mathematics

Article

Cross-Server End-to-End Patient Key Agreement Protocol for DNA-Based U-Healthcare in the Internet of Living Things

Tuan-Vinh Le [1,2]

[1] Program of Artificial Intelligence and Information Security, Fu Jen Catholic University, New Taipei 24206, Taiwan; 155315@mail.fju.edu.tw

[2] Program of Medical Informatics and Innovative Applications, Fu Jen Catholic University, New Taipei 24206, Taiwan

Abstract: (1) Background: Third-generation sequencing (TGS) technique directly sequences single deoxyribonucleic acid (DNA) molecules, enabling real-time sequencing and reducing sequencing time from a few days to a few hours. Sequencing devices can be miniaturized and DNA-reading sensors placed on the body to monitor human health and vital signs, building an "internet of living things" (IoLT) facilitating ubiquitous healthcare services. In many cases, patients may wish to directly connect to each other for purposes of sharing real-time sequencing data, medical status or trading genomic data, etc. (2) Problems: User registration for a specific service may be limited due to some reason. Registering for multiple redundant services would also result in wasted money and possible wasteful communication overhead. In addition, since medical data and health information are very sensitive, security and privacy issues in the network are of paramount importance. (3) Methods: In this article, I propose a cross-server end-to-end (CS-E2E) patient authenticated key agreement protocol for DNA-based healthcare services in IoLT networks. My work allows two patients to mutually authenticate each other through assistance of respective servers, so that they can establish a reliable shared session key for securing E2E communications. The design employs multiple cost-saving solutions and robust cryptographic primitives, including smart-card-based single sign-on, elliptic curve cryptography, biohash function, etc. (4) Results: My proposed protocol is proven to be secure against various attacks and to incur reasonable communication cost compared to its predecessor works. The protocol also provides the support for more security properties and better functionalities. (5) Conclusions: The E2E communications between the patients are properly protected using the proposed approach. This assures a secure and efficient cross-server patient conversation for multiple purposes of healthcare communication.

Keywords: third-generation sequencing (TGS); DNA-reading sensor; onsite DNA sequencing; ubiquitous healthcare (U-healthcare); internet of living things (IoLT); smart-card-based single sign-on (SC-SSO); end-to-end (E2E) communication; key agreement; three-factor authentication

MSC: 68M25

check for updates

Citation: Le, T.-V. Cross-Server End-to-End Patient Key Agreement Protocol for DNA-Based U-Healthcare in the Internet of Living Things. *Mathematics* **2023**, *11*, 1638. https://doi.org/10.3390/math11071638

Academic Editors: Ding Wang, Qi Jiang and Chunhua Su

Received: 18 February 2023
Revised: 18 March 2023
Accepted: 23 March 2023
Published: 28 March 2023

Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Second-generation sequencing (SGS), also known as next-generation sequencing (NGS), is the process of identifying the sequence of millions of short deoxyribonucleic acid (DNA) fragments in parallel [1]. When sequenced data are shared with researchers, the causes of many diseases will be identified and new drugs or precision medicines developed [2]. However, the need for longer reads and shorter sequencing times, which are the drawbacks of SGS, led to the advent of third-generation sequencing (TGS) [3]. The TGS technique directly sequences single DNA molecules, enabling real-time sequencing and reducing sequencing time from a few days to a few hours. Sequencing devices can be miniaturized and DNA-reading sensors placed on the body to monitor human health and vital signs, building an “internet of living things” (IoLT) [3,4]. Taking nanopore sequencing technologies as an example, SmidgION is a very small nanopore sequencer designed to be run on smartphones or mobile devices using their batteries and dedicated apps [3]. The DNA samples are loaded into the tiny sequencer from the sensors. The data produced include FAST5 (HDF5) files and/or FASTQ files [5]; they are either stored on the phone’s memory or uploaded to the cloud. In this way, medical clinics can screen for new viruses in seconds, and researchers can obtain DNA sequences in real time for specific analysis. Ubiquitous healthcare (U-healthcare), which is a combination of electronic and mobile healthcare, is more concerned with person-centric therapy rather than traditional hospital healthcare [3]. To this end, DNA-based sequenced data are completely useful for U-healthcare, since it facilitates patient-centric service and personalized treatment process, for instance, real-time monitoring of body fluids [3].

1.1. Research Problems

In a DNA-based U-healthcare, patients communicate with service providers in order to receive medical information and analysis results on their health status through the internet. In many cases, patients may want to directly connect to each other for the purpose of sharing real-time sequencing data, medical status or trading genomic data [2,4,6], etc. However, user registration for a specific service may be limited due to some reason. Moreover, registering for multiple redundant services would result in wasted money and possible wasteful communication overhead. Therefore, a cross-server end-to-end (CS-E2E) communication solution is required in such U-healthcare scenarios for the purposes of efficiency and convenience.

In the direct communication between patients conducted with the assistance of servers, the generated shared key must only be known to the patients; this is the basic security requirement in all E2E communications. In addition, since the communication is conducted through an insecure internet channel, and personal care data and health information are very sensitive, security concerns are of paramount importance. Adversaries may launch various attacks (e.g., replay attacks), aiming to compromise patient privacy or obstruct the service system. The legitimacy of healthcare providers (e.g., doctors, physicians, etc.) during communications also needs to be considered to avoid possible fraudulent behaviors. The two-factor authentication mechanism enabled through a combination of a password and a smart card was introduced in many existing articles to alleviate the security risks present in a single-factor mechanism [7,8]. However, once the adversaries compromise the password or the smart card successfully, the system would be vulnerable to some unavoidable attacks, e.g., impersonation attacks. Upon the demand, there would be a massive number of U-healthcare services provided by different institutions or hospitals. It is not possible for the traditional single-server system to satisfy the needs of users where they may enjoy an increasing number of medical services [7]. Moreover, remembering too many credentials in order to use multiple services may cause a certain inconvenience and directly affect communication efficiency. It is necessary to provide a better authentication mechanism, which can effectively address all the above issues. In addition, concerns regarding the computation cost and communication cost must also be considered in the design.

1.2. Contributions

In this article, I propose a cross-server E2E patient authenticated key agreement protocol for DNA-based U-healthcare services in IoLT networks. Specifically, my work allows two patients to mutually authenticate each other through the assistance of respective servers, so that they can securely establish a reliable shared session key for E2E communications. The efficiency of the communications in the proposed protocol is also considered in the design. The contributions of this work can be summarized as follows.

(1) I introduce a DNA-based U-healthcare application constructed in CS-E2E communication environments. In the proposed model, multiple servers provide U-healthcare services based on real-time DNA sequencing data produced by smart tiny sequencers

with TGS technology in the IoLT network. Patients are allowed to share healthcare data with each other directly.

(2) The protocol allows the patients to store single registered credentials on a smart card and enter the credentials once per session only. They are allowed to choose specific servers of a multi-server system from a list in the device to enjoy multiple registered services. I call this solution "smart-card-based single sign-on (SC-SSO)". Furthermore, the proposed SC-SSO is designed without a centerless solution to alleviate communication cost and reduce the security risk of third-party authority compromise.

(3) The authentication protocol is designed using three factors, combining password, smart card and biometrics. It can guarantee higher security for communications compared to the single-factor or two-factor solutions. In the protocol, a perfect forward secrecy of shared E2E session keys is assured. Patient anonymity and untraceability are provided in the protocol. Patients can also update their passwords and biometrics to ensure higher security.

(4) The security proof of my proposed protocol is presented using formal verification tools, including the real-or-random (RoR) model and Burrows–Abadi–Needham (BAN) logic. In addition, an informal analysis is provided to further discuss the resistance to various security attacks, e.g., replay attacks, impersonation attacks, etc.

1.3. Paper Structure

The remainder of the paper is organized as follows. In Section 2, the related works and research motivation are presented. In Section 3, some important technical preliminaries employed in the proposed protocol are explained. The problem statement in Section 4 presents the system model of the proposed protocol, adversarial capabilities and the formal security model. In Section 5, the design details of the proposed protocol are described. Sections 6 and 7 provide the security certificate and performance analysis of the work, respectively. In Section 8 of the paper, I conclude the work and discuss some ideas regarding future research directions.

2. Related Works

E2E communication security has been discussed in many research papers. In 2012, Fereidooni et al. [9] introduced a design of E2E key exchange and encryption protocol for accelerated satellite networks. Another E2E authentication scheme for wearable health monitoring systems proposed by Jiang et al. [10] could assure a secure communication environment for patients and service providers. In Wang et al.'s [11] work, a session key agreement scheme was proposed for E2E security in time-synchronized networks. Liu et al. [12] also conducted research on E2E security authentication protocol of narrowband internet of things (NB-IoT) for a smart grid based on the physical unclonable function. Nashwan [13] presented a two-factor authentication mechanism for E2E healthcare communications in wireless body sensor networks (WBSNs). Perez et al. [14] proposed a client-server E2E key exchange solution for IoT communications in the application layer. A multi-data multi-user E2E encryption scheme designed by Raj and Venugopalachar [15] provided an access control mechanism for electronic health records stored in clouds. In general, there was no secure cross-server solution for E2E user communications introduced in these works.

In recent years, security issues and authentication solutions in the healthcare systems have become prevalent and have attracted a lot of attention from the scientific community [16]. Deebak and Al-Turjman [17] designed a mutual authentication protocol for cloud-based medical healthcare systems, which addresses several security issues found in Ref. [18], such as smart device stolen attack, server spoofing attack, etc. In another work, a multi-factor fast authentication protocol with patient privacy protection for telecare medical information systems (TMISs) was proposed by Hsu et al. [19]. Wang et al. [20] presented an improved authentication protocol, which resolved some weaknesses of Farash et al.'s [21] scheme for smart healthcare in WBSNs. Recently, Le et al. [22] proposed a three-factor key agreement scheme for multiple healthcare services in 6G networks. Although the work was proven to withstand multiple well-known attacks, I found it was designed without the biometrics update function. The Rabin decryption operation in their protocol was no faster than the one of the RSA cryptosystem [23]. Xu et al. [24] proposed another anonymous three-factor authentication protocol with costly fuzzy extraction operation employed. Lin et al. [25] introduced a multi-server key agreement protocol with patient anonymity for 5G IoT healthcare systems. In the protocol, I found that a public parameter (N_i) of the first conveyed transcript was revealed to the public. There was also no timestamp employed in their work, which is not free from denial of service (DoS) attacks. Meshram et al. [26] proposed a password-based user authentication scheme using a smart card based on extended chaotic maps. The server in their protocol stores an additional value (SB_i) after the authentication procedure is complete. This would not be robust against desynchronization problems. Although Lin et al. [25] and Meshram et al. [26] can provide user anonymity, their works cannot achieve user untraceability, as the messages in their proposals contain some fixed parameters; the adversaries may guess the identity of the user based on these values. In addition, Lin et al. [25] cannot prevent lost smart card attacks, as unmasked user credentials are stored on smart cards directly. Shohaimay and Ismail [27] designed a secure ECC-based two-factor remote authentication protocol for cyber–physical system applications. The two-factor authentication mechanism in the protocol presented some security concerns that need addressing. The communication efficiency of their design is not very high considering the four message transcripts conveyed during the login and authentication process.

Given the drawbacks of the above works noted with specific concerns, I am motivated to propose a new protocol, which could address all the stated limitations while providing various communication functionalities. Furthermore, to the best of my knowledge, the proposed protocol is the first to address the security and privacy concerns in DNA-based healthcare systems enabled by onsite sequencing services.

3. Technical Preliminaries

This section discusses some important technical preliminaries, including the smart card technology, biohash function, elliptic curve cryptography, advanced encryption standard and the main cryptographic notations used in the paper.

3.1. Smart Card Technology

The modern smart card is designed with an embedded integrated circuit chip as either a secure microcontroller or an equivalent intelligence [28]. The card makes use of an internal memory; it can connect to a reader through physical contact or through the contactless radio frequency technique. Smart cards can store large amounts of data; moreover, they can carry out their own on-card functions, including data encryption or verification. For convenience, I recommend using a Bluetooth smart card token or a Bluetooth smart card reader in the design. In the proposed protocol, a smart card is the second factor (something one has) of a three-factor authentication mechanism.

3.2. Biohash Function

The biohash function maps the individuals' biometrics to specific binary strings, providing the tolerance of noise [29]. The biohash function provides the same security as the one-way hash functions [23]. In the proposed protocol, the biohash function is employed to tolerate a noisy biometric template, which results in a flaw in some existing works, e.g., a biometric authentication protocol proposed by Wong et al. [30]. The function also addresses the efficiency problem of the related ideas, e.g., the fuzzy extractor used in the work of Zhang et al. [29].

Definition 1. *Suppose B_i is the original biometric template of an individual and B'_i the newly input one. The input B'_i is not identical to B_i, but the difference is within an acceptable threshold. We obtain $h_{bio}(B_i) = h_{bio}(B'_i)$ given a biohash function h_{bio}.*

3.3. Elliptic Curve Cryptography (ECC)

The ECC is an asymmetric cryptosystem, which offers better performance compared with traditional systems because it employs a smaller key size with the same security [31]. Therefore, ECC-based authentication protocols are highly suitable for mobile devices in many applications scenarios. The security of the ECC is based on the elliptic curve discrete logarithm problem (ECDLP) and the elliptic curve computational Diffie–Hellman problem (ECCDHP), which are two security assumptions used in the proposed protocol. Suppose there is an elliptic curve over a finite field $Fp\ Ep(a,\ b):\ y^2 = x^3 + ax + b(mod\ p)$.

