PLOS ONE

RESEARCH ARTICLE

An Improved and Secure Anonymous Biometric-Based User Authentication with Key Agreement Scheme for the Integrated EPR Information System

Jaewook Jung, Dongwoo Kang, Donghoon Lee, Dongho Won*

Department of Computer Engineering, Sungkyunkwan University, 2066 Seoburo, Suwon, Gyeonggido 16419, Korea

* dhwon@security.re.kr

Citation: Jung J, Kang D, Lee D, Won D (2017) An Improved and Secure Anonymous Biometric-Based User Authentication with Key Agreement Scheme for the Integrated EPR Information System. PLoS ONE 12(1): e0169414. doi:10.1371/journal.pone.0169414

Editor: Muhammad Khurram Khan, King Saud University, SAUDI ARABIA

Received: October 11, 2016

Accepted: December 17, 2016

Published: January 3, 2017

Data Availability Statement: All relevant data are within the paper.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2010-0020210), http://www.nrf.re.kr/.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

Nowadays, many hospitals and medical institutes employ an authentication protocol within electronic patient records (EPR) services in order to provide protected electronic transactions in e-medicine systems. In order to establish efficient and robust health care services, numerous studies have been carried out on authentication protocols. Recently, Li et al. proposed a user authenticated key agreement scheme according to EPR information systems, arguing that their scheme is able to resist various types of attacks and preserve diverse security properties. However, this scheme possesses critical vulnerabilities. First, the scheme cannot prevent off-line password guessing attacks and server spoofing attack, and cannot preserve user identity. Second, there is no password verification process with the failure to identify the correct password at the beginning of the login phase. Third, the mechanism of password change is incompetent, in that it induces inefficient communication in communicating with the server to change a user password. Therefore, we suggest an upgraded version of the user authenticated key agreement scheme that provides enhanced security. Our security and performance analysis shows that compared to other related schemes, our scheme not only improves the security level, but also ensures efficiency.

Introduction

The development of Information and Communication Technology (ICT) with the prevalent use of the mobile Internet, smart devices, social network services, and cloud services has brought remarkable changes to our daily lives. This development has also affected the medical field, which has retained a number of conventional and inefficient methods. Recently, a large number of hospitals providing health care services have instituted EPR systems in order to remotely communicate with patients, and to efficiently process their medical records and disease management [1]. The EPR system allows the sharing of patients' medical histories, such as hospital records, diagnosis records, personal information, treatment records, and research records. Using the EPR system, all patient information is available electronically, on screen, at any

