-L. ID-Based Deniable Authentication Protocol with Key Agreement and Time-Bound Properties for 6G-Based WBAN Healthcare Environments. *Electronics* **2023**, *12*, 2682. https://doi.org/10.3390/electronics12122682

Academic Editor: Dimitris Kanellopoulos

Received: 31 March 2023
Revised: 9 June 2023
Accepted: 13 June 2023
Published: 15 June 2023

1. Introduction

Wireless Body Area Networks (WBANs) comprise minisensors placed on or within the body to monitor physiological parameters such as heart rate, temperature, and blood pressure. They wirelessly relay data to nearby hubs and devices, facilitating real-time remote health monitoring and significantly improving healthcare delivery efficiency and quality while reducing treatment costs [1,2]. WBANs can utilize a range of devices, including mobile phones, as gateways to gather sensor data and transmit them to a remote server for analysis [3]. The emergence of 6G technology, with its superior data transfer rates and lower latency, could significantly enhance WBANs' data transmission and processing capacities. Additionally, the ability of 6G networks to support more connected devices and provide extensive coverage may enable broader remote healthcare services, reaching rural and underserved populations.

1.1. Research Motivations

In 6G-enabled intelligent healthcare systems, data security and privacy are of utmost concern due to the public internet channels used for communication and the sensitive nature of health data [4]. WBAN users face potential risks, including data breaches and privacy violations. The authentication process acts as the first line of defense, verifying entity identities and granting them access to the IoT system [5]. It ensures data integrity during transmission and protects against unauthorized access or malicious attacks, which are essential for maintaining the security of the entire healthcare system and patient safety.

In contexts sensitive to privacy, users should maintain the ability to deny accessing certain information or performing certain actions, even while authenticated, to prevent leakage of login and access information to third parties [6]. This principle, known as deniable authentication, is vital in smart healthcare systems to protect users from harm or potential retaliation, such as discrimination or social stigma from accessing sensitive medical information [7]. Achieved through a noninteractive authentication methodology involving the El Gamal signature scheme and a shared session secret [8], deniable authentication safeguards against security concerns such as impersonation, forgery, and ensures continued security even if the session secret is compromised [9]. Cellular networks can implement this via message authentication codes (MACs), minimizing the risk of data breaches and privacy violations.

While deniable authentication mitigates certain risks in 6G healthcare settings, it is not a foolproof solution due to potential abuse by malicious actors [10]. Its nature can be exploited for illegal activities, and there is a risk of users misrepresenting identity or falsifying data. To counter these risks, thorough auditing and robust identity verification are vital [11]. In a 6G-IoT healthcare scenario, deniable authentication involving multiple entities such as Cloud Service Providers, Service Providers, Servers, and Users can be complex [12]. Usage of digital certificates ensures secure authentication [12], but managing multiple certificates can challenge bandwidth and system management, leading to slower response times and possible system bottlenecks [13].

Identity-based cryptography (IBC) simplifies secure communication by using unique identifiers such as emails or phone numbers as public keys, removing the need for public key certificates [14]. However, without a secure key agreement protocol, ID-based and deniable authentication can leave communication vulnerable to interception and identity fraud [15,16]. Therefore, implementing such a protocol is critical for maintaining data confidentiality and integrity in 6G-based WBAN healthcare systems.

Smart healthcare services may delegate authentication authority in time-bound slots, as proposed by Tzeng [17] and later improved by Chien [18]. This efficient method allows hierarchical access delegation and can revoke access if necessary, enhancing system security and scalability. Without this, login tickets could be exploited indefinitely, or lost devices might leave open sessions, potentially leading to unauthorized access or data breaches. A time-bound hierarchical key agreement thus restricts key derivation by class relationships and time, addressing these vulnerabilities [19].

Managing WBAN user access is growing complex with their increasing numbers. Temporal role-based access control can ease user management and time-bound access, but unexpected system compromises pose data privacy risks. Thus, there is a significant challenge for a time-bound, hierarchical content management cryptographic solution [20]. In this proposed scheme, login tickets have a set validity period, optimizing bandwidth usage in a multilevel security system. This allows only authorized users to access data and provides controlled access to one-time visit patients with ID-based identities.

Due to the sensitive nature of healthcare data, ID-based properties simplify certificate management, time-bound properties limit access periods, and deniability enhances privacy by ensuring message authenticity. As WBAN users grow, more services are provided by multiple providers, increasing the demand for intelligent systems [21]. Current architectures may not meet service demands, highlighting the need for a more secure and functional authentication protocol.

1.2. Research Contributions

In this paper, we propose an ID-based deniable authentication protocol with the key agreement and time-bound properties for 6G-based WBAN healthcare environments. Specifically, my work allows WBAN users to authenticate each other through an efficient and secure process. By incorporating time-bound properties, our protocol limits the validity of login tickets to a specific period, enhancing security and user control. Furthermore, it utilizes deniability mechanisms, ensuring that conversation traces are not revealed, even in cases of intercepted and decrypted messages. Consequently, this work successfully addresses key challenges in 6G-based WBAN healthcare environments, such as robust data transfer and storage, user authentication based on specific time slots, and preservation of identity privacy. The contributions of this work can be fulfilled by meeting the following security requirements:

- ID-Based public key systems: This simplifies key and certificate management and reduces the complexity to decrease the risks of bandwidth and vulnerabilities but still increases the security level.
- Scalablility for 6G WBAN: It manages this expansion without sacrificing the quality of service, maintaining high security and performance levels even under heavy network loads.
- Key Agreement for securing WBAN communication: It protects all entities from the risk of a third-party intercepting or compromising the key.
- Deniability of authentication for protecting user privacy: It lets the users communicate securely without leaving any trace of their conversation, even if their messages are intercepted and decrypted by an attacker. The verifier cannot convince the third party of the authentication by releasing the communication messages.
- Time-bound authentication service for secure access control: It allows authorized users to access specific resources within a limited time frame, ensuring that only authorized users can access the resources and reducing the risk of unauthorized access.

1.3. Paper Structures

The structure of this paper is organized as follows. We present the related works in Section 2. Section 3 will mention the technical preliminaries of our works. The system model of our work, including all entities with communicating roles as well as all the procedures and algorithms of our proposed protocol, will be presented in Section 4. Section 5 presents a security analysis of the proposed protocol including the AVISPA toolset. Finally, some concluding remarks and future works are given in Section 6, the conclusion of this paper.

2. Related Works

Based on Weil pairing and an elliptic curve cryptosystem, Yi [22] proposed a protocol with a group key agreement scheme to handle faults in the communication network and guarantee that all authorized participants can still communicate with each other; this protocol provides better privacy protection than traditional group key agreement protocols. However, in traditional key agreement protocols, information about the identity or affiliation of the participants might be leaked out to third parties when multiple malicious participants work together to compromise the security of the system. Xu et al. [23] proposed an authenticated asymmetric group key agreement protocol to provide secure and more efficient communication among group members while keeping their affiliations private. By combining symmetric and asymmetric encryption, the protocol achieves authentication and confidentiality and provides better performance and efficiency than traditional group key agreement protocols. Li et al. [24] introduced authenticated dynamic protocols for asymmetric group key agreement. The protocol is designed to be flexible and efficient in handling dynamic group membership, where members can join or leave the group at any time.

Wu et al. [25] and Choi et al. [26] presented a revocable ID-based authenticated group key exchange protocol to address some security concerns related to group key exchange by using the bilinear pairing technique to achieve a secure group key agreement with

resistance to malicious participants by providing better security than traditional group key agreement protocols that do not consider revocation or resistance to malicious participants. The protocol is built on a modified version of the Boneh–Franklin identity-based encryption system, providing better efficiency and scalability than traditional group key agreement protocols that use traditional public key cryptography.

Zhang et el. [27] proposed a round-efficient and sender-unrestricted dynamic group key agreement protocol for secure group communications. This protocol provides better efficiency and security than traditional group key agreement protocols, especially in large-scale dynamic group scenarios. Overall, these articles show improvements in various aspects of group key agreement protocols, such as privacy protection, dynamic group handling, security against malicious participants, efficiency, and scalability. However, none of these articles specifically address the challenges and requirements of using a robust network to store and transfer a great amount of data without corruption. Moreover, these protocols also could perform well in controlling the authentication and authorization of the users by time slots, as well as keeping the privacy of their identities.

Given the drawbacks of all the above works, I am motivated to propose a new protocol to address the security and privacy concerns in 6G-based WBAN healthcare environments.

3. Technical Preliminaries

We discuss some important technical preliminaries including Elliptic Curve Cryptography (ECC), Bilinear pairing, one-way hash function, and computational Diffie–Hellman problem (CDHP), time-bound, and security assumptions which are used in our proposed protocol.

3.1. Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is a method for implementing public key cryptography that utilizes the algebraic structure of the elliptic curve over finite fields. It was first introduced by Neal Koblitz [28] and Victor Miller [29]. Compared to other public key cryptosystems, ECC has a smaller key size, as reported in [30]. For instance, ECC can provide the same level of security strength with a 256-bit key size, while the Rivest–Shamir–Adleman cryptosystem (RSA) requires a 2048-bit key size. This makes ECC more advantageous in terms of performance, transmitted bandwidth, and storage space than RSA. Furthermore, ECC is included in modern cryptosystem applications according to international standards such as IEEE 1363-2000 [31] and ISO/IEC 11770-3 [32].

The equation of ECC can be defined as E: $y^2 = x^3 + ax + b(modp)$ over a finite field F_p, where $a, b \in Z_q$. Therefore, a non-super-singular curve can be represented as $4a^3 + 27b^2 \neq 0$. As a result, there are some properties of ECC from the definition above.

- A specific point O is on an elliptic curve, E; nonetheless, the point O is not on the elliptic curve, E. The point O is an additive identity that is regarded as the point of infinity or a zero point.
- There is a point P with coordinates (x, y) on the elliptic curve, E. The point P is reflected across the x-axis and mapped onto a point, −P (negative P). Hence, the coordinates of the point, −P, are (x, −y).
- While q is the prime order of the point, P, the qP = O, and q is an integer.
- In the finite field F_q, all of the points on the elliptic curve E are called E(F_q).

3.2. Bilinear Pairing [33]

Suppose that a cyclic additive group G_1 is generated by P, where $\forall P \in E(F_q)$. Besides, another cyclic multiplicative group G_2 has the same order q. We assume that c and d are the elements of Z_q. Furthermore, Weil pairing is a category of bilinear mapping which is a map of $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$, with the following characteristics [22,34].

- Bilinear: $\hat{e}(cR, dQ) = \hat{e}(R, Q)^{cd}$, where $\forall R, Q \in G_1$ and $c, d \in Z_q$.
- Nondegenerate: $\hat{e}(R, Q) \neq 1$, while $\forall R, Q \in G_1$.
- Computable: Given $\forall R, Q \in G_1$, there is an efficient algorithm to compute $\hat{e}(R, Q) \neq 1$.