Definition 2. *Given an integer $k \in Z_p$ and a point $P_{(x,\ y)} \in E_p$, it is easy to compute $Q_{(x,\ y)} = k.P_{(x,\ y)} \in E_p$. However, due to the ECDLP, it is computationally difficult to find the scalar k, such that $Q_{(x,\ y)} = k.P_{(x,\ y)}$ given $P_{(x,\ y)}$ and $Q_{(x,\ y)}$.*

Definition 3. *Given two integers $s, t \in Z_p$ and three points $P_{(x,\ y)}, s.P_{(x,\ y)}, t.P_{(x,\ y)} \in E_p$, the ECCDHP is used to find the point $s.t.P_{(x,\ y)} \in E_p$ given $s.P_{(x,\ y)}$ and $t.P_{(x,\ y)}$.*

3.4. Advanced Encryption Standard (AES)

The AES [32] is a symmetric encryption technique, which provides a high degree of security. AES encryption converts data into an unintelligible form, called ciphertext. Conversely, the decryption converts this ciphertext into its original form, called plaintext. The AES algorithm can generate block ciphertexts of 128 bits, with three different key sizes, namely, 128, 192 and 256 bits. If the AES encryption key is an EC point in the beginning, it can be transformed into an integer (e.g., 256-bit key) through hashing the point's x and y coordinates for the subsequent process.

3.5. Notations and Cryptographic Functions

Table 1 explains the notations and main cryptographic functions used in the proposed protocol.

Table 1. Notations and cryptographic functions used in the paper.

Notation	Description
S_m	m^{th} server
P_i	i^{th} patient
prk_m, puk_m	Private key, public key of S_m
$Cert_m$	Certificate of S_m
$\delta_{m,i}$	Signature of P_i's message signed by S_m
$G_{(x,y)}$	Basic point on the curve $Ep(a, b)$
ID_i	Identity of P_i
PW_i	Password of P_i
B_i	Biometrics of P_i
T	Timestamp
$\|\|$	Concatenation operation
$\oplus$	Exclusive-or (XOR) operation
$h(.), h_{bio}(.)$	One-way hash function, biohash function
$E_k(.), D_k(.)$	Symmetric encryption, decryption algorithms using key k
$[.]_{SC_i}$	Storage parameters in P_i's smart card
$[.]_{MD_i}$	Storage parameters in P_i's mobile device

4. Problem Statement

In this section, the system model of the proposed protocol along with the communication problem is presented. I also discuss some adversarial capabilities and describe the formal security model used in the paper.

4.1. System Model

The proposed model includes four main entities in the communication, namely patient P_i, server S_m, patient P_j and server S_n, who are communicating in a multi-server U-healthcare environment. As depicted in Figure 1, the patient P logs into and accesses services from multiple servers S. The DNA-based U-healthcare services include monitoring of body fluids, virus control, understanding disease mechanism at the molecular level, etc. [3]. The IoLT network consists of multiple DNA-reading sensors worn by P. The sensing data containing DNA samples produced by the sensors are transmitted to P's mobile device with the support of a wireless technology, e.g., Bluetooth, Wi-Fi or Zigbee. Thereafter, a sequencing process is run by a tiny sequencer connected to the mobile device. Through an open internet, the sequences produced are sent to S (e.g., doctors, data scientists, etc.) for further processing and analysis services. For example, in the monitoring of body fluids mentioned, S is allowed to keep an eye on their P's health via blood, sweat and saliva samples. The analysis results would be transmitted back to P upon their specific request. Some related services, for instance, a WBSN, can be integrated to improve the overall healthcare process and possible medical treatments. It is recommended that the communications in the proposed system model are aided with 5G or 6G mobile technology to achieve a truly real-time healthcare process [22]. Mobile devices (smart phones, tablets, etc.) are now known for their simplicity, robustness and advanced connectivity, with many brands supporting 5G. They would also support 6G, which is expected to be introduced in 2030 [33].

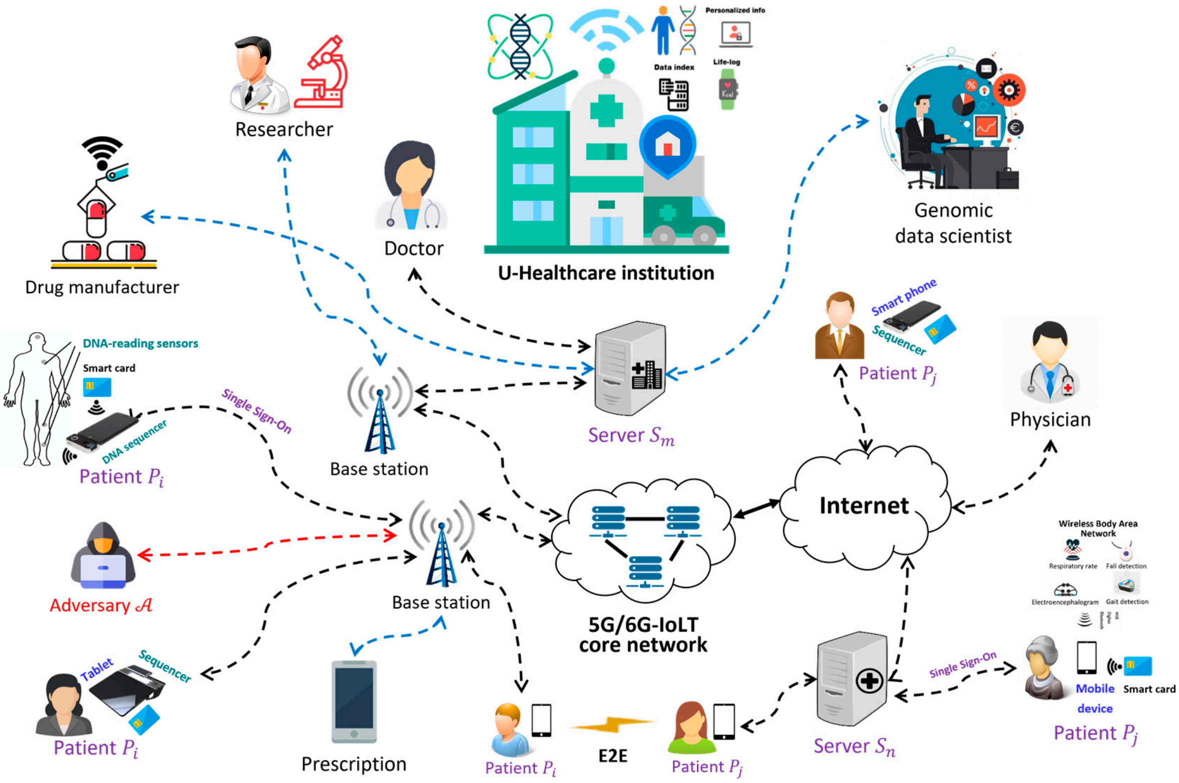

Figure 1. System model of the proposed protocol.

In the model, two patients P_i and P_j want to share personal medical information or U-healthcare data with each other. For example, a family member may wish to know the health status of another one. Since this communication is carried out via a public channel, data security and patient privacy become prominent concerns. To this end, the proposed protocol allows P_i and P_j to compute an authenticated E2E session key used to protect their communicated messages. My work designs a three-factor authentication mechanism where a patient uses a mobile device and a smart card to register with respective servers. P_i carries out a SC-SSO, which uses a single set of registered credentials to log into the system and directly communicate with P_j through the assistance of S_m and S_n.

4.2. Adversarial Capabilities

Possible attacks in healthcare systems may result in tremendous consequences and damage, including patient security violation, reduced service reliability, etc. The attacks may also affect the treatment processes and harm patients' health [19]. Upon various potential risks that I have observed, an adversary $\mathcal{A}$ may have the following attack capabilities.

- $\mathcal{A}$ has control over the open internet. This means that $\mathcal{A}$ can intercept, delete, insert or replay any transcript in each communication session.
- $\mathcal{A}$ may steal the patients' smart card and/or mobile device and then attempt to extract the secret credentials using power analysis [34].
- $\mathcal{A}$ attempts to compromise the past messages communicated between patients once they have obtained secret values or even a session key of the current communication session.
- $\mathcal{A}$ is a privileged insider of the system (e.g., admin) who may attempt to attack the patient's registered information stored in S's database.
- Legitimate patients or servers can behave as $\mathcal{A}$ and trigger similar attacks on the system.

4.3. Formal Security Model

The real-or-random (RoR) model is employed to provide the formal security proof of this work. The model is a well-known tool used to analyze the probability of the adversary breaking the cryptographic schemes [35]. In the model concerned, there exist two communicating parties, namely patient P and provider server S, which is consistent with the entities of the proposed protocol. They carry out the communications via an open internet channel. In the model, $\mathbb{C}$ is a protocol challenger, and M is a message communicated by P and S. $\mathcal{A}$ would execute the following queries to launch various attacks.

- *Send*($\mathbb{C}$, M): $\mathcal{A}$ is allowed to request M to $\mathbb{C}$; $\mathbb{C}$ replies to $\mathcal{A}$ in accordance with the rules of the proposed protocol.
- *Execute*(P, S): This passive attack allows $\mathcal{A}$ to eavesdrop on the message communicated by P and S.
- *Reveal*($\mathbb{C}$): In this attack, $\mathcal{A}$ attempts to retrieve the session key generated by $\mathbb{C}$ based on the rules of the protocol.
- $Corrupt(P, x)$: In my proposed protocol, this query returns the password of the patient, the biometrics of the patient and the parameters stored on the smart card and the device to $\mathcal{A}$ if $x = 1$, $x = 2$ and $x = 3$, respectively.
- *Test*($\mathbb{C}$): This query allows $\mathcal{A}$ to request the session key from $\mathbb{C}$; $\mathbb{C}$ replies to $\mathcal{A}$ based on the probabilistic outcome of the coin c tossed.

Definition 4. *Let $Adv_{\mathbb{C}}^{DNAHC}$ be the advantage of $\mathcal{A}$ running in polynomial time in semantically breaking the security system of the proposed protocol, where $NDAHC$ denotes the protocol (for DNA-based U-healthcare). We obtain $Adv_{\mathbb{C}}^{DNAHC} = |2Pr[c' = c] - 1|$, where c' is the guessed bit of the session key.*

5. The Proposed Protocol

Patient P_i directly communicates with patient P_j with the assistance of both servers S_m and S_n. The proposed protocol consists of four phases: system initialization phase, registration phase, login and authentication phase, and password and biometrics update phase. All the parties, including P_i, S_m, P_j, S_n, participate in the communication, so that P_i and P_j can compute a shared E2E session key. Since the communication between P_i and S_m is identical with the one between P_j and S_n, for simplicity, only the communication between P_i and S_m is presented in the registration phase and in the password and biometrics update phase.

5.1. System Initialization Phase

My proposed protocol employs the ECC proposed by the National Institute of Standards and Technology (NIST) [36]. The system generates a curve over a finite field Fp $Ep(a,\ b):\ y^2 = x^3 + ax + b(mod\ p)$ with the point $G_{(x,\ y)}$. For simplicity, two coordinates x and y of $G_{(x,\ y)}$ are ignored in the description of the protocol. S_m chooses a private key prk_m and computes its public key $puk_m = prk_m.G$. Next, S_m registers with a certificate authority and has the certificate, signature, public key and private key validated. The same procedure is also conducted by S_n.

5.2. Registration Phase

This procedure is carried out in a secure channel. P_i is allowed to register with S_m to become a legitimate service user. As shown in Figure 2, P_i and S_m perform the following steps for registration.

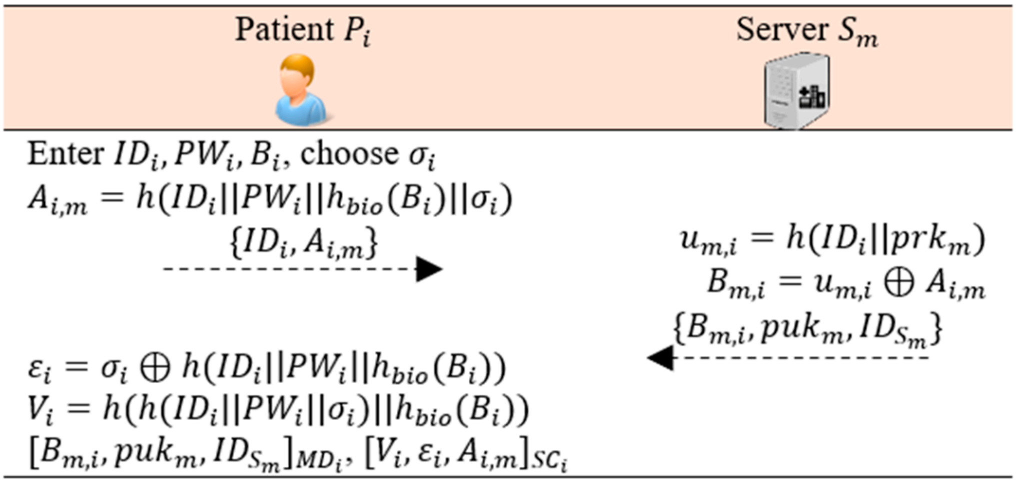
Patient P_i
Server S_m
Enter ID_i, PW_i, B_i, choose σ_i
$A_{i,m} = h(ID_i||PW_i||h_{bio}(B_i)||\sigma_i)$
$\{ID_i, A_{i,m}\}$
$u_{m,i} = h(ID_i||prk_m)$
$B_{m,i} = u_{m,i} \oplus A_{i,m}$
$\{B_{m,i}, puk_m, ID_{S_m}\}$
$\varepsilon_i = \sigma_i \oplus h(ID_i||PW_i||h_{bio}(B_i))$
$V_i = h(h(ID_i||PW_i||\sigma_i)||h_{bio}(B_i))$
$[B_{m,i}, puk_m, ID_{S_m}]_{MD_i}$, $[V_i, \varepsilon_i, A_{i,m}]_{SC_i}$

Figure 2. Registration procedure of the proposed protocol.