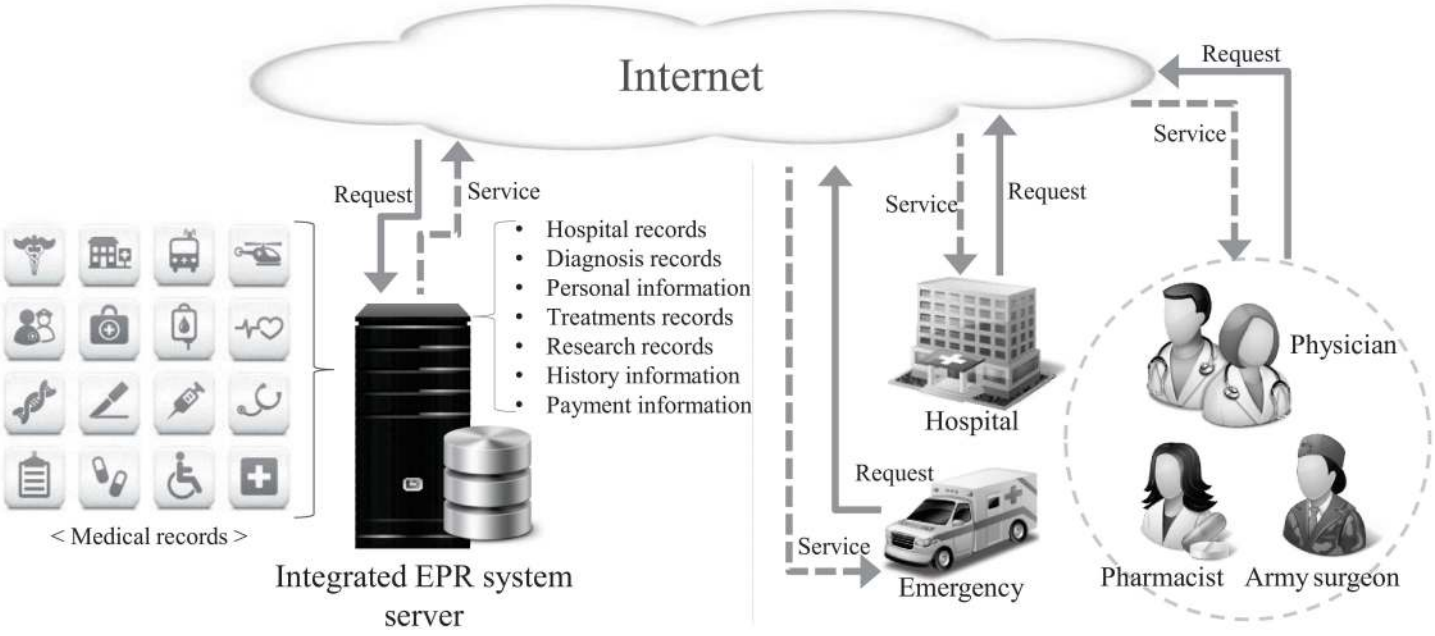

Fig 1. Integrated EPR information system.

doi:10.1371/journal.pone.0169414.g001

hospital location, at any time. In addition, EPR provides the most recent and accurate information, enabling faster diagnoses, treatment plans and discharge processes for the patient [2]. Fig 1 illustrates the integrated EPR information system.

While the users enjoy simplicity and efficiency in EPR information systems, security has emerged as a major issue in both academic and industrial fields [3, 4]. In order to guarantee reliability, authentication protocol provides security and mutual authentication when users access a foreign network.

Lamport [5] first presented an authentication technique based on passwords, and since then, many related studies [6–22] have been conducted to improve security and efficiency for various environments such as wireless environments, Telecare medical information systems (TMIS), health care systems, wireless sensor networks (WSNs), multi-server environments, and mobile pay-TV systems. In 2006, Lee et al. [6] presented a security enhanced authentication mechanism for wireless environments. Wu et al. [7], He et al. [8], Hao et al. [9], Jiang et al. [10] and Moon et al. [11] have proposed authentication method for TMIS. Amin et al. [12] and He et al. [13] proposed user anonymous authentication scheme for health care systems. In 2014, Kim et al. [14], Choi et al. [15] and Nam et al. [16] proposed authenticated key agreement mechanisms for WSN environments. In the same year, Khan [17] presented a fingerprint biometric-based self-authentication and deniable authentication scheme for electronic transaction. In addition, Chaudhry et al. [18], Amin & Biswas [19], Moon et al. [20], and Mishra et al. [21] presented authentication systems for multi-server environments. In 2016, He et al. [22] presented one-to-many authentication using bilinear pairing in mobile pay-TV systems.

Especially, in order to improve the security, various cryptography techniques are used in authentication protocol. Khan et al. [23] and Lee & Hsu [24] apply a chaotic map technique in their scheme. In 2015, Giri et al. [25] presented an RSA-based authentication method for TMIS. However, Amin & Biswas [26] demonstrated that Giri et al.'s scheme [25] cannot guarantee protection against off-line password guessing attacks and insider attacks, and suggested

an improved mechanism based on an RSA cryptosystem. In addition, Chaudhry et al. [27], Irshad et al. [28], Islam & Khan [29], Amin & Biswas [30], and Amin et al. [31, 32] presented authentication mechanisms using elliptic curves cryptography (ECC) for TMIS.