3.3. Brief Review of Time-Bound Method of Chien's Group-Oriented Range-Bound Key Agreement Protocol [18]

In 2018, Chien [18] proposed a group-oriented range-bound key agreement for an Internet of Things scenarios scheme. Chien considered time-bound security requirements for Internet of Things (IoT). The concept of time-bound is that a client can only access the resources from a server in a valid period, which is determined by the server. To explicitly form the concept above into practice, we illustrated it with a mathematical example. A parameter z determines the maximum number of time slots in a specified time zone. The specified time zone could be a day, a month, or even a year. The users send a service request to the server within the authorized time slot, which is determined by two authorized time slots: T1—the beginning, and T2—the ending. The server grants access to the client only if the client accesses the service within the authorized period. If the client accesses the service outside of the authorized time slot, the authentication token will include a negative hash time, which will cause the server to abort the service request. This authentication mechanism is based on one-way hash function properties, and it provides a secure and efficient way to authenticate clients while protecting against unauthorized access.

3.4. Security Assumptions

Security assumptions in our protocol are defined as follows.

3.4.1. Computational Diffie–Hellman Problem (CDHP) [34]

Suppose that G_1 and G_2 are from the cyclic group G of order q with a large prime p. Therefore, a bilinear map $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$, where G_1 is considered to be hard computationally. The CDHP is extremely difficult to solve abP in G_1, which is proven by Joux and Nguyen [25,26], even though a random set $< P, aP, bP >$ is provided.

3.4.2. Decisional Diffie–Hellman Problem (DDHP) [34,35]

The DDHP involves distinguishing between two values, g^{xy} and, g^z, where z is a randomly chosen integer. The DDHP is also believed to be a hard problem, but it is generally considered to be somewhat easier than the CDHP.

3.4.3. One-Way Hash Function (OWHF) [36]

Considering a function, h: x $\rightarrow$ y, where x and y are not mapped, as a hash function, for instance, y = h(x). Whenever the length of the input, x, is the output, y, it can be a fixed-length arbitrary string. However, it is hardly feasible to retrieve from y to x, which is known as h: x $\rightarrow$ y is nearly nonviable.

3.4.4. Time-Bound Method of Chien's Protocol [19]

Chien's protocol [19] uses two secure hash functions to acheive a time-bound security requirement. The system can randomly select two integers as x_s and x_e in GF(p), where p is a large prime. For a time-bound service between time periods t_1 and t_2, the system should determine and issue two secret keys, K_1 and K_2, for the user with the time-bound service access priviledge, where $K_1 = H^{t_{1-1}}(x_s)$, $K_2 = H^{z-t_2}(x_s)$, t_1 is the starting time slot, t_2 is the ending time slot, z is the total time slot, H is a secure one-way hash function, and the symbol $H^t(x_s)$ is denoted as t times hashing for the seed x_s. For securely broadcasting a message M in t time slot, the system can first derive an encryption secret key as $K = H\left(H^{t-1}(x_s) \| H^{24-t}(x_e)\right)$ and broadcast the ciphertext $C = E_K(M)$, where $E_K(M)$ denotes encryption of M over a secret key K. Upon receiving the broadcasting ciphertext C, the user with valid access privilege in t time slot can obtain the message M by performing the following tasks:

Step 1: The user computes

$$H^{t-t_1}(K_1) \tag{1}$$

Step 2: The user computes

$$H^{t_2-t}(K_2) \tag{2}$$

Step 3: The user computes K' as

$$K' = H\left(H^{t-1}(K_1) \| H^{24-t}(K_2)\right) \tag{3}$$

Step 4: The user uses K' to decrypt the ciphertext C to obtain the message M.

If a user cannot derive the valid secret key K' for the t time slot with their two secret keys, K_1 and K_2, they cannot obtain the message M. Hence, the time-bound security requirement can be achieved.

4. Our Proposed Protocol

4.1. Problem Statement

While various protocols have significantly improved group key agreement aspects such as privacy, dynamic group handling, resistance to malicious participants, and scalability, there are still gaps in the system that need to be addressed. For instance, Yi's protocol [22] offers better privacy protection, Xu et al.'s [23] enhances communication efficiency, and Li et al.'s [24] provides flexibility in dynamic group membership changes. Furthermore, Wu et al.'s [25] and Choi et al.'s [26] protocols present effective methods for addressing security concerns related to group key exchange, offering enhanced resistance to malicious activities. Zhang et al.'s [27] protocol optimizes efficiency in large-scale dynamic groups.

4.2. System Model

Figure 1 depicts our proposed system model. In our system model, there are three main roles: the Cloud Service Provider (CSP), the devices, and the Service Provider (SP). The Cloud Service Provider (CSP) serves as a trusted third party, responsible for managing both public and private parameters. The private parameters, held exclusively by the CSP, are central to its secure functioning. Devices, on the other hand, represent the users in a Wireless Body Area Network (WBAN) and are registered to the CSP. They, along with the Service Provider (SP), act as participants that authenticate each other to initiate communication sessions. The SP can represent various entities such as hospitals, emergency institutions, families, tele-medicare companies, etc. Like devices, the SP also needs to register with the CSP and authenticate its identity with registered devices. For an SP to provide healthcare services to users via registered devices, both parties need to authenticate each other and establish a shared secret key for secure communication. To ensure user privacy and restricted authorization, our proposed protocol meets the deniability and time-bound security requirements. Hence, an ID-based deniable authentication protocol with key agreement and time-bound properties for 6G-based WBAN healthcare environments should be achieved by fulfilling the following security features:

- Simplified key and certificate management, reducing complexity and increasing security.
- Scalability for 6G WBAN without compromising quality of service.
- Key agreement for secure WBAN communication, protecting against interception.
- Deniability of authentication, ensuring user privacy and non-repudiation.
- Time-bound authentication service for secure access control.
- Improved efficiency with lower computational overheads compared to existing protocols.

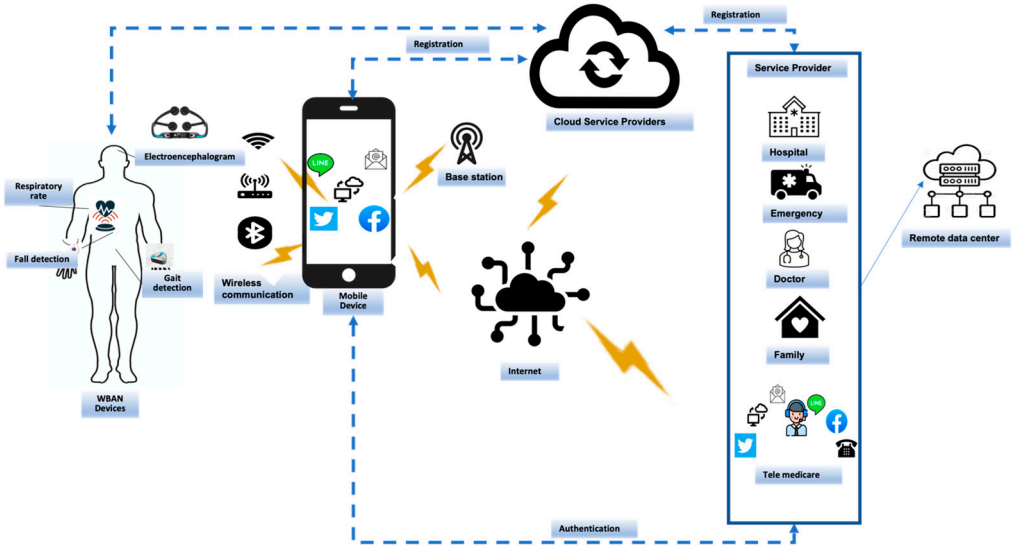

Figure 1. The system model of the proposed scheme.

4.3. Proposed Protocol

The purpose of the proposed scheme is to establish a robust and secure framework for communications, particularly within healthcare settings. The proposed protocol's security features will focus on secure key management, mutual authentication, deniable security, and time-bound access control; this scheme seeks to address the prevalent security concerns. In any secure communication protocol, the security of the private key is crucial; it must be safeguarded from unauthorized access and remain solely in the hands of the rightful owner. Mutual authentication bolsters this security further by ensuring that both parties involved in a conversation verify each other's identities, thereby avoiding interactions with imposters. Key confirmation then plays its role by validating that the shared secret key, crucial for secure communication, is correctly established by all involved parties. The addition of undeniable security into the mix ensures a robust proof of authenticity, preventing a signer from falsely denying their participation in the message exchange. Lastly, the principle of forward security secures past session keys, ensuring that even in the event of a key compromise, the confidentiality of past communications remains intact.

The proposed protocol consists of three phases: system setup, key generation, and identity authentication phases. In the system phase, we will use the Setup algorithm to generate the public and secret parameters used in the whole proposed protocol; the Cloud Service Provider, or *CSP*, is also responsible for carrying out this task. In the key generation phase, we will use the Keygen algorithm to generate public and private keys for Service Providers and devices. Lastly, in the authentication phase, the devices and the *SP* will use the AuthID algorithm to authenticate each other and compose a shared key for further communication. Table 1 describes notations and cryptographic functions used in the protocol. The detailed scheme is presented in the three phases as follows.

In the setup phase, the *CSP* will use the Setup algorithm to select a large prime q and choose an elliptic curve function

$$y^2 = x^3 + ax + bmodq \neq 0 \tag{4}$$

It then generates two cyclic groups G_1 and G_2 from the prime order q and the bilinear pairings $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$. Then, the *CSP* chooses a point P from the G_1 with order q. It will take a random number $s \in Z_q$ as the system private key and calculate

$$P_{pub} = sP \tag{5}$$

as the system public key and choose three one-way hash functions, namely, H_2, H_3, and h, in which $H_2 : \{0,1\} \rightarrow Z_q$ and $H_3 : \{0,1\} \rightarrow G_1$. It is noted that $H_2^t(x)$ denotes x times hashing for the seed x. It is also noted that *date* denotes access date.

Table 1. Notations used in the proposed protocol.

Protocol Symbol	HLPSL Notations	Description
ID_x	IDD, IDSP	The identification of the entity x, where $x = \{i, j\}$.
D, SP, CSP	D, SP, CSP	The abbreviations of Device, Service Provider, and Cloud Service Provider individually.
Q_x	UPPERQD, UPPERQSP	The public key of the entity x, where $x = \{i, j\}$.
S_x	DS, SPS	The private key of the entity x, where $x = \{i, j\}$.
MAC_x	MACI, MACJ	The authentication code for entity x, where $x = \{i, j\}$.
θ	THETA	A random number.
Pub_parameter		The public parameter of the system.
Priv_parameter		The private parameter of the system.
G_1, G_2	G1, G2	The generators are based on the cyclic additive group, which is generated by P.
P	P	A point from the generator with a prime.
q	LOWERQ	A large prime.
s	S	The system secret value.
$H_2()$, $H_3()$	H2, H3	The one-way hash algorithm.
C	CIPHER	The ciphertexts.
k	SKcspd, SKcspsp	The symmetric encryption key
$K_{i,j}$	KIJ	The shared key.
$E_k(m)$	$\{m\}\_k$	A message is encrypted by the symmetric encryption key k.
$D_k(m)$	$\{m\}\_k$	The message is decrypted by the symmetric encryption key k.
$AT_{a,i}$, $AT_{b,i}$	ATA, ATB	Two secret tokens which are generated by *CSP* and published to the Service Provider.