*Step R*1: P_i enters the identity ID_i, password PW_i and the biometrics B_i. P_i selects a random number σ_i and computes $A_{i,m} = h(ID_i||PW_i||h_{bio}(B_i)||\sigma_i)$. Next, P_i sends $\{ID_i, A_{i,m}\}$ to S_m.

*Step R*2: Upon receiving $\{ID_i, A_{i,m}\}$, S_m computes $u_{m,i} = h(ID_i||prk_m)$ and $B_{m,i} = u_{m,i} \oplus A_{i,m}$. Next, S_m sends $\{B_{m,i}, puk_m, ID_{S_m}\}$ to P_i.

*Step R*3: Upon receiving $\{B_{m,i}, puk_m\}$, P_i computes $\varepsilon_i = \sigma_i \oplus h(ID_i||PW_i||h_{bio}(B_i))$ and $V_i = h(h(ID_i||PW_i||\sigma_i)||h_{bio}(B_i))$. Finally, P_i stores $\{B_{m,i}, puk_m, ID_{S_m}\}$ and $\{V_i, \varepsilon_i, A_{i,m}\}$ on the mobile device MD_i and the smart card SC_i, respectively.

5.3. Login and Authentication Phase

This phase is conducted via an unreliable channel, where P_i and P_j log in and mutually authenticate with S_m and S_n, respectively. P_i and P_j also authenticate with each other and compute a shared session key through the assistance of S_m and S_n. Figure 3 shows the whole procedure in this phase.

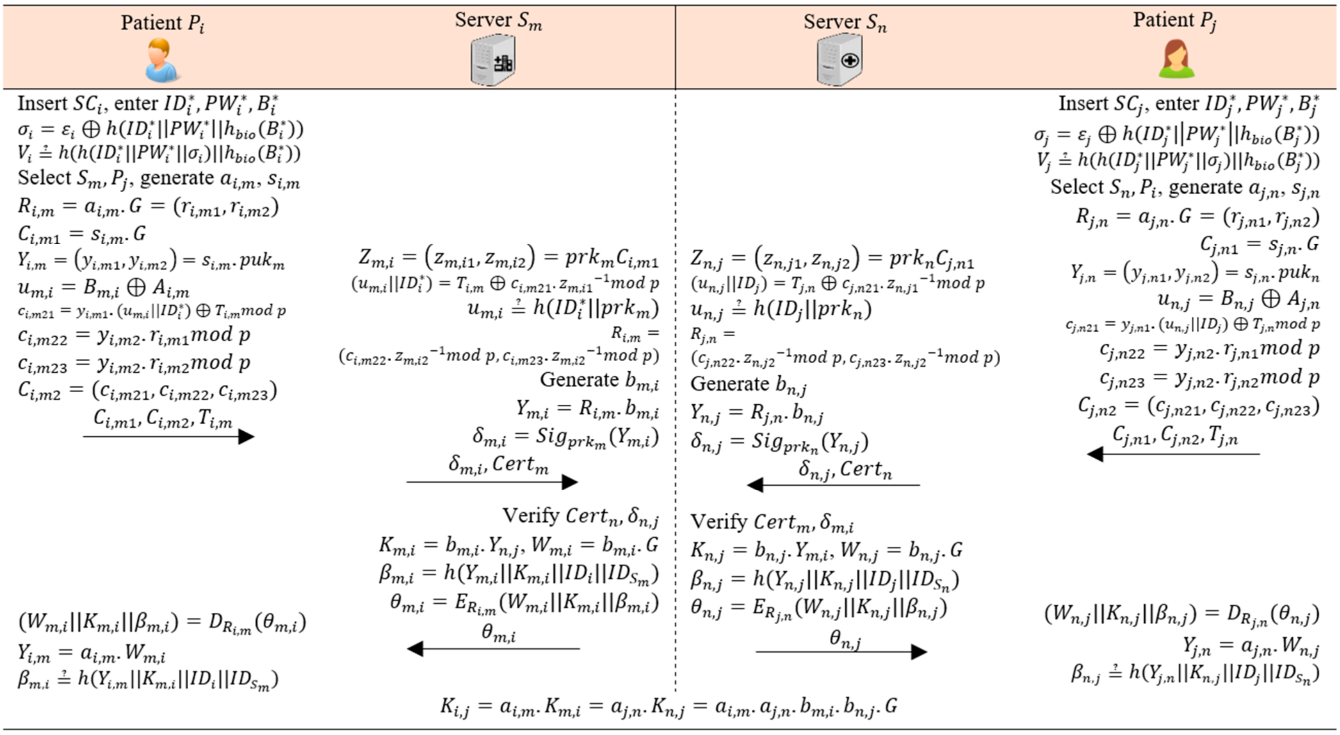
Patient P_i
Server S_m
Server S_n
Patient P_j
Insert SC_i, enter ID_i^*, PW_i^*, B_i^*
$\sigma_i = \varepsilon_i \oplus h(ID_i^*||PW_i^*||h_{bio}(B_i^*))$
$V_i \overset{?}{=} h(h(ID_i^*||PW_i^*||\sigma_i)||h_{bio}(B_i^*))$
Select S_m, P_j, generate $a_{i,m}, s_{i,m}$
$R_{i,m} = a_{i,m}.G = (r_{i,m1}, r_{i,m2})$
$C_{i,m1} = s_{i,m}.G$
$Y_{i,m} = (y_{i,m1}, y_{i,m2}) = s_{i,m}.puk_m$
$u_{m,i} = B_{m,i} \oplus A_{i,m}$
$c_{i,m21} = y_{i,m1}.(u_{m,i}||ID_i^*) \oplus T_{i,m} mod\ p$
$c_{i,m22} = y_{i,m2}.r_{i,m1} mod\ p$
$c_{i,m23} = y_{i,m2}.r_{i,m2} mod\ p$
$C_{i,m2} = (c_{i,m21}, c_{i,m22}, c_{i,m23})$
$C_{i,m1}, C_{i,m2}, T_{i,m}$
$Z_{m,i} = (z_{m,i1}, z_{m,i2}) = prk_m C_{i,m1}$
$(u_{m,i}||ID_i^*) = T_{i,m} \oplus c_{i,m21}.z_{m,i1}{}^{-1} mod\ p$
$u_{m,i} \overset{?}{=} h(ID_i^*||prk_m)$
$R_{i,m} =$
$(c_{i,m22}.z_{m,i2}{}^{-1} mod\ p, c_{i,m23}.z_{m,i2}{}^{-1} mod\ p)$
Generate $b_{m,i}$
$Y_{m,i} = R_{i,m}.b_{m,i}$
$\delta_{m,i} = Sig_{prk_m}(Y_{m,i})$
$\delta_{m,i}, Cert_m$
Verify $Cert_n, \delta_{n,j}$
$K_{m,i} = b_{m,i}.Y_{n,j}, W_{m,i} = b_{m,i}.G$
$\beta_{m,i} = h(Y_{m,i}||K_{m,i}||ID_i||ID_{S_m})$
$\theta_{m,i} = E_{R_{i,m}}(W_{m,i}||K_{m,i}||\beta_{m,i})$
$\theta_{m,i}$
$(W_{m,i}||K_{m,i}||\beta_{m,i}) = D_{R_{i,m}}(\theta_{m,i})$
$Y_{i,m} = a_{i,m}.W_{m,i}$
$\beta_{m,i} \overset{?}{=} h(Y_{i,m}||K_{m,i}||ID_i||ID_{S_m})$
$Z_{n,j} = (z_{n,j1}, z_{n,j2}) = prk_n C_{j,n1}$
$(u_{n,j}||ID_j) = T_{j,n} \oplus c_{j,n21}.z_{n,j1}{}^{-1} mod\ p$
$u_{n,j} \overset{?}{=} h(ID_j||prk_n)$
$R_{j,n} =$
$(c_{j,n22}.z_{n,j2}{}^{-1} mod\ p, c_{j,n23}.z_{n,j2}{}^{-1} mod\ p)$
Generate $b_{n,j}$
$Y_{n,j} = R_{j,n}.b_{n,j}$
$\delta_{n,j} = Sig_{prk_n}(Y_{n,j})$
$\delta_{n,j}, Cert_n$
Verify $Cert_m, \delta_{m,i}$
$K_{n,j} = b_{n,j}.Y_{m,i}, W_{n,j} = b_{n,j}.G$
$\beta_{n,j} = h(Y_{n,j}||K_{n,j}||ID_j||ID_{S_n})$
$\theta_{n,j} = E_{R_{j,n}}(W_{n,j}||K_{n,j}||\beta_{n,j})$
$\theta_{n,j}$
Insert SC_j, enter ID_j^*, PW_j^*, B_j^*
$\sigma_j = \varepsilon_j \oplus h(ID_j^*||PW_j^*||h_{bio}(B_j^*))$
$V_j \overset{?}{=} h(h(ID_j^*||PW_j^*||\sigma_j)||h_{bio}(B_j^*))$
Select S_n, P_i, generate $a_{j,n}, s_{j,n}$
$R_{j,n} = a_{j,n}.G = (r_{j,n1}, r_{j,n2})$
$C_{j,n1} = s_{j,n}.G$
$Y_{j,n} = (y_{j,n1}, y_{j,n2}) = s_{j,n}.puk_n$
$u_{n,j} = B_{n,j} \oplus A_{j,n}$
$c_{j,n21} = y_{j,n1}.(u_{n,j}||ID_j) \oplus T_{j,n} mod\ p$
$c_{j,n22} = y_{j,n2}.r_{j,n1} mod\ p$
$c_{j,n23} = y_{j,n2}.r_{j,n2} mod\ p$
$C_{j,n2} = (c_{j,n21}, c_{j,n22}, c_{j,n23})$
$C_{j,n1}, C_{j,n2}, T_{j,n}$
$(W_{n,j}||K_{n,j}||\beta_{n,j}) = D_{R_{j,n}}(\theta_{n,j})$
$Y_{j,n} = a_{j,n}.W_{n,j}$
$\beta_{n,j} \overset{?}{=} h(Y_{j,n}||K_{n,j}||ID_j||ID_{S_n})$
$K_{i,j} = a_{i,m}.K_{m,i} = a_{j,n}.K_{n,j} = a_{i,m}.a_{j,n}.b_{m,i}.b_{n,j}.G$

Figure 3. Login and authentication procedure of the proposed protocol.

Step A1: P_i inserts SC_i, enters ID_i^*, PW_i^*, B_i^* and computes $\sigma_i = \varepsilon_i \oplus h(ID_i^*||PW_i^*||h_{bio}(B_i^*))$. The SC_i verifies whether $V_i \overset{?}{=} h(h(ID_i^*||PW_i^*||\sigma_i)||h_{bio}(B_i^*))$. If there is a match, this allows P_i to select a server S_m from an app interface for logging into a specific service and to select a P_j that P_i wishes to communicate with. Next, P_i generates two random numbers $a_{i,m}$, $s_{i,m}$ and a timestamp $T_{i,m}$ and computes $R_{i,m} = a_{i,m}.G = (r_{i,m1}, r_{i,m2})$, $C_{i,m1} = s_{i,m}.G$, $Y_{i,m} = (y_{i,m1}, y_{i,m2}) = s_{i,m}.puk_m$, $u_{m,i} = B_{m,i} \oplus A_{i,m}$, $c_{i,m21} = y_{i,m1}.(u_{m,i}||ID_i) \oplus T_{i,m} mod\ p$, $c_{i,m22} = y_{i,m2}.r_{i,m1} mod\ p$, $c_{i,m23} = y_{i,m2}.r_{i,m2} mod\ p$ and $C_{i,m2} = (c_{i,m21}, c_{i,m22}, c_{i,m23})$. P_i sends a message $\{C_{i,m1}, C_{i,m2}, T_{i,m}\}$ to S_m as a login request.

Step A2: Upon receiving the login request message, S_m computes $Z_{m,i} = (z_{m,i1}, z_{m,i2}) = prk_m C_{i,m1}$ and $(u_{m,i}||ID_i) = T_{i,m} \oplus c_{i,m21}.z_{m,i1}{}^{-1} mod\ p$. S_m then checks whether $u_{m,i} \overset{?}{=} h(ID_i||prk_m)$ to confirm the authenticity of P_i. Next, S_m generates a random number $b_{m,i}$ and computes $R_{i,m} = (c_{i,m22}.z_{m,i2}{}^{-1} mod\ p,\ c_{i,m23}.z_{m,i2}{}^{-1} mod\ p)$, $Y_{m,i} = R_{i,m}.b_{m,i}$ and signature $\delta_{m,i} = Sig_{prk_m}(Y_{m,i})$. Thereafter, S_m sends $\{\delta_{m,i}$, certificate $Cert_m\}$ to S_n and waits for the message $\{\delta_{n,j}$, certificate $Cert_n\}$ sent by S_n.

Step A3: Upon receiving the message, S_m verifies $Cert_n, \delta_{n,j}$ using the public key of S_n. If the verification is successful, S_m computes $K_{m,i} = b_{m,i}.Y_{n,j}$, $W_{m,i} = b_{m,i}.G$, $\beta_{m,i} = h(Y_{m,i}||K_{m,i}||ID_i||ID_{S_m})$ and ciphertext $\theta_{m,i} = E_{R_{i,m}}(W_{m,i}||K_{m,i}||\beta_{m,i})$. Next, S_m sends $\{\theta_{m,i}\}$ to P_i.