Note: HLPSL is the abbreviation of High-Level Protocol Specification Language for security proof.

CSP can apply a symmetric cryptosystem, for example, RSA to encrypt the message m with key k; on the contrary, $D_k(m)$ is used for decrypting the message with key k. Finally, the *CSP* publishes $Pub\_params = \left(G_1, G_2, P, q, H_2(), H_3(), P_{pub}, E_k(m)\right)$ and preserves $Priv\_params = s$ as a system secret. The specific steps of this phase are described in Algorithm 1.

In the key generation phase, the device, D_i, and the Service Provider, SP_j, are involved in obtaining their public and private key as registration. The private keys of the device and the Service Provider are generated by the *CSP* while they provide their identity to the *CSP* via a secure channel. Due to the input in this phase being the device's identifier D_i and the Service Provider's identifier SP_j the output will be the public and private keys of the device, D_i, and the Service Provider, SP_j. Algorithm 2 will depict the specific steps of this phase.

During the authentication phase, the device D_i and the Service Provider SP_j will authenticate each other through the AuthID algorithm and compose a shared key for further communication. The authentication procedure is provided in Algorithm 3, as follows (Figure 3).

Algorithm 1: Setup

Input : The sec ret parameter $\mathbf{1}^{k}$

Output: The public parameters $Pub\_params = \left(\boldsymbol{G_1}, \boldsymbol{G_2}, \boldsymbol{P}, \boldsymbol{q}, \boldsymbol{H_2}(), \boldsymbol{H_3}(), \boldsymbol{P_{pub}}, \boldsymbol{E_k}(\boldsymbol{m})\right)$ and the private parameters $Priv\_params = s$.

Algorithm:

(1) Select a large prime q, and choose an elliptic curve function $\boldsymbol{y^2 = x^3 + ax + bmodq \neq 0}$, which must meet $\boldsymbol{4a^2 + 27b^3 modq \neq 0}$, where $\boldsymbol{x, y \in Z_q}$ and $\boldsymbol{a, b \in Z_q}$.

(2) Generate two cyclic groups $\boldsymbol{G_1}$ and $\boldsymbol{G_2}$ and the bilinear pairings $\hat{\boldsymbol{e}} : \boldsymbol{G_1 \times G_1 \rightarrow G_2}$.

(3) Choose a point P from the $\boldsymbol{G_1}$.

(4) Take a random number $\mathbf{s} \in \mathbf{Z_q^*}$ as the system private key and calculate $\boldsymbol{P_{pub} = sP}$ as the system public key.

(5) Pick two one-way hash functions, namely, $\boldsymbol{H_2 \wedge H_3}$, where $\boldsymbol{H_2 : \{0,1\} \rightarrow Z_q}$ and $\boldsymbol{H_3 : \{0,1\} \rightarrow G_1}$.

(6) Determine a symmetric algorithm E, such as encryption standards DES and AES, where $\boldsymbol{E_k(m)}$ and $\boldsymbol{D_k(m)}$ are denoted as the encryption and decryption of the message m over the secret key k, respectively.

(7) Publish $Pub\_params = \left(\boldsymbol{G_1}, \boldsymbol{G_2}, \boldsymbol{P}, \boldsymbol{q}, \boldsymbol{H_2}(), \boldsymbol{H_3}(), \boldsymbol{P_{pub}}, \boldsymbol{E_k}(\boldsymbol{m})\right)$, and preserve $Priv\_params = s$.

Algorithm 2: KeyGen

Input: The identity $\boldsymbol{ID_i}$ of $\boldsymbol{D_i}$ and the identity $\boldsymbol{ID_j}$ of Service Provider $\boldsymbol{SP_j}$

Output : The private key $\boldsymbol{S_i}$of the device $\boldsymbol{D_i}$ and the private key $\boldsymbol{S_i}$ of the Service Provider $\boldsymbol{SP_j}$

Algorithm:

(1) The Cloud Service Provider CSP sends a request for $\boldsymbol{ID_x}$'s for device $\boldsymbol{D_i}$ and service provider $\boldsymbol{SP_j}$, where x = i and j. That implies that $\boldsymbol{ID_i}$ stands for device i (i.e., $\mathrm{x} = \mathrm{i}$) and $\boldsymbol{ID_j}$ stands for Service Provider j (i.e., x = j). As in Figure 2, it is noted that $\boldsymbol{ID_x}$ stands for the ID of both Device/Service Provider, where $\boldsymbol{x = \{i, j\}}$.

(2) The device $\boldsymbol{D_i}$ provides the $\boldsymbol{ID_i}$ to the *CSP*.

(3) The Service Provider $\boldsymbol{SP_j}$ submits the identity $\boldsymbol{ID_j}$ to the *CSP*.

(4) Conduct

$$\boldsymbol{Q_i = H_3(ID_i)} \tag{6}$$

$$\boldsymbol{Q_j = H_3(ID_j)} \tag{7}$$

(5) Calculate the device's and Service Provider's private keys as

$$\boldsymbol{S_i = sQ_i} \tag{8}$$

$$\boldsymbol{S_j = sQ_j} \tag{9}$$

(6) Transmit S_i to the device, and send $\boldsymbol{S_j}$ back to the Service Provider, $\boldsymbol{SP_j}$, via the secure channel.

Algorithm 3: AuthID

(1) forward the parameter, (t_1, t_2), $Seed_{a,i}$ and $Seed_{b,i}$ and the identity ID_j of the Service Provider SP_j to the device D_i.

(2) The device D_i generates a random number θ and keeps the value in private with the following steps:

$$a = \hat{e}(P + Q_j, S_i) \tag{10}$$

$$B = \theta Q_j \tag{11}$$

$$v_i = a^{\theta} \tag{12}$$

(3) The device encrypts its identity with a symmetric key v_i and calculates the authentication token, $Seed_{a,i}$ and $Seed_{b,i}$, which have been registered to the *CSP*. The element of the authentication token is composed of the public key (Q_i), the identity (ID_i), the access date (*date*), the subscription starting time (t_1), and the subscription ending time (t_2) of the device, D_i. Besides, the public key Q_j of the Service Provider, $\mathbf{SP}_j$, performs the following equations:

(4) Forward the parameter, (t_1, t_2), $Seed_{a,i}$ and $Seed_{b,i}$, and the identity ID_j of the Service Provider SP_j to the device D_i.

(5) The device D_i generates a random number θ and keeps the value in private with the following steps:

$$a = \hat{e}(P + Q_j, S_i) \tag{13}$$

$$B = \theta Q_j \tag{14}$$

$$v_i = a^{\theta} \tag{15}$$

(6) The device encrypts its identity with a symmetric key v_i and calculates the authentication token, $Seed_{a,i}$ and $Seed_{b,i}$, which have been registered to the *CSP*. The element of the authentication token is composed of the public key (Q_i), the identity (ID_i), the access date (*date*), the subscription starting time (t_1), and the subscription ending time (t_2) of the device, D_i. Besides, the public key Q_j of the Service Provider, $\mathbf{SP}_j$, performs the following equations:

$$C = E_{v_i} \tag{16}$$

$$Q_i = H_3(ID_i) \tag{17}$$

$$Q_j = H_3(ID_j) \tag{18}$$

$$Seed_{a,i} = H_2\left(Q_i \| ID_i \| Q_j \| date \| t_1 \| t_2\right) \tag{19}$$

$$Seed_{b,i} = H_2\left(date \| Q_i \| ID_i \| Q_j \| t_1 \| t_2\right) \tag{20}$$

(7) The device D_i applies the hash function $H_2()$ to calculate a message authentication code MAC_i. Therefore, the message authentication code MAC_i includes a parameter **a**, the symmetric key v_i, the encrypted message **C**, the identity of the device ID_i, and two authentication tokens $Seed_{a,i}$ and $Seed_{b,i}$, where

$$MAC_i = H_2(a \| v_i \| ID_i \| h^{t-1}(Seed_{a,i}) \| h^{z-t}(Seed_{b,i})) \tag{21}$$

(8) The device D_i sends the message (B, MAC_i, **C**) to the Service Provider SP_j.

(9) Due to the Service Provider SP_j obtaining a message (B, MAC_i, $\mathbf{C}$), it forms the device D_i. Hence, the Service Provider SP_j will compute the following equations:

$$v_i' = \hat{e}(P_{pub} + S_j, B) \tag{22}$$

$$ID_i = D_{v_i'}(C) \tag{23}$$

$$a' = \hat{\mathbf{e}}(P_{pub} + S_j, Q_i) \tag{24}$$

(10) The Service Provider SP_j applies the hash function, $H_2()$ to exam the integrity of the message authentication code MAC_i. The Service Provider SP_j computes MAC_i' as:

$$MAC_i' = H_2\left(a' \| v_i' \| t_1 \| C \| ID_i \| h^{t_1 - t}(AT_{a,i}) \| h^{t_2 - t}(AT_{b,i})\right) \tag{25}$$

If MAC_i' equals to MAC_i, it implies that the integrity of the message authentication code MAC_i is valid. Otherwise, the request from the device D_i will be dropped, and the Algorithm will be terminated.

(11) The Service Provider SP_j chooses a random number λ and computes the symmetric key v_j, a shared key $K_{i,j}$, and MAC_j, where

$$v_j = \left(a'\right)^{\lambda} \tag{26}$$

$$K_{i,j} = \left(v_i'\right)^{\lambda} \tag{27}$$

$$MAC_j = H_2(K_{i,j} \| v_j \| v_i' \| MAC_i) \tag{28}$$

(12) The Service Provider SP_j transmits another set of parameters (v_j, MAC_j) to the device D_i, which launches the service request.

(13) The device D_i contributes the shared key with its θ through the following equation:

$$K_{i,j} = v_j^{\theta} \tag{29}$$

(14) As soon as the shared key $K_{i,j}$ has been built by the device D_i, the device D_i will combine $K_{i,j}$, v_j, v_i', MAC_i, and ID_j with the hash function $H_2()$ as the message authentication code MAC_j. The device D_i checks the message authentication code MAC_j by computing MAC_j' as

$$MAC_j' = H_2(K_{i,j} \| v_j \| v_i' \| MAC_i) \tag{30}$$

If MAC_j' equals to MAC_j, it implies that the integrity of the message authentication code MAC_j is valid. Otherwise, the request from the service provider SP_j will be dropped and the algorithm will be terminated.

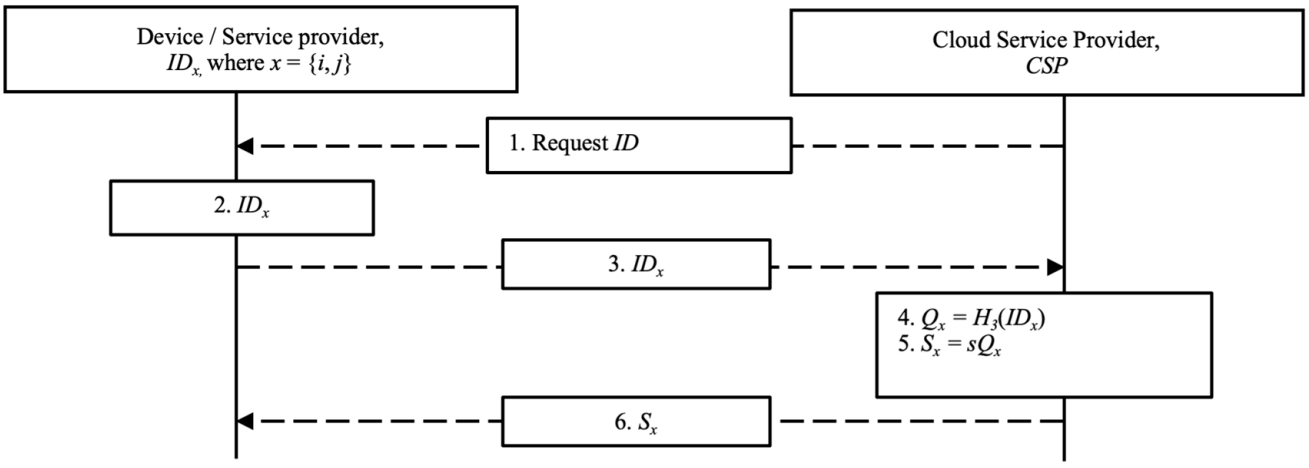

Figure 2. The key generation phase.

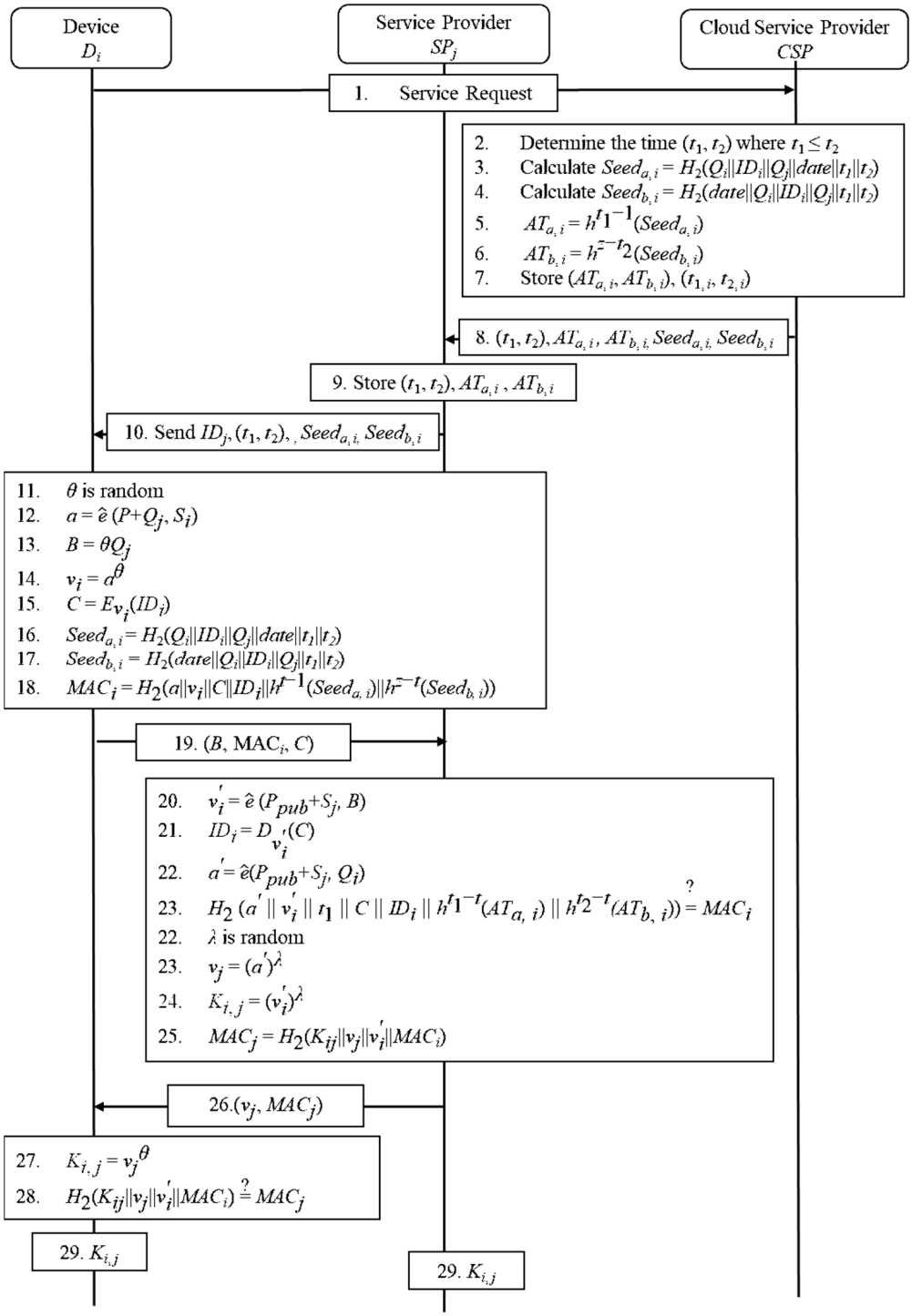

Figure 3. The ID-Authentication protocol.

5. Security Analysis

To evaluate the security of the proposed scheme, various security properties are examined in this section, including security proof of the private key security, mutual authentication, key confirmation, undeniable security, forward security, and an analysis of security using AVISPA.

5.1. Security Proof

5.1.1. Security of Private Keys

The private key of a device D_i cannot be self-generated, as it is provided by the *CSP*. However, a malicious device D_Ω generates its private key using the public parameter $H_3()$ and its identity ID_Ω. Even though the malicious device uses its secret s' to create the private key S_Ω, the system's private key s remains secret. However, the illegal private key S_Ω cannot be used for mutual authentication, and the Service Provider will reject requests from the malicious device due to the computational difficulty of the ECDLP that protects the security of the private key as the following equations:

$$Q_\Omega = H_3(ID_\Omega)$$

$$S_\Omega = s' Q_\Omega$$

5.1.2. Mutual Authentication

To gain access to the Service Provider as a regular device, a malicious device must satisfy the condition $ID_\sigma = D_{v'_\sigma}(C)$ and falsify the message (B, MAC_i, C) by computing $v'_\sigma = \hat{e}\left(P_{pub} + S_j, B\right)$. However, since the message authentication code generated by the malicious device does not contain the system secret s from the KeyGen phase, it will fail to obtain the shared key $K_{\sigma,j}$. Similarly, if a malicious Service Provider tries to communicate with legal devices by impersonating as a legitimate Service Provider, it can falsify the message $(v_\sigma, MAC_\sigma, ID_\sigma)$. However, it cannot pass the validation check for the message authentication code MAC_j, because it does not have the correct value of parameter "a" containing the secret s from the key generation phase. The legal device will therefore reject the session with the malicious Service Provider.

5.1.3. Key Confirmation

Assuming that the malicious device D_Ω is trying to convert the message (B, MAC_i, C, v_i, MAC_j, t_1, t_2) to extract the shared key $K_{i,j}$ among the device D_i and the Service Provider SP_j, the malicious device D_Ω may compose the secret key as below.

If a malicious device D_Ω wants to obtain the shared key $K_{i,j}$ between device D_i and Service Provider SP_j by modifying the message (B, MAC_i, C, v_i, MAC_j, t_1, t_2), it can create a secret key using the following equation:

$$\begin{aligned} K_{i,j} = v_i^\lambda &= \hat{e}(P_{pub} + S_j, B)^\lambda \\ &\hat{e}(P_{pub} + S_j, tQ_i)^\lambda \\ &\hat{e}(P_{pub} + S_j, tQ_i)^{\theta\lambda} \\ &\left(a^\lambda\right)^\theta \\ &v_j^\theta \end{aligned}$$

It is important to note that the malicious device D_Ω cannot obtain θ or λ, as they are generated by the legitimate device and Service Provider, respectively. Therefore, D_Ω cannot generate a falsified shared key $K_{\Omega,j} \vee K_{i,\Omega}$ to participate in a legitimate session and deceive the legitimate device or Service Provider.

5.1.4. Deniability

Once the device D_i and the Service Provider SP_j have each generated their own shared key, they can decrypt messages that have been encrypted using that key. This ensures that both parties can verify the authenticity of the message. However, SP_j cannot determine the identity of the device D_i, as it is not included in the encrypted message.

5.1.5. Forward Security

The shared key $K_{i,j}$ is computed by the Service Provider SP_j and the device D_i, because they select different random numbers from time to time. The possibility of computing the same shared key in two separate conferences is low. The shared key $K_{i,j}$ is generated by both the Service Provider SP_j and the device D_i by selecting different random numbers each time. The likelihood of computing the same shared key in two separate sessions is minimal.

5.2. A Formal Security Verification Using AVISPA

We will utilize AVISPA to analyze our protocol and present the simulation results and HLPSL script details in this section. The protocol involves three roles: the device, the Service Provider, and the cloud Service Provider. However, AVISPA does not support the Weil pairing method, so we substituted it with a hash function, which is a common approach in related research. Additionally, the current AVISPA version does not include the elliptic curve key initialization phase, so we defined the public key, private key, and session key beforehand as PUBLICP, inv(PUBLICP), and SK, respectively.

The environment role defines the protocol's design goals, including secrecy of DS, SPS, and KIJ, and authentication of the Service Provider and device using MACI and MACJ. The protocol involves five roles: CSP as the Cloud Service Provider, SP as the Service Provider, D as the device, the session overseeing all communication channels, and the environment testing for attacks within an insecure channel, covering variables, procedures, roles, session, and multiple security goals. Encrypted messages using symmetric keys, such as SKcspd, between CSP and D are considered secure, and symmetric keys, such as SKcspd, are inaccessible to attackers. Finally, a brief profile of the HLSPL script for all roles will be provided.

Figure 4 presents the HLPSL specification of a Cloud Service Provider, which is played by the CSP. Initially, the CSP has knowledge of all the agents (D, SP, CSP), symmetric keys for secure communication (SKcspd, SKcspsp, SKspd), a predefined hash function (H, H2, H3, MUL), a public key generated by a predefined elliptic curve algorithm (PUBLICP), and a send/receive channel under the Dolev–Yao model (noted with dy) (SND, RCV). The keyword "played_by CSP" indicates that this is the role of the Cloud Service Provider. After the keyword "def=", the scripting of the CSP's steps and procedures starts in detail. The keyword "local" indicates that all local variables should be declared in this section, including a variable "State" declared as a natural number of data types.