Step A4: Upon receiving the message, P_i obtains $W_{m,i}, K_{m,i}, \beta_{m,i}$ by symmetrically decrypting $\theta_{m,i}$ using the key $R_{i,m}$. Next, P_i computes $Y_{i,m} = a_{i,m}.W_{m,i}$ and verifies whether $\beta_{m,i} \overset{?}{=} h(Y_{i,m}||K_{m,i}||ID_i||ID_{S_m})$.

Session key establishment: A similar procedure is carried out by P_j and S_n, so that P_j can obtain a legitimate $K_{n,j}$. Thereafter, P_i and P_j compute a common key $K_{i,j} = a_{i,m}.K_{m,i} = a_{j,n}.K_{n,j} = a_{i,m}.a_{j,n}.b_{m,i}.b_{n,j}.G$. In this way, a shared patient E2E session key $K_{i,j}$ is established.

5.4. *Password and Biometrics Update Phase*

In this phase, P_i updates their password and biometrics stored in SC_i to enhance the security. As depicted in Figure 4, the procedure is performed as follows.

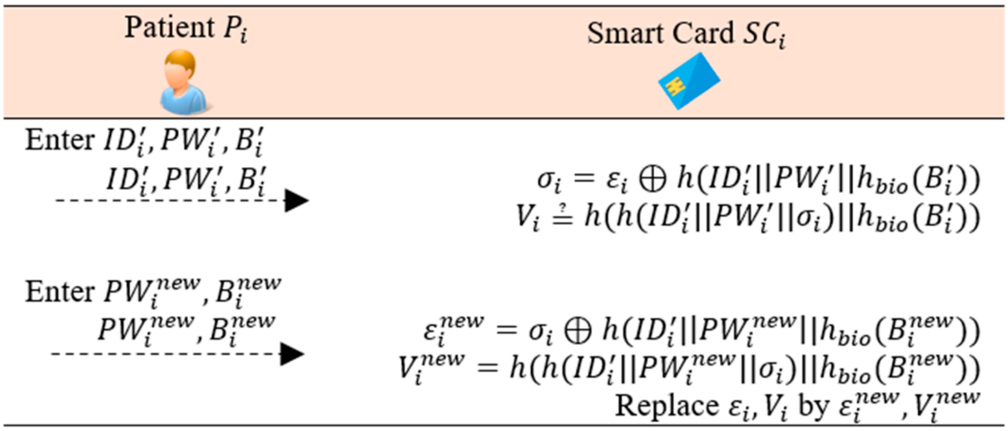
Patient P_i
Smart Card SC_i
Enter ID_i', PW_i', B_i'
ID_i', PW_i', B_i'
$\sigma_i = \varepsilon_i \oplus h(ID_i'||PW_i'||h_{bio}(B_i'))$
$V_i \overset{?}{=} h(h(ID_i'||PW_i'||\sigma_i)||h_{bio}(B_i'))$
Enter PW_i^{new}, B_i^{new}
PW_i^{new}, B_i^{new}
$\varepsilon_i^{new} = \sigma_i \oplus h(ID_i'||PW_i^{new}||h_{bio}(B_i^{new}))$
$V_i^{new} = h(h(ID_i'||PW_i^{new}||\sigma_i)||h_{bio}(B_i^{new}))$
Replace ε_i, V_i by $\varepsilon_i^{new}, V_i^{new}$

Figure 4. Password and biometrics update procedure of the proposed protocol.

Step U1: P_i enters ID_i', PW_i', B_i' into SC_i. SC_i computes $\sigma_i = \varepsilon_i \oplus h\left(ID_i'||PW_i'||h_{bio}\left(B_i'\right)\right)$. Next, SC_i verifies whether $V_i \overset{?}{=} h\left(h\left(ID_i'||PW_i'||\sigma_i\right)||h_{bio}\left(B_i'\right)\right)$. If there is a match, SC_i requests P_i to enter new credentials PW_i^{new}, B_i^{new}.

Step U2: Upon receiving PW_i^{new}, B_i^{new}, SC_i computes $\varepsilon_i^{new} = \sigma_i \oplus h\left(ID_i'||PW_i^{new}||h_{bio}\left(B_i^{new}\right)\right)$ and $V_i^{new} = h\left(h\left(ID_i'||PW_i^{new}||\sigma_i\right)||h_{bio}\left(B_i^{new}\right)\right)$. Finally, SC_i replaces ε_i, V_i with $\varepsilon_i^{new}, V_i^{new}$. The new password and biometrics are provided to the smart card.

6. Security Analysis

This section provides a security evaluation of my proposed protocol. RoR model, BAN logic and an informal analysis are included in the analysis. First, the success probability of $\mathcal{A}$ in attacking the protocol is analyzed with the standard RoR model. Thereafter, a mutual authentication proof of communication between the patients is presented using the BAN logic. Finally, a semantic security analysis provides further insight into various possible attacks, which can be prevented in the protocol.

6.1. *Formal Security Analysis Using RoR Model*

As mentioned, I provide the formal security proof of the protocol using the widely accepted ROR model. The analysis is primarily presented for the communication between P_i and S_m. The communication between P_j and S_n can also be achieved using similar arguments, so that E2E communication between P_i and P_j is provably secure. In this proof, several games are included where $\mathcal{A}$ makes various queries discussed in Section 4.3 in order to perform the attacks. The following are the notations used in the proof.

- L_{hash}: Length of a hash value.
- L_{number}: Length of a random number.
- $L_{biometrics}$: Length of a biometrics value.
- q_{hash}: Total number of hash oracle queries.
- q_{send}: Total number of *Send* queries.
- $q_{execute}$: Total number of *Execute* queries.
- l_h: List of hash oracle outputs.
- l_o: List of random oracle results.
- l_m: List of communicated messages between P_i and S_m.
- $\varepsilon_{biometrics}$: Probability of biometrics false positive.
- C', s': Zipf parameters.

Definition 5. *$\mathbb{C}$ enters an accepted state after receiving the last message in the session. All communicated messages $M_1 = \{C_{i,m1}, C_{i,m2}, T_{i,m}\}$ and $M_2 = \{\theta_{m,i}\}$ are concatenated, forming a session with the identification "s_id".*

Definition 6. *There are some conditions for $P_i^{T_c}$ and ${S_m}^{T_c^*}$, as follows: (1) they are in an accepted state; (2) they mutually authenticate each other in the same session s_id; and (3) both are mutual partners of each other. $P_i^{T_c}$ and ${S_m}^{T_c^*}$ are called "partners" if they simultaneously satisfy all the conditions.*

Definition 7. *There are some conditions for $\mathbb{C}$, as follows: (1) $\mathbb{C}$ is in an accepted state; (2) the query Reveal($\mathbb{C}$) was never submitted; and (3) fewer than two Corrupt(P_i, x) queries were submitted. $\mathbb{C}$ can satisfy the freshness rule if $\mathbb{C}$ simultaneously meets all the conditions. In fact, my protocol would still be safe even if $\mathcal{A}$ submits queries "Corrupt$(P_i, 1)$ and Corrupt$(P_i, 3)$ " or "Corrupt$(P_i, 2)$ and Corrupt$(P_i, 3)$ ", since $\mathcal{A}$ is not able to compromise the masked credentials stored on the smart card.*

Definition 8. *Let $Adv_{\mathcal{A}}^{ECDLP}(t_{\mathcal{A}})$ be the advantage of $\mathcal{A}$ in breaking the ECDLP assumption. Since the assumption holds, $Adv_{\mathcal{A}}^{ECDLP}(t_{\mathcal{A}})$ is defined as a negligible probability with execution time $t_{\mathcal{A}}$.*

Definition 9. *Let $Adv_{\mathcal{A}}^{ECCDHP}(t_{\mathcal{A}})$ be the advantage of $\mathcal{A}$ in breaking the ECCDHP assumption. Similarly, $Adv_{\mathcal{A}}^{ECCDHP}(t_{\mathcal{A}})$ is defined as a negligible probability with execution time $t_{\mathcal{A}}$.*

Definition 10. *The value $max\left\{C'.q_{send}^{s'}, q_{send}\left(\frac{1}{2^{L_{biometrics}}}, \varepsilon_{biometrics}\right)\right\}$ is sufficiently small, so that $\mathcal{A}$ cannot guess the credentials of P_i [19].*

Theorem 1. *Since $\mathcal{A}$ has the following negligible probability of breaking our security system, the proposed protocol is semantically secure.*

$$\begin{aligned} Adv_{\mathbb{C}}^{DNAHC} \leq & \frac{(q_{send}+q_{execute})^3+6q_{send}}{2^{L_{number}}} + \frac{{q_{hash}}^2+14q_{hash}}{2^{L_{hash}}} \\ & +2max\left\{\left(C'.q_{send}^{s'}\right), q_{send}\left(\frac{1}{2^{L_{biometrics}}}, \varepsilon_{biometrics}\right)\right\} \\ & +6q_{hash}(q_{send}+q_{execute}+1)Adv_{\mathcal{A}}^{ECDLP}(t_{\mathcal{A}}) \\ & +2q_{hash}(q_{send}+q_{execute}+1)Adv_{\mathcal{A}}^{ECCDHP}(t_{\mathcal{A}}) \end{aligned} \tag{1}$$

Proof. Six simulated games are included in the proof, namely $Game_0, Game_1, Game_2, Game_3, Game_4, Game_5$, so that the success probability of $\mathcal{A}$'s attack gradually increases. The ultimate purpose of $\mathcal{A}$ is to retrieve the bit c with the *Test* query after each game finishes. $Pr[S_i]$ denotes the success probabilities where S_i ($i = 0, 1, 2, 3, 4, 5$) are the events in different games. A protocol simulator $\mathcal{B}$ is set to play the role of the challenger $\mathbb{C}$.

$Game_0$: This game starts the simulation, and it is identical to the real protocol in the random oracles. c is tossed by $\mathcal{B}$ to start the game. We have

$$Adv_{\mathbb{C}}^{DNAHC} = |2Pr[S_0] - 1| \tag{2}$$

$Game_1$: This game presents all the queries discussed in Section 4.3. Table 2 describes a simulation of the queries in accordance with the rule of the proposed protocol. G_1 creates three lists: l_h, l_r and l_m. Because of the indistinguishability between G_0 and G_1, we obtain

$$Pr[S_1] = Pr[S_0] \tag{3}$$

Table 2. Simulation of the Hash, Reveal, Test, Corrupt, Execute and Send oracle queries.

The *Hash* query is simulated as follows, where $\boldsymbol{M_i}$ is a message.
If the record $(\boldsymbol{M_i}, \boldsymbol{h(M_i)})$ is found in the list $\boldsymbol{l_h}$, return $\boldsymbol{h(M_i)}$;
otherwise, choose a $\boldsymbol{h'(M_i)} \in \boldsymbol{Z_p^*}$ and add $(\boldsymbol{M_i}, \boldsymbol{h'(M_i)})$ into $\boldsymbol{l_h}$;
in this way, a similar procedure is performed to create $\boldsymbol{l_o}$.

Simulation of the *Reveal*($\mathbb{C}$) query is simply performed as follows.
Once $\mathbb{C}$ is in an accepted state, the session key formed by $\mathbb{C}$ is returned.

Simulation of the *Test*($\mathbb{C}$) query is performed as follows.
$\mathbb{C}$tosses the coin $\boldsymbol{c}$. If $\boldsymbol{c = 1}$, the query returns an available $\boldsymbol{SK}$; otherwise, the query returns a random number.

The query *Corrupt*$(\boldsymbol{P_i}, \boldsymbol{x})$ is simulated as follows.
If $\boldsymbol{x = 1}$, the query outputs $\boldsymbol{PW_i}$.
If $\boldsymbol{x = 2}$, the query outputs $\boldsymbol{B_i}$.
If $\boldsymbol{x = 3}$, the query outputs the parameters stored in $\boldsymbol{SC_i}$ or $\boldsymbol{MD_i}$.

Simulation of the *Execute*$(\boldsymbol{P_i}, \boldsymbol{S_m})$ query occurs in succession to simulation of the *Send*$(\mathbb{C}, \boldsymbol{M_i})$ query, which is described as follows.
$\boldsymbol{P_i}$ sends $\boldsymbol{M_1}$ to $\boldsymbol{S_m}$, and $\boldsymbol{S_m}$ sends $\boldsymbol{M_2}$ to $\boldsymbol{P_i}$. We have: $<\boldsymbol{C_{i,m1}}, \boldsymbol{c_{i,m21}}, \boldsymbol{c_{i,m22}}, \boldsymbol{c_{i,m23}}, \boldsymbol{T_{i,m}}> \leftarrow$ *Send*$(\boldsymbol{P_i}$, *start*), $<\{\boldsymbol{W_{m,i}}||\boldsymbol{K_{m,i}}||\boldsymbol{h(Y_{m,i}}||\boldsymbol{K_{m,i}}||\boldsymbol{ID_i}||\boldsymbol{ID_{S_m}})\}_{\boldsymbol{R_{i,m}}}> \leftarrow$ *Send*$(\boldsymbol{S_m}$,
$<\boldsymbol{C_{i,m1}}, \boldsymbol{c_{i,m21}}, \boldsymbol{c_{i,m22}}, \boldsymbol{c_{i,m23}}, \boldsymbol{T_{i,m}}>$)
Finally, $\boldsymbol{M_1} = \langle \boldsymbol{C_{i,m1}}, \boldsymbol{c_{i,m21}}, \boldsymbol{c_{i,m22}}, \boldsymbol{c_{i,m23}}, \boldsymbol{T_{i,m}} \rangle$ and
$\boldsymbol{M_2} = \left\langle \{\boldsymbol{W_{m,i}}||\boldsymbol{K_{m,i}}||\boldsymbol{h(Y_{m,i}}||\boldsymbol{K_{m,i}}||\boldsymbol{ID_i}||\boldsymbol{ID_{S_m}})\}_{\boldsymbol{R_{i,m}}} \right\rangle$ are returned.