The following sets of variables are separated by commas (,) with specified data types, including a set of variables with a message data type declared as "text". The keyword "init State:= 1" initializes the local variables, and the "State" starts at step 1. The "transition" keyword marks the beginning of how the protocol executes and behaves. In State 1, the key generation phase between the CSP and D is initiated within a secure channel using the symmetric key SKcspd. After the CSP receives the identity of D with a symmetric key, the state switches to 3, and the CSP calculates the public key of D, UPPERQ. Then, the CSP applies its secret to register the device and sends the public key UPPERQ and private key DS back to D with a symmetric key.

The same procedure is conducted between the Service Provider and the CSP, with variables for the identity of the Service Provider, IDSP, the symmetric key, SKcspsp, the public key, UPPERSP, and private key, SPS. After the key generation phase, the CSP will receive a service request from the device for a time-bound delegation in State 5 and will perform the time-bound delegation for the device when the state approaches 7. In step 7, the CSP will assign T1, T2, and DATE within the authorized period, generate two authentication seeds, SEEDA and SEEDB, from the public key of the device, the identity of the device, and the public key of the Service Provider, and produce two authentication tokens, ATA and ATB, respectively. Finally, the CSP sends T1, T2, ATA, ATB, SEEDA, and SEEDB to the device. The notation "end role" indicates the completion of the CSP's role. Some variables are marked with a prime (') when the CSP assigns a new value to the variables. The ": =" stands for being defined as, and the following string "new()"

is instantiating a new value to the primed (′) variable. The symbol " /\ " denotes the conjunction action.

```
% Role of Cloud Service Provider, CSP
role cloud_service_provider (D, SP, CSP: agent, SKcspd, SKcspsp, SKspd:symmetric_key, % in se-
cure channel
H, H2, H3, MUL: hash_func, PUBLICP: public_key, SND, RCV: channel(dy))
played_by CSP
def=
local State: nat,
SERVICEREQUEST, UPPERQD, UPPERQSP, TEMP_N, NEWKEY,
T1, T2, ATA, ATB, DATE, SEEDA, SEEDB, G1, G2, UPPERP, P, LOWERQ: message,
IDD, IDSP, S, DS, SPS: text
init State := 1
transition
% Device Registration
% This phase is for registering the device
1. State = 1 /\ RCV ({IDD'}_SKcspd) =|>
State' := 3
/\ UPPERQD' := H3(IDD')
/\ TEMP_N' := new()
/\ DS' := MUL(S, UPPERQD')
/\ SND({DS'.TEMP_N'.UPPERQD'}_SKcspd)
/\ NEWKEY' := H(P, TEMP_N')
/\ secret(DS', ds, {CSP, D})
/\ secret(NEWKEY', newkey_d, {CSP, D})
2. State = 3 /\ RCV({DS}_NEWKEY) =|>
State' := 5
/\ request(CSP, D, csp_d_ds, DS)
% Service Provider Registration
% This phase is for registering the serivce provider
3. State = 1 /\ RCV({IDSP'}_SKcspsp) =|>
State' := 3
/\ UPPERQSP' := H3(IDSP')
/\ TEMP_N' := new()
/\ SPS' := MUL(S, UPPERQSP')
/\ SND({SPS'.TEMP_N'.UPPERQSP'}_SKcspsp)
/\ NEWKEY' := H(P, TEMP_N')
/\ secret(SPS', sps, {CSP, SP})
/\ secret(NEWKEY', newkey_sp, {CSP, SP})
4. State = 3 /\ RCV({SPS}_NEWKEY) =|>
State' := 5
/\ request(CSP, D, csp_sp_sps, SPS)
% Time-bound register from device
5. State = 5 /\ RCV(SERVICEREQUEST') =|>
State' := 7
/\ T1' := new()
/\ T2' := new()
/\ DATE' := new()
/\ SEEDA' := H2(UPPERQD.IDD.UPPERQSP.DATE'.T1'.T2')
/\ SEEDB' := H2(DATE'.UPPERQD.IDD.UPPERQSP.T1'.T2')
/\ ATA' := H(SEEDA')
/\ ATB' := H(SEEDB')
/\ SND(T1'.T2'.ATA'.ATB'.SEEDA'.SEEDB')
end role
```

Figure 4. The HLPSL script for the role of Cloud Service Provider, which is played by CSP.

Following the description of CSP, we will describe the role of device D and provide the HLPSL script for it in Figure 5. The device's knowledge includes agents (D, SP, CSP), symmetric keys (SKcspd, SKcspsp, SKspd), hash functions (H, H2, H3, MUL), a public key (PUBLICP), and communication channels (SND and RCV). The device's structure is similar to CSP, with the same local variables and keywords. However, the "init State" keyword starts from 0 to indicate the script's starting point. This expression also appears in the Service Provider's role and is permitted to execute on AVISPA.

```
% Role of Device
role device (D, SP, CSP: agent, SKcspd, SKcspsp, SKspd:symmetric_key, H, H2, H3, MUL:
hash_func, PUBLICP: public_key, SND, RCV: channel(dy))
played_by D
def=
local State: nat,
SERVICEREQUEST, GROUPKEY, UPPERQD, UPPERQSP, T1, T2, VJ, SEEDA, SEEDB,
THETA, TEMP_N, NEWKEY, ALPHA, B, VI, DATE, G1, G2, UPPERP, P, LOWERQ: message,
IDD, IDSP, S, MACJ, DS, CIPHER, MACI, KIJ: text, PREP, BILINEARPAIR: hash_func
init State := 0
transition
% Device Registration
% This phase is for registering the device
1. State = 0 /\ RCV(start) =|>
State' := 2
/\ IDD' := new() /\ SND({IDD'}_SKcspd)
% Recive the DS from CSP
2. State = 2 /\ RCV({DS'.TEMP_N'}_SKcspd) =|>
State' := 4
/\ NEWKEY' := H(P, TEMP_N')
/\ SND({DS'}_NEWKEY')
/\ witness(D, CSP, csp_d_ds, DS')
/\ SND(SERVICEREQUEST)
% Device delegation
% This phase is for delegating the token
3. State = 4 /\ RCV(T1'.T2'.SEEEDA'.SEEDB'.IDSP') =|>
State' := 6
/\ DATE' := new()
/\ THETA' := new()
/\ ALPHA' := BILINEARPAIR(PREP(P, UPPERQSP), DS)
/\ B' := MUL(THETA', UPPERQSP)
/\ VI' := exp(ALPHA', THETA')
/\ CIPHER' := {IDD}_VI'
/\ UPPERQD' := H3(IDD)
/\ UPPERQSP' := H3(IDSP')
/\ SEEDA' := H2(UPPERQD'.IDD.UPPERQSP'.DATE'.T1'.T2')
/\ SEEDB' := H2(DATE'.UPPERQD'.IDD.UPPERQSP'.T1'.T2')
/\ MACI' := H2(ALPHA'.VI'.CIPHER'.IDD.H(SEEDA').H(SEEDB'))
/\ SND(B'.MACI'.CIPHER')
% Generating the shared key
% This phase is for delegating the token
4. State = 6 /\ RCV(VJ'.MACI') =|>
State' := 8
/\ KIJ' := exp(VJ', THETA)
/\ MACJ' := H2(KIJ'.VJ'.VI.MACI)
/\ request(D, SP, device_serviceprovider_maci, MACI)
/\ secret(KIJ', sp_d_key, {SP, D})
/\ SND(MACJ')
/\ witness(D, SP, device_serviceprovider_macj, MACJ')
end role
```

Figure 5. The HLPSL script for the role of the device which is played by *D*.

After declaring the local variables and essential keywords, we present the script's workflow. The device sends the "RCV(start)" keyword to initialize the script, and then sends its identity (IDD) to CSP via the secure channel using SKcspsp. The device receives a private key (DS) from CSP through the secure channel protected by SKcspsp. The device then sends a service request (SERVICEREQUEST) to CSP and receives authorized seeds, subscription starting and ending times, and the Service Provider's identity (IDSP) in State 4.

In State 6, the device performs authentication using the received parameters within a valid time slot. The device and Service Provider generate their public keys from their identities. After authentication, the device issues a number (B) multiplied with ALPHA from bilinear pairing, a message authentication code (MACI), and a cipher containing the device's identity (CIPHER) to the Service Provider.

The first authentication process is successfully completed at the Service Provider, resulting in the message containing a parameter (VJ) and another message authentication (MACJ), which is transmitted to the device. In State 7, the shared key (KIJ) and another round of authentication are constructed on the device. Two authentication procedures to validate for AVISPA are specified in State 8. The first is a request to determine whether the message authentication code (MACI) is valid, and the second is an event when the device calculates the message authentication code (MACJ) and wants to check it with the Service Provider.

The "exp" keyword denotes a mathematical exponent computation, and the "PREP" keyword is defined as a hash function for addition computation, since AVISPA's built-in function does not support it. The "BILINEARPAIR" keyword refers to the Weil pairing, which is also a hash function. Finally, the "end role" keyword is necessary to complete the device's role actions.

To summarize the role of the Service Provider, we present a brief description and provide the HLPSL script for it in Figure 6. The initial part of the script remains the same as that for the device, with symmetric keys, hash functions, public key, and communication channels unchanged. The Service Provider is denoted as SP in the script, and local variables and keywords are similar to those used in the device script. However, the Service Provider script differs in the identity of the Service Provider, IDSP, the symmetric key for secure communication between the Service Provider and the CSP, SKcspsp, the public key, UPPERQSP, and the private key, SPS.

The key generation phase is similar to that in the device script, but with these new values. After generating the keys, the Service Provider receives a message which is from state 0 to 4, (T1'.T2'.ATA'.ATB'.SEEDA'.SEEDB') from the CSP and keeps ATA and ATB, forwarding T1, T2, and IDSP to the device. Next, the Service Provider receives a message from the device and performs an authentication phase to authenticate and grant the device the subscribed services. Keywords such as "PREP," "BILINEARPAIR," and "exp" are used in the script, as in the device script. However, a new shared key, KIJ, is generated by the Service Provider and kept secret between the Service Provider and the device. The script ensures the secrecy of KIJ using the keyword "secret" and identifier "sp_d_key" for AVISPA auditing. The Service Provider also verifies the validity of the shared key using the message authentication code, MACJ. Finally, the script includes the string "request (SP, D, device_serviceprovider_macj, MACJ)" and the keyword "end role" to conclude the activities.

Figure 7 illustrates the role of the session, which involves the device, Service Provider, and *CSP*. The starting parts of the script are the same as in the device role, including the symmetric keys (*SKcspd*, *SKcspsp*, *SKspd*), (*H*, *H*2, *H*3, *MUL*), the public key (*PUBLICP*), and communication channels (*SND*, *RCV*). However, after the "def=" keyword, the communication channels for each agent are declared as local variables, and each agent is assigned to send and receive channels. To compose the session and communicate across different channels, the keyword "*composition*" is used to instruct AVISPA on how the session is constructed. The session is built up with the *CSP*, device, and Service Provider, with the same header information as in the agent roles, including symmetric keys, hash functions,

public key, and communication channels. Once the communication channels (*SD*, *RD*, *SSP*, *RSP*, *SCSP*, *RCSP*) are assigned to their respective agents, the session is complete. Finally, the script concludes with the "*end role*" keyword to ensure proper operation.

```
% Role of Service Provider
role service_provider (D, SP, CSP: agent, SKcpsd, SKcspsp, SKspd:symmetric_key, H, H2, H3,
MUL: hash_func, PUBLICP: public_key, SND, RCV: channel(dy))
played_by SP
def=
local State: nat,
SERVICEREQUEST, GROUPKEY, UPPERQD, UPPERQSP, T1, T2, ATA, ATB, B, SEEDA, SEEDB,
THETA, NEWKEY, TEMP_N, ALPHA, VI, VJ, DATE, LAMBDA, G1, G2, UPPERP, P, Q: message,
IDD, IDSP, S, CIPHER, MACI, SPS, KIJ, MACJ: text, PREP, BILINEARPAIR: hash_func
init State := 0
transition
% Service Provider Registration
% This phase is for registering the Service Provider
1. State = 0 /\ RCV(start) =|>
State' := 2
/\ IDSP' := new()
/\ SND({IDSP'}_SKcspsp)
% Recive the SPS from CSP
2. State = 2 /\ RCV({SPS'.TEMP_N'}_SKcspsp) =|>
State' := 4
/\ NEWKEY' := H(P, TEMP_N')
/\ SND({SPS'}_NEWKEY')
/\ witness(SP, CSP, csp_sp_sps, SPS')
% Received service request
3. State = 4 /\ RCV(T1'.T2'.ATA'.ATB'.SEEDA'.SEEDB') =|>
State' := 6
/\ SND(T1.T2.SEEDA.SEEDB.IDSP)
% Service Provider delegation and generate the shared key
% This phase is for calculating and validating the value of MACI
% and generating the shared key.
4. State = 6 /\ RCV(B'.MACI'.CIPHER') =|>
State' := 8
/\ LAMBDA' := new()
/\ VI' := BILINEARPAIR(PREP(PUBLICP, SPS), B)
/\ IDD' := {CIPHER'}_VI'
/\ UPPERQD' := H3(IDD')
/\ ALPHA' := BILINEARPAIR(PREP(PUBLICP, SPS), UPPERQD')
/\ MACI' := H2(ALPHA'.VI'.CIPHER'.IDD'.ATA.ATB)
/\ VJ' := exp(ALPHA', LAMBDA')
/\ KIJ' := exp(VI', LAMBDA')
/\ MACJ' := H2(KIJ'.VJ'.VI'.MACI'.MACJ')
/\ SND(VJ'.MACI')
/\ secret(KIJ', sp_d_key, {SP, D})
5. State = 8 /\ RCV(MACJ) =|>
State' := 9
/\ request(SP, D, device_serviceprovider_macj, MACJ)
end role
```

Figure 6. The HLPSL script for the role of the Service Provider, which is played by *SP*.

```
% Session
role session ( D, SP, CSP: agent, SKcspd, SKcspsp, SKspd:symmetric_key, PUBLICP: pub-
lic_key, H, H2, H3, MUL: hash_func)
def=
local   SD, RD,   % Channel between D to CSP
SSP, RSP, % Channel between SP and CSP
SCSP, RCSP: channel (dy) % Channel between SP and D
composition
cloud_service_provider(D, SP, CSP, SKcspd, SKcspsp, SKspd, H, H2, H3, MUL, PUBLICP, SD,
RD)
/\ device(D, SP, CSP, SKcspsp, SKcspsp, SKspd, H, H2, H3, MUL, PUBLICP, SSP, RSP)
/\ service_provider(D, SP, CSP, SKcspd, SKcspsp, SKspd, H, H2, H3, MUL, PUBLICP, SCSP,
RCSP)
end role
```

Figure 7. The HLPSL script for the role of the session.

In Figure 8, different from the previous script of the roles in the parentheses, in other words (), the header of the environment is empty parentheses. All variables are declared and instantiated using the "*const*" keyword. The agents, *D*, *SP*, and *CSP*, are instantiated as *d, sp, and csp*, respectively. Symmetric keys, *skcspd, skcspsp, and skspd*, are also instantiated, representing *SKcspd, SKcspsp*, and *SKspd*. However, a new symmetric key, *ski*, is introduced as an instance that an intruder can use. The hash function is instantiated as *h, h2, h3, and mul*, representing *H, H2, H3, and MUL*, respectively. Two public keys based on the elliptic curve cryptosystem are instantiated as publicp, representing *PUBLICKP*, and *publicpi*, which is built for the intruder. The "*protocol_id*" keyword declares the identifiers mentioned earlier for the goal section. The "*intruder_knowledge*" keyword is used to define the knowledge that the intruder possesses, including all agents, public keys (including the intruder's public key), and hash functions. The "*inv()*" keyword is used to invert a key, allowing AVISPA to convert a given public key to a private key. Each session is composed of a "session" keyword and all the instances. These sessions simulate possible attacks on the protocol under the symmetric key, *ski*. The goal section is critical for AVISPA to validate the safety of the protocol *SAFE* or *UNSAFE*. The "*secrecy_of*" keyword is used to indicate that the identifiers and related instances are intended to keep secrets, such as *ds, sps, and sp_d_key*. The "*authentication_on*" keyword is used when agents request authentication, such as *device_serviceprovider_maci, device_serviceprovider_macj, csp_d_ds, csp_sp_sps.*

```
% environment
role environment()
def=
const   d, sp, csp: agent, skcspd, skcspsp, skspd, ski: symmetric_key, h, h2, h3, mul:
hash_func,
publicp, publicpi: public_key, ds, sps, cipher, sp_d_key, csp_d_ds, csp_sp_sps, newkey_d,
newkey_sp, device_serviceprovider_maci, device_serviceprovider_macj: protocol_id
intruder_knowledge = {d, sp, csp, publicp, publicpi, inv(publicpi), h, h2, h3,mul}
composition
session(d, sp, csp, skcspd, skcspsp, skcspd, publicp, h, h2, h3, mul)
/\ session(d, i, csp, ski, ski, ski, publicp, h, h2, h3, mul)
/\ session(i, sp, csp, ski, ski, ski, publicp, h, h2, h3, mul)
/\ session(d, sp, i, ski, ski, ski, publicp, h, h2, h3, mul)
end role
% set goal
goal
secrecy_of ds, sps, sp_d_key, newkey_d, newkey_sp
authentication_on device_serviceprovider_maci, device_serviceprovider_macj, csp_d_ds,
csp_sp_sps
end goal
environment()
```

Figure 8. The HLPSL script for the role of environment.

We examine and test our suggested protocol using the OFMC and CL-AtSe back-end checkers. The OFMC report, presented in Figure 9, confirms that our protocol is secure and meets all the security goals we designed. We also use the CL-AtSe back-end checker to validate our protocol, and Figure 10 shows the results, indicating that our protocol is secure and satisfies the security goals. To provide a reference, we include execution snapshots in Figures 11 and 12.

```
% OFMC
% Version of 2006/02/13
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/span/span/testsuite/results/
PrivacyPerservedUserAuthenticationwithTimeBoundforIOTenvrionment.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 9.17s
visitedNodes: 2944 nodes
depth: 9 plies
```

Figure 9. The results of the OFMC summary report.

```
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
TYPED_MODEL
PROTOCOL
/home/span/span/testsuite/results/
PrivacyPerservedUserAuthenticationwithTimeBoundforIOTenvrionment.if
GOAL
As Specified
BACKEND
CL-AtSe
STATISTICS
Analysed : 374 states
Reachable: 74 states
Translation: 0.14 s
Computation: 0.00 s
```

Figure 10. The result of the CL-AtSe summary.

Several recent studies have highlighted the potential benefits of 6G in healthcare, including improved communication speed and capacity [37] and enhanced quality of life for patients through telecare services [38]. In 2023, Suraci et al. [39] pointed out several security risks of deploying 6G in a healthcare environment such as impersonate risk or data breached risk. To eliminate these risks, we can use mutual authentication. Hence, providing an ID-based mutual authentication feature is considered crucial in 6G healthcare environments, because it will ensure that both users and devices are verified and authenticated each other to protect patients' safety, sensitive information, and prevent unauthorized access. Hence, only legitimate and authorized entities can interact with

the healthcare system. After performing authentication protocol, only the system can accomplish other cryptography to achieve the confidentiality, integrity, and availability for healthcare service and information.

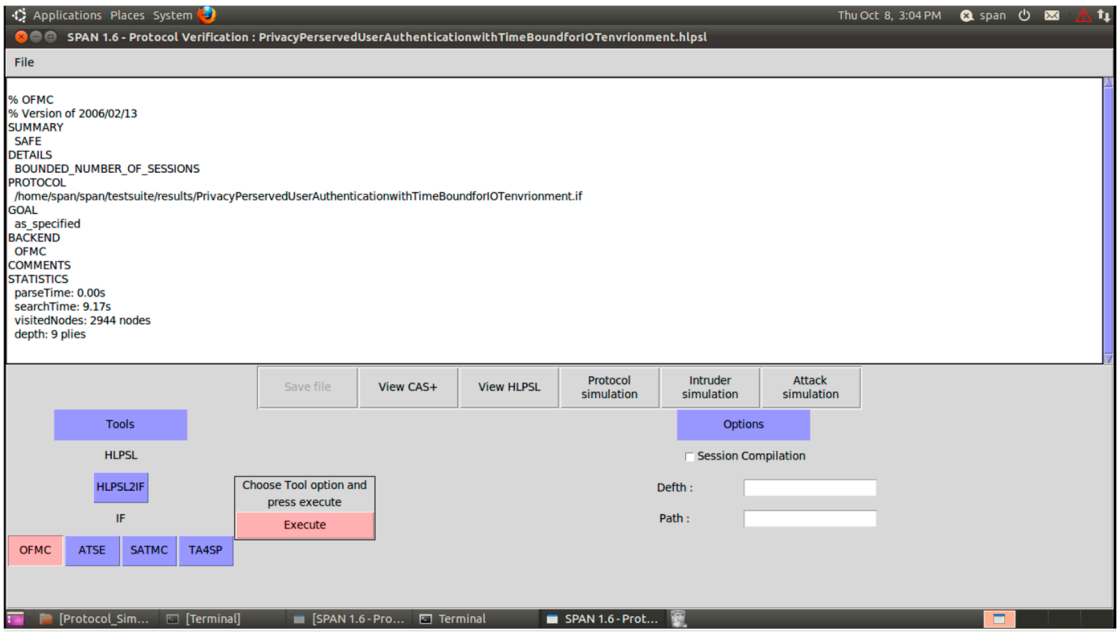

Figure 11. The snapshot of the formal security analysis of the proposed protocol using OFMC back-end.