Following the rules of the proposed protocol, the *Send* query is executed below.

- $\mathcal{A}$ creates a *Send*$(\boldsymbol{P_i}$, *start*) query; $\mathbb{C}$ replies to $\mathcal{A}$ as follows. $\mathbb{C}$ computes $\boldsymbol{C_{i,m1} = s_{i,m}.G}$, $\boldsymbol{c_{i,m21} = y_{i,m1}.(u_{i,m}||ID_i) \oplus T_{i,m} mod\ p}$, $\boldsymbol{c_{i,m22} = y_{i,m2}.r_{i,m1} mod\ p}$, $\boldsymbol{c_{i,m23} = y_{i,m2}.r_{i,m2} mod\ p}$, chooses $\boldsymbol{T_{i,m}}$ and outputs $\boldsymbol{M_1} = \langle \boldsymbol{C_{i,m1}}, \boldsymbol{c_{i,m21}}, \boldsymbol{c_{i,m22}}, \boldsymbol{c_{i,m23}}, \boldsymbol{T_{i,m}} \rangle$.
- $\mathcal{A}$ creates a *Send*$(\boldsymbol{S_m}, \langle \boldsymbol{C_{i,m1}}, \boldsymbol{c_{i,m21}}, \boldsymbol{c_{i,m22}}, \boldsymbol{c_{i,m23}}, \boldsymbol{T_{i,m}} \rangle)$ query; $\mathbb{C}$ replies to $\mathcal{A}$ as follows. $\mathbb{C}$ computes $\boldsymbol{Z_{m,i} = prk_m C_{i,m1}}$, $\boldsymbol{(u_{i,m}||ID_i) = T_{i,m} \oplus c_{i,m21}.z_{m,i1}{}^{-1} mod\ p}$, checks $\boldsymbol{u_{m,i}}$ and calculates the point $\boldsymbol{R_{i,m}}$. The session will be terminated if the check on $\boldsymbol{u_{m,i}}$ does not hold. Otherwise, $\mathbb{C}$ outputs ciphertext $\boldsymbol{M_2} = \left\langle \{\boldsymbol{W_{m,i}}||\boldsymbol{K_{m,i}}||\boldsymbol{h(Y_{m,i}}||\boldsymbol{K_{m,i}}||\boldsymbol{ID_i}||\boldsymbol{ID_{S_m}})\}_{\boldsymbol{R_{i,m}}} \right\rangle$.
- $\mathcal{A}$ creates a *Send*$(\boldsymbol{U_i}, \left\langle \{\boldsymbol{W_{m,i}}||\boldsymbol{K_{m,i}}||\boldsymbol{h(Y_{m,i}}||\boldsymbol{K_{m,i}}||\boldsymbol{ID_i}||\boldsymbol{ID_{S_m}})\}_{\boldsymbol{R_{i,m}}} \right\rangle)$ query; $\mathbb{C}$ replies to $\mathcal{A}$ as follows. $\mathbb{C}$ decrypts $\{\boldsymbol{W_{m,i}}||\boldsymbol{K_{m,i}}||\boldsymbol{h(Y_{m,i}}||\boldsymbol{K_{m,i}}||\boldsymbol{ID_i}||\boldsymbol{ID_{S_m}})\}_{\boldsymbol{R_{i,m}}}$, computes $\boldsymbol{Y_{i,m} = a_{i,m}.W_{m,i}}$ and checks $\boldsymbol{\beta_{i,m}}$. If the check on $\boldsymbol{\beta_{i,m}}$ does not hold, $\mathbb{C}$ terminates the session. Otherwise, a session key $\boldsymbol{K_{i,j} = a_{i,m}.K_{m,i}}$ is established, and the session is terminated.

Game$_2$: The collision probabilities of the hash oracle and random oracle queries are considered in this game for all transcripts communicated between P_i and S_m. Based on the birthday paradox, we can obtain the highest probability of hash queries as $\frac{q_{hash}{}^2}{2^{L_{hash}+1}}$. In the login and authentication phase, there are three random numbers $a_{i,m}, s_{i,m}, b_{m,i}$ generated by P_i and S_m to construct two messages $\{C_{i,m1}, C_{i,m2}, T_{i,m}\}$ and $\{\theta_{m,i}\}$. Its collision probability is at most $\frac{(q_{send}+q_{execute})^3}{2^{L_{number}+1}}$. As G_1 and G_2 are indistinguishable, we have

$$|Pr[S_2] - Pr[S_1]| \leq \frac{(q_{send} + q_{execute})^3}{2^{L_{number}+1}} + \frac{q_{hash}{}^2}{2^{L_{hash}+1}} \tag{4}$$

Game$_3$: This game is similar to the previous game, but the queries are executed for each specific transcript. G_3 consists of two cases consistent with two transcripts sent by P_i and S_m.

+ *Case 1*: The query *Send*(S_m, M_1) is considered in this case. The messages C_1 are computed from four values $C_{i,m1}, c_{i,m21}, c_{i,m22}, c_{i,m23}$, which result in a probability of $4\frac{q_{hash}}{2^{L_{hash}}}$ in total. Note that I do not consider $T_{i,m}$ in M_1 for the hash oracle, as the timestamp is not difficult to retrieve or generate. On the other hand, the random numbers $a_{i,m}, s_{i,m}$ contained in M_1 have a probability of $2\frac{q_{send}}{2^{L_{number}}}$.

+ *Case 2*: I consider the query $Send(P_i, M_2)$ in this case. Suppose the values $W_{m,i}, K_{m,i}$ and the hash $\beta_{m,i}$ contained in messages M_2 are divulged to $\mathcal{A}$ in order to perform the attacks. To this end, the maximum probability is up to $3\frac{q_{hash}}{2^{L_{hash}}}$. The random number $b_{m,i}$ has a probability of $\frac{q_{send}}{2^{L_{number}}}$.

Overall, this results in the following total probability:

$$|Pr[S_3] - Pr[S_2]| \leq 7\frac{q_{hash}}{2^{L_{hash}}} + 3\frac{q_{send}}{2^{L_{number}}} \tag{5}$$

Game$_4$: I consider the guessing attacks executed by $\mathcal{A}$ in this game. Four cases are presented as follows.

+ *Case 1*: $\mathcal{A}$ executes the query $Corrupt(P_i, x = 1)$ to guess the password of P_i. Next, $\mathcal{A}$ executes the query $Send(S_m, M_1)$ for the attacks. In this case, the highest probability is $(C'.q_{send}^{s'})$.

+ *Case 2*: $\mathcal{A}$ creates the query $Corrupt(P_i, x = 2)$ to retrieve the biometrics of P_i. Since $\mathcal{A}$ also creates the query $Send(S_m, M_1)$ in this case, the simulated probability is at most $max\left\{q_{send}\left(\frac{1}{2^{L_{biometrics}}}, \varepsilon_{biometrics}\right)\right\}$.

+ *Case 3*: $\mathcal{A}$ attempts to break the ECDLP assumption (using the *Hash* oracle queries) to compromise the numbers $a_{i,m}, s_{i,m}, b_{m,i}$ based on the values $R_{i,m}, C_{i,m1}, Y_{m,i}$, respectively. Its maximum collision probability is up to $3q_{hash}Adv_{\mathcal{A}}^{ECDLP}(t_{\mathcal{A}})$.

+ *Case 4*: $\mathcal{A}$ attempts to break the ECCDHP assumption (using the *Hash* oracle queries) to directly compromise the key $K_{i,j} = (a_{i,m}.b_{m,i}).(a_{j,n}.b_{n,j}).G$ given the received values $Y_{m,i} = (a_{i,m}.b_{m,i}).G$ and $(a_{j,n}.b_{n,j}).G$. The maximum collision probability is up to $q_{hash}Adv_{\mathcal{A}}^{ECCDHP}(t_{\mathcal{A}})$.

Since G_3 and G_4 are identical without the above attacks, we obtain

$$\begin{aligned} |Pr[S_4] - Pr[S_3]| \leq max\left\{\left(C'.q_{send}^{s'}\right), q_{send}\left(\frac{1}{2^{L_{biometrics}}}, \varepsilon_{biometrics}\right)\right\} \\ +3q_{hash}Adv_{\mathcal{A}}^{ECDLP}(t_{\mathcal{A}}) + q_{hash}Adv_{\mathcal{A}}^{ECCDHP}(t_{\mathcal{A}}) \end{aligned} \tag{6}$$

Game$_5$: A forward secrecy attack scenario is simulated in this final game. $\mathcal{A}$ creates the *Execute*, *Send* and *Hash* oracle queries to retrieve the session keys from the old transcripts sent by P_i and S_m. The game is simulated with the advantage in breaking the ECDLP assumption and the ECCDHP assumption. To this end, the *Test* query is created to return the session key to $\mathcal{A}$. Since $\mathcal{A}$ has to break the ECDLP three times in a row or break the ECCDHP one time, we have

$$\begin{aligned} |Pr[S_5] - Pr[S_4]| \leq 3q_{hash}(q_{send} + q_{execute})Adv_{\mathcal{A}}^{ECDLP}(t_{\mathcal{A}}) \\ +q_{hash}(q_{send} + q_{execute})Adv_{\mathcal{A}}^{ECCDHP}(t_{\mathcal{A}}) \end{aligned} \tag{7}$$

After executing all the games, $\mathcal{A}$ guesses the bit b' with the probability of the *Test* query as follows.

$$Pr[S_5] = \frac{1}{2} \tag{8}$$

According to Equations (3)–(8), and applying the triangular inequality, we obtain

$$\begin{aligned} |Pr[S_0] - \tfrac{1}{2}| &= |Pr[S_1] - Pr[S_5]| \\ &\leq |Pr[S_1] - Pr[S_2]| \\ &+|Pr[S_2] - Pr[S_3]| \\ &+|Pr[S_3] - Pr[S_4]| \\ &+|Pr[S_4] - Pr[S_5]| \end{aligned} \tag{9}$$

Based on Equations (2)–(9), we can achieve the following equation:

$$\begin{gathered}\tfrac{1}{2}Adv_{\mathbb{C}}^{DNAHC}=\left|Pr[S_0]-\tfrac{1}{2}\right| \\ \leq \frac{(q_{send}+q_{execute})^3}{2^{L_{number}+1}}+\frac{{q_{hash}}^2}{2^{L_{hash}+1}} \\ +7\frac{q_{hash}}{2^{L_{hash}}}+3\frac{q_{send}}{2^{L_{number}}} \\ +max\left\{\left(C'.q_{send}^{s'}\right),q_{send}\left(\frac{1}{2^{L_{biometrics}}},\varepsilon_{biometrics}\right)\right\} \\ +3q_{hash}(q_{send}+q_{execute}+1)Adv_{\mathcal{A}}^{ECDLP}(t_{\mathcal{A}}) \\ +q_{hash}(q_{send}+q_{execute}+1)Adv_{\mathcal{A}}^{ECCDHP}(t_{\mathcal{A}})\end{gathered} \tag{10}$$

The final result can be easily achieved as follows:

$$\begin{gathered}Adv_{\mathbb{C}}^{DNAHC}\leq\frac{(q_{send}+q_{execute})^3+6q_{send}}{2^{L_{number}}}+\frac{{q_{hash}}^2+14q_{hash}}{2^{L_{hash}}} \\ +2max\left\{\left(C'.q_{send}^{s'}\right),q_{send}\left(\frac{1}{2^{L_{biometrics}}},\varepsilon_{biometrics}\right)\right\} \\ +6q_{hash}(q_{send}+q_{execute}+1)Adv_{\mathcal{A}}^{ECDLP}(t_{\mathcal{A}}) \\ +2q_{hash}(q_{send}+q_{execute}+1)Adv_{\mathcal{A}}^{ECCDHP}(t_{\mathcal{A}})\end{gathered} \tag{11}$$

Therefore, I claim Theorem 1. The proposed protocol is proven to be semantically secure, since the above probability is completely negligible. □

6.2. Authentication Proof Using BAN Logic

BAN logic is a well-known tool, which provides a mutual authentication proof of cryptographic protocols [27]. Based on the rules and analytical logic defined in the tool, I aim to prove that P_i and P_j believe the key $K_{i,j}$ computed as a shared secret known only to them. Some notations I use for the proof are described as follows.

- $A \mid\equiv X$: A believes statement M.
- $A \lhd M$: A sees statement M.
- $\#(M)$: Formula M is fresh.
- $A \mid\sim M$: A once said statement M.
- (M, N): M or N is one part of formula (M, N).
- $A \Longrightarrow M$: A has jurisdiction over statement M.
- $\langle M\rangle_N$: This represents M combined with formula N.
- $A \overset{K}{\leftrightarrow} B$: Value K is known only to A and B, and it is used for their communication.
- $A \overset{M}{\Leftrightarrow} B$: Formula M is a secret known only by A and B. Only A and B can use M to authenticate each other.

Based on the principle of BAN logic and the procedure of the proposed protocol, the following six authentication goals should be satisfied.

***Goal* 1:** $S_m \mid\equiv \left(S_m \overset{R_{i,m}}{\leftrightarrow} P_i\right)$. S_m believes $R_{i,m}$ is a secret value sent by P_i, and $R_{i,m}$ is a secret key shared by S_m and P_i. **(G1)**

***Goal* 2:** $P_i \mid\equiv \left(P_i \overset{R_{i,m}}{\leftrightarrow} S_m\right)$. P_i believes $R_{i,m}$ is a secret key shared by P_i and S_m. **(G2)**

***Goal* 3:** $S_n \mid\equiv \left(S_n \overset{R_{j,n}}{\leftrightarrow} P_j\right)$. S_n believes $R_{j,n}$ is a secret value sent by P_j, and $R_{j,n}$ is a secret key shared by S_n and P_j. **(G3)**

***Goal* 4:** $P_j \mid\equiv \left(P_j \overset{R_{j,n}}{\leftrightarrow} S_n\right)$. P_j believes $R_{j,n}$ is a secret key shared by P_j and S_n. **(G4)**

***Goal* 5:** $P_i \mid\equiv \left(P_i \overset{K_{i,j}}{\leftrightarrow} P_j\right)$. P_i believes $K_{i,j}$ is a secret value sent by P_j, and $K_{i,j}$ is a secret key shared by P_i and P_j. **(G5)**

***Goal* 6:** $P_j \mid\equiv \left(P_j \overset{K_{i,j}}{\leftrightarrow} P_i\right)$. P_j believes $K_{i,j}$ is a secret value sent by P_i, and $K_{i,j}$ is a secret key shared by P_j and P_i. (**G6**)

I consider four messages communicated in the login and authentication phase for the analysis, described as follows.

Message 1. $P_i \to S_m : (a_{i,m}.G, y_{i,m1}.(u_{i,m}||ID_i) \oplus T_{i,m} mod\ p, y_{i,m2}.r_{i,m1} mod\ p, y_{i,m2}.r_{i,m2} mod\ p, T_{i,m})$.
Message 2. $S_m \to P_i : (b_{m,i}.G, b_{m,i}.Y_{n,j}, h(Y_{m,i}||K_{m,i}||ID_i||ID_{S_m}))_{R_{i,m}}$.
Message 3. $P_j \to S_n : (a_{j,n}.G, y_{j,n1}.(u_{j,n}||ID_j) \oplus T_{j,n} mod\ p, y_{j,n2}.r_{j,n1} mod\ p, y_{j,n2}.r_{j,n2} mod\ p, T_{j,n})$.
Message 4. $S_n \to P_j : (b_{n,j}.G, b_{n,j}.Y_{m,i}, h(Y_{n,j}\big|\big|K_{n,j}\big|\big|ID_j||ID_{S_n}))_{R_{j,n}}$.

The idealized form of these messages used in the BAN logic is given below.
Message 1. $P_i \to S_m : (\langle C_{i,m1}, c_{i,m21}, s_{i,m}, a_{i,m}\rangle_{X_{i,m}}, T_{i,m})$.
Message 2. $S_m \to P_i : \langle b_{m,i}.G, b_{m,i}.Y_{n,j}, h(Y_{m,i}, K_{m,i}, ID_i, ID_{S_m})\rangle_{X_{i,m}}$.
Message 3. $P_j \to S_n : (\langle C_{j,n1}, c_{j,n21}, s_{j,n}, a_{j,n}\rangle_{X_{j,n}}, T_{j,n})$.
Message 4. $S_n \to P_j : \langle b_{n,j}.G, b_{n,j}.Y_{m,i}, h\big(Y_{n,j}, K_{n,j}, ID_j, ID_{S_n}\big)\rangle_{X_{j,n}}$.

Some logical rules provided by the tool are specified as follows.

- $\frac{A \overset{K}{\Leftrightarrow} B, B \lhd \langle M\rangle_K}{A|\equiv B|\sim M}$: Seeing rule (R1);
- $\frac{A|\equiv B|\sim(M,N)}{A|\equiv B|\sim M}$: Interpretation rule (R2);
- $\frac{A|\equiv \#(M)}{A|\equiv \#(M,N)}$: Freshness rule (R3);
- $\frac{A|\equiv \#(M), A|\equiv B|\sim M}{A|\equiv B|\equiv M}$: Verification rule (R4);
- $\frac{A|\equiv B \Rightarrow M, A|\equiv B|\equiv M}{A|\equiv M}$: Jurisdiction rule (R5);
- $\frac{A|\equiv(M,N)}{A|\equiv M}$: Additional rule (R6).

Based on the idealized form, the following assumptions are made for the proof of the proposed protocol.

- $S_m \mid\equiv P_i \overset{X_{i,m}}{\Leftrightarrow} S_m$: Assumption 1 (A1);
- $S_m \mid\equiv \#(T_{i,m})$: Assumption 2 (A2);
- $S_m \Rightarrow (prk_m)$: Assumption 3 (A3);
- $P_i \mid\equiv \#\big(Y_{n,j}\big)$: Assumption 4 (A4);
- $S_m \Rightarrow (b_{i,m})$: Assumption 5 (A5);
- $P_i \Rightarrow (a_{i,m})$: Assumption 6 (A6).

Based on the above rules, assumptions and protocol procedures, a mutual authentication proof of my work is performed in the following steps.

- S_1: According to Message 1, we have $S_m \lhd (\langle C_{i,m1}, c_{i,m21}, s_{i,m}, a_{i,m}\rangle_{X_{i,m}}, T_{i,m})$.
- S_2: Based on R1 and A1, we obtain $S_m \mid\equiv P_i \mid\sim (C_{i,m1}, c_{i,m21}, s_{i,m}, a_{i,m}, T_{i,m})$.
- S_3: Based on R2, we obtain $S_m \mid\equiv P_i \mid\sim (s_{i,m}, a_{i,m}, T_{i,m})$.
- S_4: According to R3 and A2, we obtain $S_m \mid\equiv \#(s_{i,m}, a_{i,m})$.
- S_5: Based on R4, we obtain $S_m \mid\equiv P_i \mid\equiv (s_{i,m}, a_{i,m})$.
- S_6: According to R5 and S_5, we have $S_m \mid\equiv (s_{i,m}, a_{i,m})$.
- S_7: Based on R6, we obtain $S_m \mid\equiv s_{i,m}$, and $S_m \mid\equiv a_{i,m}$.
- S_8: According to A3 and $R_{i,m} = (s_{i,m}.y.a_{i,m}.x\ mod\ p).\Big(prk_m.s_{i,m}.y)^{-1} mod\ p,\ (s_{i,m}.a_{i,m}.y^2\ mod\ p).(prk_m.s_{i,m}.y)^{-1} mod\ p\Big)$, we obtain $S_m \mid\equiv \left(S_m \overset{R_{i,m}}{\leftrightarrow} P_i\right)$ (**G1** achieved).
- S_9: According to Message 2, we have $P_i \lhd \Big(\langle b_{m,i}.G, b_{m,i}.Y_{n,j}, \beta_{m,i}\rangle_{X_{i,m}}\Big)$.
- S_{10}: Based on R1 and A1, we obtain $P_i \mid\equiv S_m \mid\sim \big(b_{m,i}.G, b_{m,i}.Y_{n,j}, \beta_{m,i}\big)$.
- S_{11}: Based on R2, we obtain $P_i |\equiv\ S_m \mid\ \sim \big(b_{m,i}.Y_{n,j}, \beta_{m,i}\big)$.
- S_{12}: Using R3, A4 and A5, we obtain $P_i \mid\ \equiv\ \#(\beta_{m,i})$.

- S_{13}: Based on A6, S_{12} and the rule of the protocol, we obtain $P_i \mid\equiv \left(P_i \overset{R_{i,m}}{\leftrightarrow} S_m\right)$ (**G2** achieved).
- S_{14}: Using similar arguments of S_8 and S_{13} for Message 3 and Message 4, we can obtain $S_n \mid\equiv \left(S_n \overset{R_{j,n}}{\leftrightarrow} P_j\right)$ (**G3** achieved) and $P_j \mid\equiv \left(P_j \overset{R_{j,n}}{\leftrightarrow} S_n\right)$ (**G4** achieved), respectively.
- S_{15}: Based on A4, A5, A6, S_{11} and $K_{i,j} = a_{i,m}.b_{m,i}.Y_{n,j}$, we obtain $P_i \mid\equiv \left(P_i \overset{K_{i,j}}{\leftrightarrow} P_j\right)$ (**G5** achieved).
- S_{16}: Using similar arguments of S_{15}, we can obtain $P_j \mid\equiv \left(P_j \overset{K_{i,j}}{\leftrightarrow} P_i\right)$ (**G6** achieved).

Therefore, the proposed protocol achieves **G1**, **G2**, **G3**, **G4**, **G5** and **G6**. Hence, it can assure that both P_i and P_j mutually authenticate each other.

6.3. Informal Security Analysis

In this subsection, I further discuss the various security features of the proposed protocol and explain its resistance to multiple well-known attacks. The analysis primarily involves the communication between P_i and S_m. Similar arguments can be used to analyze the communication between P_j and S_n, thereby assuring P_i and P_j securely share an E2E common key. The details are as follows.

User anonymity, user untraceability and message unlinkability: The identity ID_i of P_i is masked in $c_{i,m21}$ of the message sent by P_i. The message conveyed by S_m also does not make the ID_i publicly visible. Therefore, ID_i cannot be revealed to $\mathcal{A}$ during transmission of the messages. Each value contained in message $\{C_{i,m1}, C_{i,m2}, T_{i,m}\}$, message $\{\delta_m, Cert_m\}$ and message $\{\theta_{m,i}\}$ of each session is completely not identical, since they are computed using different random numbers and timestamps. Therefore, it is not possible for $\mathcal{A}$ to identify any two transcripts conveyed by a single P_i. In addition, there are no constants found when linking each value of $\{C_{i,m1}, C_{i,m2}, T_{i,m}, \delta_m, Cert_m, \theta_{m,i}\}$ with each other for the purpose of tracing. Thus, the proposed protocol simultaneously achieves user anonymity, user untraceability and message unlinkability.

Robust mutual authentication: Based on the login request message $\{C_{i,m1}, C_{i,m2}, T_{i,m}\}$ from P_i, S_m computes $Z_{m,i}$ using its private key prk_m in order to retrieve $u_{i,m}, ID_i$. S_m verifies the legitimacy of P_i by checking whether $u_{m,i} = h(ID_i||prk_m)$. On the other hand, upon receiving the message, P_i decrypts $\theta_{m,i}$ using $R_{i,m}$. P_i checks the legitimacy of S_m by confirming whether $\beta_{i,m} = h(Y_{i,m}||K_{m,i}||ID_i||ID_{S_m})$. Value $Y_{n,j}$ sent from S_n is also reliable upon successful checks on $\delta_n, Cert_n$. If one of the above checks do not hold, the communication will be terminated; otherwise, it allows P_i to compute the E2E key $K_{i,j}$. Furthermore, in Section 6.2, a mutual authentication proof of communication between P_i and P_j is provided. Thus, my protocol achieves robust mutual authentication.

Perfect forward secrecy: Suppose $\mathcal{A}$ somehow obtained secret values, random numbers or even a session key communicated in the current session. $\mathcal{A}$ intends to use these values to attack past communications. Since the values are completely different in each communication session, it is not possible for $\mathcal{A}$ to carry out these attacks. For example, $\mathcal{A}$ is not able to use the current key $K^*_{i,j} = a^*_{i,m}.a^*_{j,n}.b^*_{m,i}.b^*_{n,j}.G$ to decrypt a ciphertext encrypted using a past key $K_{i,j} = a_{i,m}.a_{j,n}.b_{m,i}.b_{n,j}.G$. Therefore, the conclusion is established.

E2E keysecurity: If S_m acts as $\mathcal{A}$ and attempts to attack the shared key of P_i and P_j, $\mathcal{A}$ needs to know the number $a_{i,m}$ randomly selected by P_i used to compute $K_{i,j} = a_{i,m}.K_{m,i}$. Due to the ECDLP, it is not possible to retrieve $a_{i,m}$ from the given $R_{i,m}$, where $R_{i,m} = a_{i,m}.G$. In addition, $\mathcal{A}$ will not compute the key $K_{i,j} = (a_{i,m}.b_{m,i}).(a_{j,n}.b_{n,j}).G$ successfully given the values $Y_{m,i} = (a_{i,m}.b_{m,i}).G$ and $(a_{j,n}.b_{n,j}).G$ unless $\mathcal{A}$ is able to break the ECCDHP. Thus, the security of the E2E key is assured.

Resistance to DoS attacks: Defending against a DoS attack is one of the toughest tasks in cyber security, since its attack mechanism is mostly based on computer or network resources. In this analysis, I discuss the resistance of the protocol to possible risks of a

DoS attack, which may affect communication performance. At first, the card SC_i always checks the legitimacy of P_i based upon V_i and their input credentials ID_i^*, PW_i^*, B_i^*. If the verification does not hold, the system will immediately terminate the session. Therefore, $\mathcal{A}$ is not able to flood the communication with subsequent steps. S_m also identifies P_i upon ID_i, as well as verifying the freshness of $u_{m,i}$ through some ECC-based lightweight computation steps. Repeatedly retransmitting $\{C_{i,m1}, C_{i,m2}, T_{i,m}\}$ to disrupt S_m's services would not work efficiently considering the redundant resources on the server side. Moreover, the communication will be stopped if the check on $u_{m,i}$ fails. Hence, the conclusion is established.

Resistance to MITM attacks: In intercepting the login message, $\mathcal{A}$ may use its own parameters to forge and generate a candidate message. The purpose is to act as a middle man to compromise the conveyed messages between P_i and S_m without being noticed. However, since $\mathcal{A}$ does not know the key prk_m and the identity ID_i, it is not possible for $\mathcal{A}$ to compute a correct $u_{i,m}$ for verification and a correct $Z_{m,i}$ for generating $R_{i,m}$. Without $R_{i,m}$, $\mathcal{A}$ is also not able to create a tampered message $\{\theta_{m,i}\}$ sent to P_i. Thus, my protocol is robust against MITM attacks.