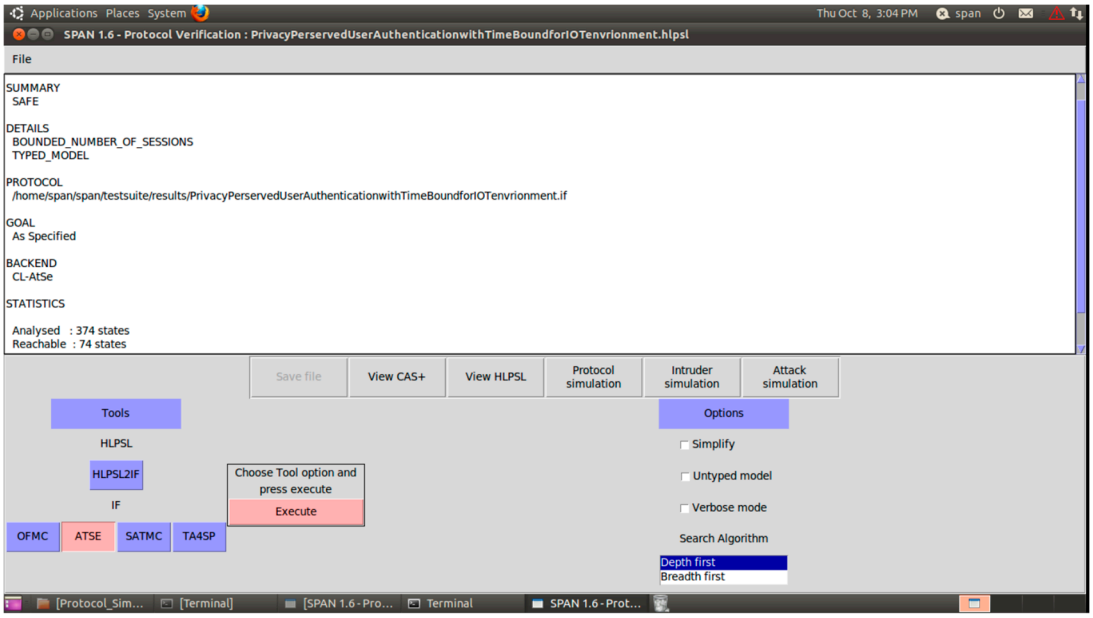

Figure 12. The snapshot of the formal security analysis of the proposed protocol using CL-AtSe back-end.

Suraci et al. [40] proposed a more secured and lightweight 6G eHealth system. The study used a device-to-device mutual authentication protocol to mitigates security issues. It can ensure that both communicating entities verify each other's identity before sending important data. However, their study cannot achieve better performance in mutual authentication and time-bound 6G-based healthcare environments. Later, Le et al. [3] also proposed a three-factor authentication protocol for multiple service providers in 6G-aided

intelligent healthcare systems. In their study, they can provide fast authentication and time-bound security features to overcome the above drawbacks to strengthen the security requirements and accelerate the communication processes.

In a 6G-based mutual authentication healthcare system, there might be a risk of authenticated communication messages disclosure to the third party and violation of the privacy and confidentiality of patients. Hence, the deniability feature is important in 6G healthcare environments to provide user privacy. However, both schemes [3,40] do not provide deniability protocols to provide entities/users with privacy. Considering scalability of the system, we must use the ID-based Deniable Authentication Protocol with Key Agreement and Time-Bound Properties for 6G-based WBAN healthcare environments to gain better performance.

While some studies have identified potential risks and proposed partial solutions, there is still a need to develop comprehensive and robust security measures for 6G deployment in healthcare environments. Considering the importance of security and privacy concerns in our research, we tabulate the comparison results of various functions achieved by different protocols in Table 2. The symbol ✓denotes that the protocol achieves a specific function. We also use the symbol ✗ to denote that the function is not achieved by the protocol. It is observed that the proposed protocol provides more security properties and functionalities as compared with the previous protocols in terms of ID-based, deniability mutual authentication, key agreement, and time-bound properties for 6G-based WBAN healthcare environments. In particular, only our work introduces deniability ID-based key agreement authentication and time-bound authentication solutions in the proposed 6G-IoT WBAN healthcare environment.

Table 2. Comparison of functionality.

Functions	**[22]**	**[23]**	**[24]**	**[25]**	**[26]**	**[27]**	**[40]**	**[3]**	**Ours**
Provide 6G-based intelligent healthcare environment	✗	✗	✗	✗	✗	✗	✓	✓	✓
Provide key agreement authentication	✓	✓	✗	✗	✓	✓	✓	✓	✓
Provide deniability authentication	✗	✓	✓	✗	✗	✗	✗	✓	✓
Provide ID-based mutual authentication	✗	✓	✗	✓	✓	✗	✗	✗	✓
Provide time-bound authentication	✗	✗	✗	✗	✗	✗	✗	✓	✓

5.3. Performance Analysis

We assessed the interpretation of our protocol with other study protocols [22–27] and adopted the estimation time, which is based on the result of the application with the PBC library on a common hardware platform with Intel Core i5-4460 CPU at 3.2 GHz. In order to unite the benchmark baseline, we assume that n is the number of group members. A cyclic additive group and a multiplicative group with the order q, G_1, and G_2, respectively, $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$ implies the permitted bilinear map. Besides, the generally adopted 80-bit security (equal to RSA-1024 bit or ECC-160 bit) level is regarded. Therefore, we organized the operation time consumption in Table 3 for the comparison in Table 4 with an illustration in Figure 13. The y-axis of Figure 13 is the estimated time for calculation, which is applied with logarithm adjustments with the base of two and the x-axis is the number of groups.

Table 3. The running time of the computing operations.

Notation	**Descriptions**	**Time Overheads**
TBP	The bilinear pairing operation	13.909 ms
TSM	The scalar multiplication operation in G_1	6.869 ms
TEXP	The exponentiation operation in G_2	0.140 ms

Note: ms means million seconds.

Table 4. Computational overheads comparison.

Protocol	Number of T_{SM}	Number of T_{EXP}	Number of T_{BP}
Choi et al. [26]	$8n$	-	$4n$
Zhang et al. [27]	$n^2 + 5n$	-	$4n$
Xu et al. [23]	$2n^2 - 2n$	-	$n^2 + 2n$
Yi et al. [22]	$7n$	$2n$	$2n^2$
Wu et al. [25]	$2n^2 + 9n$	-	$8n$
Li et al. [24]	$2n^2$	$2n^2 + 2n$	$3n$
Our protocol	$2n$	$5n$	$4n$

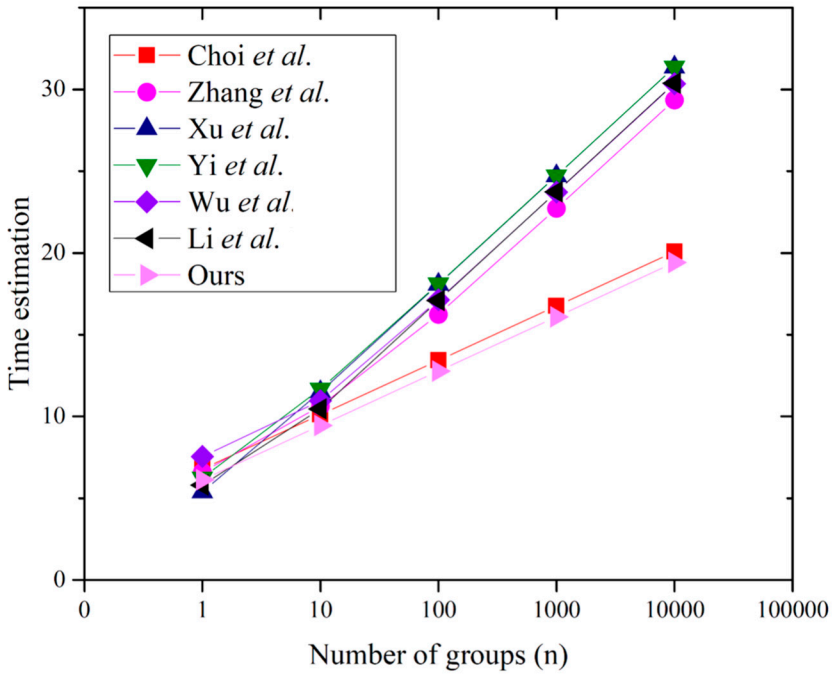

Figure 13. The results of our analysis [22–27].

Scalar multiplication refers to multiplying a scalar value (typically an integer) with a point on an elliptic curve. The running time of scalar multiplication in elliptic curve cryptography depends on the chosen algorithm, such as the double-and-add algorithm or the Montgomery ladder algorithm. These algorithms provide efficient ways to perform scalar multiplication on elliptic curves. The running time of scalar multiplication is typically proportional to the number of bits in the scalar value, which determines the number of iterations required [41]. Exponentiation, on the other hand, involves raising a base value to a large exponent (typically an integer) and computing the result. The running time of exponentiation depends on the algorithm used, such as the square-and-multiply algorithm or the binary exponentiation algorithm. These algorithms provide efficient ways to compute exponentiation, taking advantage of properties such as squaring and modular reductions. The running time of exponentiation is typically proportional to the number of bits in the exponent, which determines the number of multiplications required [42]. In terms of running time, scalar multiplication on elliptic curves tends to be more computationally expensive than exponentiation in traditional modular arithmetic. This is because scalar multiplication involves multiple iterations of point additions on the elliptic curve, while exponentiation typically involves a series of multiplications. Additionally, the specific algorithms used for scalar multiplication and exponentiation can also impact their respective running times.

Le et al.'s scheme [3] is designed for E-healthcare services between patients and services, not for devices. Moreover, [40] is designed based on symmetric cryptography, not public key infrastructure. Hence, our computational comparison does not include the two 6G-based schemes. The results demonstrate that our protocol is the most efficient by contracting to others; however, the efficiency of Choi et al. [26] is almost the same as our protocol.

6. Conclusions

Healthcare systems that utilize the 6G network architecture offer quick and effortless communication channels between WBAN users and healthcare Service Providers, thereby ensuring speedy analysis of medical reports for multiple patients. Nevertheless, security and privacy remain significant issues in these systems. In this paper, we have proposed an ID-based deniable authentication protocol with key agreement and integrated time-bound properties. It allows WBAN users, Service Providers, and cloud Service Providers to establish secured healthcare communications efficiently. The main contributions of our proposed protocol are given below:

- Our proposed protocol is based on ID-based public key systems. It simplifies key and certificate management and reduces the complexity to decrease the risks of bandwidth and vulnerabilities, but still increases the security level.
- Our proposed protocol can achieve scalablility for 6G WBAN. It manages this expansion without sacrificing the quality of service, maintaining high security and performance levels even under heavy network loads.
- Our proposed protocol can achieve key agreement for securing WBAN communication. It protects all entities from the risk of a third party intercepting or compromising the key.
- Our proposed protocol can achieve deniability of authentication for protecting user privacy. It lets the users communicate securely without leaving any trace of their conversation, even if their messages are intercepted and decrypted by an attacker. The verifier cannot convince the third party of the authentication by releasing the communication messages.
- Our proposed protocol can achieve time-bound authentication service for secure access control. It allows authorized users to access specific resources within a limited time frame, ensuring that only authorized users can access the resources and reducing the risk of unauthorized access.
- Our proposed protocol can gain better efficiency than previously proposed protocols in terms of computational overheads.
- Our proposed protocol can gain better security than previously proposed protocols by applying the AVISPA tool to give a formal security verification.