Resistance to replay attacks: Suppose $\mathcal{A}$ intercepts and resends the message $\{C_{i,m1}, C_{i,m2}, T_{i,m}\}$ to S_m in order to perform a replay attack on the subsequent communication session. In my protocol, timestamp $T_{i,m}$, which can only be used once, is employed to check whether the message is resent. Moreover, even if $\mathcal{A}$ can somehow pass the timestamp challenge, the replay attack will also fail, since $\mathcal{A}$ does not know $R_{i,m}$ and $a_{i,m}$ to decrypt $\theta_{m,i}$ and compromise $K_{i,j}$. Therefore, the conclusion is established.

Resistance to online and offline password guessing attacks: In the online login interface, $\mathcal{A}$ may enter a guessed password (even with a correct identity and correct biometrics) into the system. Based on the rule of my protocol, SC_i will check the value V_i and easily decline the candidate password entered by $\mathcal{A}$. Suppose $\mathcal{A}$ somehow obtains values $A_{i,m}$ and V_i and then $\mathcal{A}$ attempts to guess P_i's password based on these hash values. However, other than PW_i, $A_{i,m}$ and V_i contain ID_i, B_i, σ_i. Therefore, it is not possible for $\mathcal{A}$ to compute the candidate hashes $A'_{i,m}, V'_i$ and compare them with $A_{i,m}, V_i$ to guess the correct password. Thus, the conclusion is established. Along with the password, biometrics is also fully protected in my protocol during the communication process, which guarantees a strong three-factor authentication mechanism. Moreover, the password and biometrics update function is provided in the proposed protocol, which further enhances the security of PW_i and B_i.

Resistance to stolen smart card attacks: Suppose the card SC_i of P_i is somehow lost and $\mathcal{A}$ has obtained it; $\mathcal{A}$ conducts a power analysis [34] and retrieves all the parameters stored in SC_i. Nevertheless, the password PW_i and biometrics B_i are not directly stored in SC_i; they are therefore not exposed to $\mathcal{A}$ upon power analysis. Even if $\mathcal{A}$ can simultaneously obtain MD_i and SC_i, $\mathcal{A}$ is not able to pass the smart card verification without ID_i, PW_i, B_i when entering the credentials to the login system. With the obtained MD_i and SC_i, it is also not possible for $\mathcal{A}$ to spoof S_m with $u_{m,i} = B_{m,i} \oplus A_{i,m}$, as $\mathcal{A}$ does not know the ID_i for the check $u_{m,i} \overset{?}{=} h(ID_i||prk_m)$. If $\mathcal{A}$ uses V_i obtained from SC_i for the verification, $\mathcal{A}$ can also not compute a valid $C_{i,m21}$ for the login request without ID_i. Hence, my proposed protocol can fully prevent lost smart card attacks.

Resistance to impersonation attacks: Suppose $\mathcal{A}$ has obtained the identity ID_i and then uses it to compute a candidate login request for the purpose of impersonating P_i. Due to the stated resistance to online and offline password guessing attacks, PW_i will not be revealed to $\mathcal{A}$. Moreover, B_i is completely protected and possessed by P_i only; MD_i and SC_i are also carefully preserved to prevent them from being retrieved $u_{m,i}$. Therefore, upon the obtained ID_i, it is still not possible for $\mathcal{A}$ to compute a correct login message $\{C_{i,m1}, C_{i,m2}, T_{i,m}\}$. Thus, the proposed protocol can withstand impersonation attacks.

Resistance to insider attacks: Each server S_m is accepted as trustworthy during the registration procedure because P_i registers their secret information to gain services from S_m. No sensitive values are stored in S_m's database after registration. Moreover, my protocol is also designed without unmasked biometrics database or plaintext password table required. Hence, the protocol can resist insider attacks.

Resistance to desynchronization attacks: During the communication process, two acknowledgements $u_{m,i}$ and $\beta_{i,m}$ are generated for the verifications, which prevent user impersonation and server impersonation. These values will be deleted after the communication sessions are completed. P_i and S_m do not further store any redundant values after each authentication procedure finishes. Therefore, the proposed work completely withstands desynchronization attacks.

7. Performance Evaluation

In this section, a detailed comparative study of the proposed protocol and several related protocols (which are most similar to mine) discussed in Section 2 is presented. Various aspects, including functionality, communication cost and computation cost, are considered in the performance comparison.

7.1. Functionality

The results of a comparison of various functionalities achieved by the protocols are tabulated in Table 3. The √ symbol signifies that the protocol achieves a specific functionality. The × symbol signifies that the function is not achieved by the protocols. The – symbol means that a specific functionality is not available in the protocol. It is observed that my proposed protocol provides the support for more functionalities and security properties compared with the related works. Only my protocol includes a IoLT-based U-healthcare application and a cross-server E2E communication in the design. The proposed work is also the only one to provide user biometrics update for a three-factor authentication solution with the cost-saving biohash function employed.

Table 3. Comparison of functionalities and security properties.

Functionalities	[11]	[12]	[13]	[17]	[19]	[20]	[22]	[24]	[25]	[26]	[27]	**Mine**
Provision of IoLT-based U-healthcare application	×	×	×	×	×	×	×	×	×	×	×	√
Provision of E2E communication	√	√	√	×	×	×	×	×	×	×	×	√
Provision of cross-server communication	×	×	×	×	×	×	×	×	×	×	×	√
Provision of three-factor authentication	×	×	×	×	√	×	√	√	×	×	×	√
Provision of centerless authentication	√	√	√	√	√	√	√	√	√	√	√	√
Provision of SC-SSO solution	×	×	×	×	√	×	√	×	√	×	×	√
Provision of user anonymity	–	√	√	√	√	√	√	√	√	√	√	√
Provision of user untraceability	–	√	√	√	√	√	√	√	×	×	√	√
Provision of message unlinkability	–	√	√	√	√	√	√	√	×	×	√	√
Provision of robust mutual authentication	–	√	√	√	√	√	√	√	√	√	√	√
Provision of perfect forward secrecy	√	√	√	√	√	√	√	√	√	√	√	√
Provision of user password update	–	–	√	–	√	√	√	√	√	√	√	√
Provision of user biometrics update	–	–	–	–	×	–	×	√	–	–	–	√
Provision of mathematical security proof	×	×	×	×	√	×	√	√	×	×	√	√
Resistance to DoS attacks	√	√	√	√	√	√	√	√	×	√	√	√
Resistance to MITM attacks	√	√	√	√	√	√	√	√	√	√	√	√
Resistance to replay attacks	√	√	√	√	√	√	√	√	√	√	√	√
Resistance to online password guessing attacks	–	–	√	–	√	√	√	√	√	√	√	√
Resistance to offline password guessing attacks	–	–	√	–	√	√	√	√	√	√	√	√
Resistance to stolen smart card attacks	–	–	√	√	√	√	√	√	×	√	√	√
Resistance to impersonation attacks	√	√	√	√	√	√	√	√	√	√	√	√
Resistance to insider attacks	√	√	√	√	√	√	√	√	–	√	√	√
Resistance to desynchronization attacks	√	√	√	√	√	√	√	√	√	×	√	√

7.2. Communication Cost

I use some parameters defined for the communicational evaluation as follows. A length of 1024 bits is assumed to be the size of the asymmetric encryptions or decryptions (e.g., RSA cryptosystem) and the Chebyshev polynomials for assuring strong security. Each block of a symmetric encryption or a symmetric decryption has a length of 256 bits. The size of an identity, a password and a biometrics value is 128 bits. The size of a random number or a hash value is 160 bits. A single elliptic curve point multiplication operation has a length of 320 bits. The size of each timestamp is 32 bits.

The total communication rounds and the length of all transcripts conveyed in each authentication session are considered as the communication cost of the protocols. In the login and authentication phase of my proposed protocol, the transcripts include $(C_{i,m1}, c_{i,m21}, c_{i,m22}, c_{i,m23}, T_{i,m})$ and $(\theta_{m,i})$, which consume a length of (320 bits + 3*160 bits + 32 bits) and 256 bits, respectively. The total length is 1088 bits. In addition, the protocol is executed in two rounds of communication. The costs of the remaining protocols are calculated in a similar way. For a fair comparison, I do not include the communication between S_m and S_n of the proposed protocol in the evaluation. Table 4 and Figure 5 tabulate the detailed comparison of the communication cost of different models. It is observed that my proposed work is one of the most efficient protocols. Only the work of Le et al. [22] is more efficient than mine in this evaluation. However, according to Table 3, my work provides many more functionalities and security properties compared to that of Le et al. [22].

Table 4. Comparison of communication cost.

Protocols	Total Communication Rounds	Total Cost of P_i and S_m (bits)
Le et al. [22]	2	512
Xu et al. [24]	3	1344
Lin et al. [25]	3	5736
Meshram et al. [26]	2	3072
Shohaimay and Ismail [27]	3	1376
Mine	2	1088

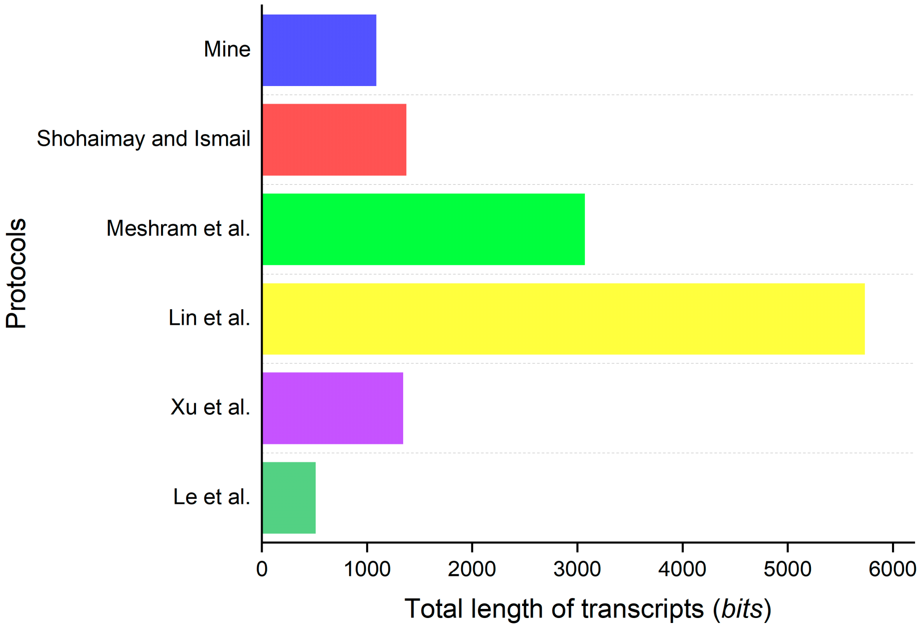

Figure 5. Total length of transcripts of different protocols [22,24–27].

7.3. Computation Cost

The computation cost is calculated by the execution time of all cryptographic operations in each protocol. I consider the time of computing an XOR negligible, as the operation is extremely fast. In addition, the difference between the execution times of a biohash function and a one-way hash function is too small [29]. For simplicity, they are assumed to be similar. I denote the following cryptographic functions and operations for the evaluation in this subsection.

- T_{FE}: Time of running fuzzy extraction function.
- T_{CM}: Time of running a Chebyshev chaotic polynomial mapping.
- T_{PM}: Time of operating an EC point multiplication.
- T_{PA}: Time of operating an EC point addition.
- T_{SED}: Time of running a symmetric encryption or symmetric decryption.
- T_{M}: Time of calculating a modular squaring.
- T_{QR}: Time of calculating a square root module N.
- T_{H}: Time of running a hash function.

The result of the comparison of computational cost of multiple protocols is presented in Table 5 and Figure 6. Based on the result, for each communication session, the proposed protocol is more efficient than Shohaimay and Ismail's [27] protocol. The incurred costs in the works of Le et al. [22], Lin et al. [25] and Meshram et al. [26] are less than the ones in my protocol, Xu et al.'s [24] protocol and Shohaimay and Ismail's [27] protocol. Nevertheless, my protocol provides support for more functional properties and is better than the ones of Le et al. [22], Lin et al. [25] and Meshram et al. [26] in terms of communicational efficiency.

Table 5. Comparison of computation cost.

Protocols	Time Complexities of P_i Side	Time Estimation of P_i Side (ms)	Time Complexities of S_m Side	Time Estimation of S_m Side (ms)	Total Time Estimation (ms)
Le et al. [22]	$T_M + T_{SED} + 9T_H$	≈0.00744	$T_{QR} + 2T_{SED} + 8T_H$	≈1.17560	≈1.18304
Xu et al. [24]	$T_{FE} + 4T_{PM} + 9T_H$	≈2.54621	$3T_{PM} + 5T_H$	≈1.52745	≈4.07366
Lin et al. [25]	$2T_{CM} + 2T_{SED} + 7T_H$	≈0.06353	$2T_{CM} + 2T_{SED} + 5T_H$	≈0.06215	≈0.12568
Meshram et al. [26]	$2T_{CM} + 11T_H$	≈0.06521	$2T_{CM} + 9T_H$	≈0.06383	≈0.12904
Shohaimay and Ismail [27]	$4T_{PM} + 2T_{PA} + 7T_H$	≈2.05063	$4T_{PM} + T_{PA} + 5T_H$	≈2.04235	≈4.09298
Mine	$4T_{PM} + T_{SED} + 4T_H$	≈2.03530	$3T_{PM} + T_{SED} + 2T_H$	≈1.52592	≈3.56122

Based on Refs. [22,23], $T_{FE} \approx 0.508$ ms, $T_{CM} \approx 0.02881$ ms, $T_{PM} \approx 0.508$ ms, $T_{PA} \approx 0.0069$ ms, $T_{SED} \approx 0.00054$ ms, $T_M \approx 0.00069$ ms, $T_{QR} \approx 1.169$ ms and $T_H \approx 0.00069$ ms.

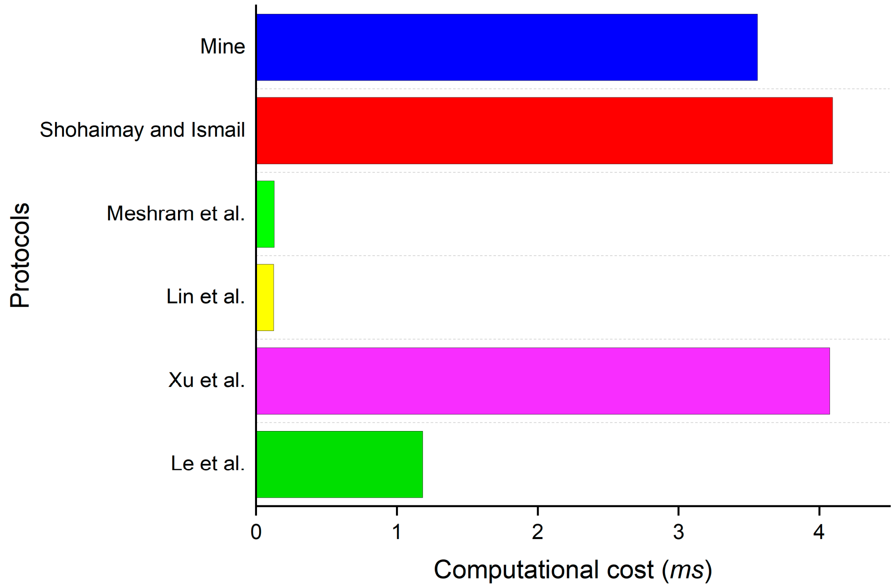

Figure 6. Computational cost of a single communication session in different protocols [22,24–27].

Furthermore, I consider a scenario where a single patient is using multiple U-healthcare services. Since the cost-saving SC-SSO solution is employed in my work, some operations before smart card verification, such as $\sigma_i = \varepsilon_i \oplus h(ID_i^*||PW_i^*||h_{bio}(B_i^*))$ and

$V_i \overset{?}{=} h(h(ID_i^*||PW_i^*||\sigma_i)||h_{bio}(B_i^*))$, are only computed once for communications with multiple servers. The result is depicted in Figure 7. It is indicated that when the number of servers (s) increases, the proposed protocol incurs a more and more rational cost compared to the ones of Xu et al. [24] and Shohaimay and Ismail [27]. As a matter of fact, it incurs an acceptable computational cost considering such superiority over all related protocols in various aspects, which are discussed in Sections 7.1 and 7.2.

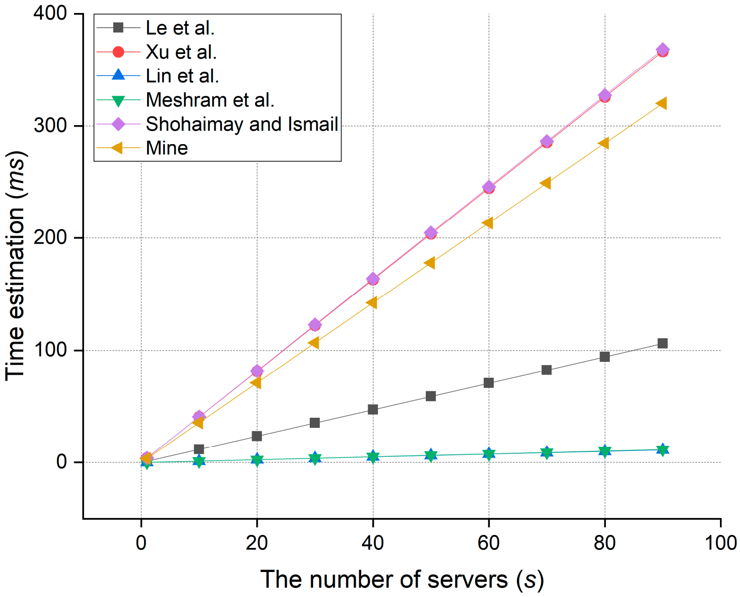

Figure 7. Computation costs of communications between a single patient and multiple servers [22,24–27].

8. Conclusions

In this paper, I proposed a CS-E2E patient authentication protocol for DNA-based U-healthcare services in the IoLT. The proposed protocol allows two patients to mutually authenticate each other and compute a secret shared key with the assistance of respective servers. In this way, patients can securely establish a reliable private channel for E2E healthcare communications. Based on results of the security analysis, my protocol is proven to be free from various attacks; it also provides the support for more security properties and better functionalities. Multiple cost-saving solutions, including SC-SSO, ECC, the biohash function, are employed in the design. A performance evaluation of multiple aspects, including the computational cost and communicational cost, is also presented, which indicates that the protocol incurs reasonable costs compared to related works.

In future works, I intend to design a certificateless-based E2E patient authenticated key exchange scheme for another healthcare security scenario. All credentials stored on the mobile device may be moved to the smart card in order to enable service availability on multiple devices. Here, there is a trade-off consideration between security and functionality, since the attackers only need to compromise the smart card for the attacks without obtaining the device. Furthermore, I would seek solutions, which can further reduce the computational cost and improve the whole communication efficiency of the current proposed approach—for instance, the EC point addition replacing EC point multiplication in some operations.

Funding: This research and its APC was funded by National Science and Technology Council (Taiwan), grant number NSTC-112-2222-E-030-001, and by Fu Jen Catholic University (Taiwan), grant number A0211018-1.

Data Availability Statement: Not available.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Normand, R.; Yanai, I. An introduction to high-throughput sequencing experiments: Design and bioinformatics analysis. In *Deep Sequencing Data Analysis*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 1038, pp. 1–26.
2. Grishin, D.; Obbad, K.; Estep, P.; Cifric, M.; Zhao, Y.; Church, G. *Blockchain-Enabled Genomic Data Sharing and Analysis Platform*; Nebula Genomics: San Francisco, CA, USA, 2018.
3. Raza, K.; Qazi, S. Chapter 5—Nanopore sequencing technology and Internet of living things: A big hope for U-healthcare. In *Sensors for Health Monitoring*; Dey, N., Chaki, J., Kumar, R., Eds.; Academic Press: Cambridge, MA, USA, 2019; pp. 95–116.
4. Pizzolante, R.; Castiglione, A.; Carpentieri, B.; De Santis, A.; Palmieri, F.; Castiglione, A. On the protection of consumer genomic data in the Internet of Living Things. *Comput. Secur.* **2018**, *74*, 384–400. [CrossRef]
5. Bolognini, D.; Bartalucci, N.; Mingrino, A.; Vannucchi, A.M.; Magi, A. NanoR: A user-friendly R package to analyze and compare nanopore sequencing data. *PLoS ONE* **2019**, *14*, e0216471. [CrossRef] [PubMed]
6. Shabani, M. Blockchain-based platforms for genomic data sharing: A de-centralized approach in response to the governance problems? *J. Am. Med. Inform. Assoc.* **2019**, *26*, 76–80. [CrossRef] [PubMed]
7. Hsu, C.; 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]
8. Kumari, A.; Jangirala, S.; Abbasi, M.Y.; Kumar, V.; Alam, M. ESEAP: ECC based secure and efficient mutual authentication protocol using smart card. *J. Inf. Secur. Appl.* **2020**, *51*, 102443. [CrossRef]
9. Fereidooni, H.; Taheri, H.; Mahramian, M. E2E KEEP: End to End Key Exchange and Encryption Protocol for Accelerated Satellite Networks. *Int. J. Commun. Netw. Syst. Sci.* **2012**, *5*, 228–237.
10. Jiang, Q.; Ma, J.; Yang, C.; Ma, X.; Shen, J.; Chaudhry, S.A. Efficient end-to-end authentication protocol for wearable health monitoring systems. *Comput. Electr. Eng.* **2017**, *63*, 182–195. [CrossRef]
11. Wang, Q.; Huang, X.; Mengistu, D. Session Key Agreement for End-to-End Security in Time-Synchronized Networks. In Proceedings of the 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague, Czech Republic, 3–6 July 2018.
12. Liu, D.; Liu, X.; Zhang, H.; Yu, H.; Wang, W.; Ma, L.; Chen, J.; Li, D. Research on End-to-End Security Authentication Protocol of NB-IoT for Smart Grid Based on Physical Unclonable Function. In Proceedings of the 2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN), Chongqing, China, 14–16 June 2019.
13. Nashwan, S. An End-to-End Authentication Scheme for Healthcare IoT Systems Using WMSN. *Comput. Mater. Contin.* **2021**, *68*, 607–642. [CrossRef]
14. Pérez, S.; Hernández-Ramos, J.L.; Raza, S.; Skarmeta, A. Application Layer Key Establishment for End-to-End Security in IoT. *IEEE Internet Things J.* **2020**, *7*, 2117–2128. [CrossRef]
15. Raj, B.S.S.; Venugopalachar, S. Multi-data Multi-user End to End Encryption for Electronic Health Records Data Security in Cloud. *Wirel. Pers. Commun.* **2022**, *125*, 2413–2441. [CrossRef]
16. Alsaeed, N.; Nadeem, F. Authentication in the Internet of Medical Things: Taxonomy, Review, and Open Issues. *Appl. Sci.* **2022**, *12*, 7487. [CrossRef]
17. Deebak, B.D.; Al-Turjman, F. Smart Mutual Authentication Protocol for Cloud Based Medical Healthcare Systems Using Internet of Medical Things. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 346–360. [CrossRef]
18. Chiou, S.-Y.; Ying, Z.; Liu, J. Improvement of a Privacy Authentication Scheme Based on Cloud for Medical Environment. *J. Med. Syst.* **2016**, *40*, 101. [CrossRef] [PubMed]
19. Hsu, C.L.; Le, T.V.; Hsieh, M.C.; Tsai, K.Y.; Lu, C.F.; Lin, T.W. Three-Factor UCSSO Scheme with Fast Authentication and Privacy Protection for Telecare Medicine Information Systems. *IEEE Access* **2020**, *8*, 196553–196566. [CrossRef]
20. Yuanbing, W.; Wanrong, L.; Bin, L. An Improved Authentication Protocol for Smart Healthcare System Using Wireless Medical Sensor Network. *IEEE Access* **2021**, *9*, 105101–105117. [CrossRef]
21. Farash, M.S.; Turkanović, M.; Kumari, S.; Hölbl, M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Netw.* **2016**, *36*, 152–176. [CrossRef]
22. 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]
23. Shuai, M.; Xiong, L.; Wang, C.; Yu, N. A secure authentication scheme with forward secrecy for industrial internet of things using Rabin cryptosystem. *Comput. Commun.* **2020**, *160*, 215–227. [CrossRef]
24. Xu, D.; Chen, J.; Liu, Q. Provably secure anonymous three-factor authentication scheme for multi-server environments. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 611–627. [CrossRef]
25. Lin, T.-W.; Hsu, C.L.; Le, T.V.; Lu, C.F.; Huang, B.Y. A Smartcard-Based User-Controlled Single Sign-On for Privacy Preservation in 5G-IoT Telemedicine Systems. *Sensors* **2021**, *21*, 2880. [CrossRef]
26. Meshram, C.; Ibrahim, R.W.; Deng, L.; Shende, S.W.; Meshram, S.G.; Barve, S.K. A robust smart card and remote user password-based authentication protocol using extended chaotic maps under smart cities environment. *Soft Comput.* **2021**, *25*, 10037–10051. [CrossRef]

27. Shohaimay, F.; Ismail, E.S. Improved and Provably Secure ECC-Based Two-Factor Remote Authentication Scheme with Session Key Agreement. *Mathematics* **2023**, *11*, 5. [CrossRef]
28. Alliance, S.C. *Smart Cards and Biometrics*; The Smart Card Alliance Physical Access Council: Princeton Junction, NJ, USA, 2011.
29. Zhang, L.; Zhang, Y.; Tang, S.; Luo, H. Privacy Protection for E-Health Systems by Means of Dynamic Authentication and Three-Factor Key Agreement. *IEEE Trans. Ind. Electron.* **2018**, *65*, 2795–2805. [CrossRef]
30. Wong, A.M.-K.; Hsu, C.L.; Le, T.V.; Hsieh, M.C.; Lin, T.W. Three-Factor Fast Authentication Scheme with Time Bound and User Anonymity for Multi-Server E-Health Systems in 5G-Based Wireless Sensor Networks. *Sensors* **2020**, *20*, 2511. [CrossRef]
31. Sowjanya, K.; Dasgupta, M.; Ray, S. Elliptic Curve Cryptography based authentication scheme for Internet of Medical Things. *J. Inf. Secur. Appl.* **2021**, *58*, 102761. [CrossRef]
32. Dworkin, M.J.; Barker, E.B.; Nechvatal, J.R.; Foti, J.; Bassham, L.E.; Roback, E.; Dray, J.F., Jr. *Announcing the Advanced Encryption Standard (AES)*; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2001.
33. Alraih, S.; Shayea, I.; Behjati, M.; Nordin, R.; Abdullah, N.F.; Abu-Samah, A.; Nandi, D. Revolution or Evolution? Technical Requirements and Considerations towards 6G Mobile Communications. *Sensors* **2022**, *22*, 762. [CrossRef]
34. Mangard, S.; Oswald, E.; Popp, T. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
35. Liu, W.; Wang, X.; Peng, W.; Xing, Q. Center-Less Single Sign-On with Privacy-Preserving Remote Biometric-Based ID-MAKA Scheme for Mobile Cloud Computing Services. *IEEE Access* **2019**, *7*, 137770–137783. [CrossRef]
36. Barker, E. *Recommendation for Key Management*; Part 1, Revision 4; NIST Special Publication: Gaithersburg, MD, USA, 2016; pp. 800–857.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.