In future works, we will consider further improving the efficiency of the work with conference key distribution to reduce the cost of computing and storing. Other rigorous methods of authentication including three-factor authentication would also be an interesting research direction to consider.

Author Contributions: Conceptualization, C.-L.H.; Methodology, C.-L.H.; Software, G.-L.C.; Validation, C.-L.H. and A.-T.N.; Formal analysis, C.-L.H., A.-T.N. and G.-L.C.; Writing—original draft, C.-L.H. and A.-T.N.; Visualization, A.-T.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Science and Technology Council in Taiwan under Grant NSTC -111-2410-H-182-007-MY2, Grant NSTC-111-2221-E-155-038, Grant NSTC-111-2218-E-218-004-MBK and Grant NSTC-112-2811-H-182-002. It was also supported in part by the Healthy Aging Research Center.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bhatti, D.S.; Saleem, S.; Imran, A.; Iqbal, Z.; Alzahrani, A.; Kim, H.; Kim, K.-I. A Survey on Wireless Wearable Body Area Networks: A Perspective of Technology and Economy. *Sensors* **2022**, *22*, 7722. [CrossRef]
2. Hasan, K.; Biswas, K.; Ahmed, K.; Nafi, N.S.; Islam, M.S. A comprehensive review of wireless body area network. *J. Netw. Comput. Appl.* **2019**, *143*, 178–198. [CrossRef]
3. Le, T.-V.; Lu, C.-F.; Hsu, C.-L.; Do, T.K.; Chou, Y.-F.; Wei, W.-C. A novel three-factor authentication protocol for multiple service providers in 6G-aided intelligent healthcare systems. *IEEE Access* **2022**, *10*, 28975–28990. [CrossRef]
4. Alabdulatif, A.; Khalil, I.; Saidur Rahman, M. Security of Blockchain and AI-Empowered Smart Healthcare: Application-Based Analysis. *Appl. Sci.* **2022**, *12*, 11039. [CrossRef]
5. Alsaeed, N.; Nadeem, F. Authentication in the Internet of Medical Things: Taxonomy, Review, and Open Issues. *Appl. Sci.* **2022**, *12*, 7487. [CrossRef]
6. Hsu, C.-L.; Chuang, Y.-H.; Hung, M.-T. An Efficient Deniable Authentication Protocol from Pairings to Protect Users' Privacy. *Chiang Mai J. Sci.* **2014**, *41*, 1384–1391.
7. Di Raimondo, M.; Gennaro, R. New Approaches for Deniable Authentication. In Proceedings of the 12th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 7–11 November 2005; pp. 112–121.
8. Lee, W.-B.; Wu, C.-C.; Tsaur, W.-J. A novel deniable authentication protocol using generalized ElGamal signature scheme. *Inf. Sci.* **2007**, *177*, 1376–1381. [CrossRef]
9. Shao, Z. Efficient deniable authentication protocol based on generalized ElGamal signature scheme. *Comput. Stand. Interfaces* **2004**, *26*, 449–454. [CrossRef]
10. Rasmussen, K.; Gasti, P. Weak and Strong Deniable Authenticated Encryption: On Their Relationship and Applications. In Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust (PST), Belfast, Ireland, 28–30 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1–10.
11. Zarour, M.; Alenezi, M.; Ansari, M.T.J.; Pandey, A.K.; Ahmad, M.; Agrawal, A.; Kumar, R.; Khan, R.A. Ensuring data integrity of healthcare information in the era of digital health. *Healthc. Technol. Lett.* **2021**, *8*, 66–77. [CrossRef]
12. Mavridis, I.; Georgiadis, C.; Pangalos, G. Access-rule certificates for secure distributed healthcare applications over the Internet. *Health Inform. J.* **2002**, *8*, 127–137. [CrossRef]
13. Alezabi, K.A.; Hashim, F.; Hashim, S.J.; Ali, B.M. An Efficient Authentication and Key Agreement Protocol for 4G (LTE) Networks. In Proceedings of the 2014 IEEE Region 10 Symposium, Kuala Lumpur, Malaysia, 14–16 April 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 502–507.
14. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. *SIAM J. Comput.* **2003**, *32*, 586–615. [CrossRef]
15. Li, F.; Xiong, P.; Jin, C. Identity-based deniable authentication for ad hoc networks. *Computing* **2014**, *96*, 843–853. [CrossRef]
16. Zhu, H.; Zhang, Y.; Zhang, Y.; Li, H. A Novel and Provable Authenticated Key Agreement Protocol with Privacy Protection Based on Chaotic Maps towards Mobile Network. *Int. J. Netw. Secur.* **2016**, *18*, 116–123.
17. Tzeng, W.-G. A time-bound cryptographic key assignment scheme for access control in a hierarchy. *IEEE Trans. Knowl. Data Eng.* **2002**, *14*, 182–188. [CrossRef]
18. Chien, H.-Y. Group-oriented range-bound key agreement for Internet of Things scenarios. *IEEE Internet Things J.* **2018**, *5*, 1890–1903. [CrossRef]
19. Odelu, V. A Dynamic Time-Bound Access Control for Secure Hierarchical Content Sharing. *TechRxiv* **2023**. [CrossRef]
20. Hsu, C.-L.; Le, T.-V.; Lu, C.-F.; Lin, T.-W.; Chuang, T.-H. A privacy-preserved E2E authenticated key exchange protocol for multi-server architecture in edge computing networks. *IEEE Access* **2020**, *8*, 40791–40808. [CrossRef]
21. Kobayashi, S.; Kane, T.B.; Paton, C. The privacy and security implications of open data in healthcare. *Yearb. Med. Inform.* **2018**, *27*, 041–047. [CrossRef]
22. Yi, X. Identity-based fault-tolerant conference key agreement. *IEEE Trans. Dependable Secur. Comput.* **2004**, *1*, 170–178.
23. Xu, C.; Li, Z.; Mu, Y.; Guo, H.; Guo, T. Affiliation-hiding authenticated asymmetric group key agreement. *Comput. J.* **2012**, *55*, 1180–1191. [CrossRef]
24. Li, M.; Xu, X.; Guo, C.; Tan, X. AD-ASGKA–authenticated dynamic protocols for asymmetric group key agreement. *Secur. Commun. Netw.* **2016**, *9*, 1340–1352. [CrossRef]
25. Wu, T.-Y.; Tseng, Y.-M.; Tsai, T.-T. A revocable ID-based authenticated group key exchange protocol with resistant to malicious participants. *Comput. Netw.* **2012**, *56*, 2994–3006. [CrossRef]
26. Choi, K.Y.; Hwang, J.Y.; Lee, D.H. Efficient ID-based group key agreement with bilinear maps. In Proceedings of the Public Key Cryptography–PKC 2004: 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, 1–4 March 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 130–144.
27. Zhang, L.; Wu, Q.; Domingo-Ferrer, J.; Qin, B.; Dong, Z. Round-efficient and sender-unrestricted dynamic group key agreement protocol for secure group communications. *IEEE Trans. Inf. Secur.* **2015**, *10*, 2352–2364. [CrossRef]
28. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
29. Miller, V.S. *Use of Elliptic Curves in Cryptography*; Springer: Berlin/Heidelberg, Germany, 1986.
30. Barker, E.; Barker, W. *Recommendation for Key Management, Part 2: Best Practices for Key Management Organization*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2018.

31. Jablon, D. IEEE P1363 Standard Specifications for Public-Key Cryptography. In Proceedings of the IEEE NIST Key Management Workshop CTO Phoenix Technologies Treasurer, Gaithersburg, MD, USA, 1–2 November 2001; pp. 1–26.
32. Cremers, C.; Horvat, M. Improving the ISO/IEC 11770 Standard for Key Management Techniques. In Proceedings of the Security Standardisation Research: First International Conference, SSR 2014, London, UK, 16–17 December 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 215–235.
33. Miller, V.S. The Weil pairing, and its efficient calculation. *J. Cryptol.* **2004**, *17*, 235–261. [CrossRef]
34. Joux, A.; Nguyen, K. Separating decision Diffie–Hellman from computational Diffie–Hellman in cryptographic groups. *J. Cryptol.* **2003**, *16*, 239–247. [CrossRef]
35. Boneh, D. The Decision Diffie-Hellman Problem. In Proceedings of the Algorithmic Number Theory: Third International Symposiun, ANTS-III, Portland, OR, USA, 21–25 June 1998; Springer: Berlin/Heidelberg, Germany, 2006; pp. 48–63.
36. Winternitz, R.S. Producing a One-Way Hash Function from DES. In *Advances in Cryptology: Proceedings of Crypto 83*; Springer: Berlin/Heidelberg, Germany, 1984; pp. 203–207.
37. Nayak, S.; Patgiri, R. 6G communication technology: A vision on intelligent healthcare. *Health Inform. A Comput. Perspect. Healthc.* **2021**, *932*, 1–18.
38. Mucchi, L.; Jayousi, S.; Caputo, S.; Paoletti, E.; Zoppi, P.; Geli, S.; Dioniso, P. How 6G Technology Can Change the Future Wireless Healthcare. In Proceedings of the 2020 2nd 6G Wireless Summit (6G SUMMIT), Levi, Finland, 17–20 March 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
39. Suraci, C.; Pizzi, S.; Molinaro, A.; Araniti, G. Business-Oriented Security Analysis of 6G for eHealth: An Impact Assessment Approach. *Sensors* **2023**, *23*, 4226. [CrossRef]
40. Suraci, C.; Pizzi, S.; Molinaro, A.; Araniti, G. MEC and D2D as Enabling Technologies for a Secure and Lightweight 6G eHealth System. *IEEE Internet Things J.* **2021**, *9*, 11524–11532. [CrossRef]
41. Liu, H.; Zhou, Y.; Zhu, N. A novel elliptic curve scalar multiplication algorithm against power analysis. *Math. Probl. Eng.* **2013**, *2013*, 862508. [CrossRef]
42. Robert, J.-M.; Negre, C.; Plantard, T. Efficient Fixed-base exponentiation and scalar multiplication based on a multiplicative splitting exponent recoding. *J. Cryptogr. Eng.* **2019**, *9*, 115–136. [CrossRef]