In 2012, Wu et al. [33] first presented an efficient user authentication technique for an integrated EPR information system. They used lightweight operations in their protocol including one-way hash operations and bitwise XOR operations in order to enhance efficiency. However, Lee et al. [34] pointed out that Wu et al. [33] overlooked the possibility of stolen smart card attack through power consumption analysis [35]. Lee et al. [34] then suggested an improved version that addressed the issue of Wu et al.'s [33] technique. However, Wen [36] demonstrated that Lee et al.'s scheme [34] still had some weaknesses, such as off-line password guessing attacks and user impersonation attacks, and proposed an enhanced new strategy. Unfortunately, Li et al. [37] demonstrated that Wen's scheme [36] cannot prevent password disclosure attack nor provide efficient password change. In 2015, Das [38] discovered that Lee et al.'s scheme [34] and Wen's scheme [36] shared the same three vulnerabilities. First, the password change phase of both schemes had no verification process of the user's previous password. Second, their schemes were not protected against insider attack. Third, in their studies, formal security analysis was not conducted. In an attempt to compensate for these defects, Das [38] presented an upgraded scheme. However, Mir et al. [39] discovered that Das's scheme [38] is not protected against off-line/on-line password guessing attacks, and propose a secure anonymous authentication mechanism. Li et al. [40] recently also demonstrated that Das's scheme [38] could not satisfy security requirements because it is not protected against modification attacks and user duplication attacks. They then suggested an enhanced new authentication mechanism.

However, we have discovered that Li et al.'s scheme [40] comprises critical security weaknesses. Their scheme: (i) cannot prevent off-line password guessing attacks and server spoofing attacks, (ii) is unable to preserve user anonymity, (iii) does not identify incorrect passwords promptly in login stage, and (iv) has non-user-friendly password changing procedure, since it requires communication with the server. In this current research, our main contribution is as follows. First, we describe the weaknesses of the above scheme. Second, we propose a more developed authentication mechanism for an integrated EPR information system. Third, we show that the proposed mechanism satisfies the various security requirements. Finally, we demonstrate that the proposed mechanism has good performance in terms of computation cost and time consumption.

The remainder of this paper is structured as follows. Section 2 provides some background information on prior knowledge. In Section 3, we briefly explain Li et al.'s authentication procedure. Section 4 demonstrates the vulnerabilities of Li et al.'s scheme. A detailed explanation of our proposed scheme is provided in Section 5. In Section 6, we evaluate whether our proposed scheme can withstand various attacks while satisfying our claim that the basic requirements of the security scheme are provided. In Section 7, we analyze the performance of the proposed scheme and in Section 8, we provide a conclusion to the paper.

Preliminary Knowledge

In this section, we will describe basic knowledge in terms of security properties and introduce bio-hash function [41], which is used in our proposed scheme.

Security requirements

Multiple security requirements should be considered in order to implement a secure and efficient authentication mechanism. In this subsection, based on previous researches [33, 34, 36,

38, 40, 42–44], we outline some of the important requirements of an authentication scheme. In Section 6, these requirements will be employed in order to scrutinize the security of prior schemes and our proposed scheme.

1. **User anonymity**: In an authentication mechanism, even if an attacker extracts some information stored in a smart card or eavesdrops the exchanged message in the communication group, the user's identity should be preserved.
2. **Mutual authentication**: An authentication mechanism should execute several steps to achieve mutual authentication which is to test all transmitted messages to judging the legitimacies.
3. **Session key agreement**: After the verification process has completed, the user and server should assign the session key to each other.
4. **Password verification process**: If a user erroneously enters an incorrect password in the login phase, the password should be detected before performing the verification phase.
5. **User friendliness**: An authentication mechanism provides a password change procedure with which a user can freely update their password without communicating with the server.
6. **Robustness**: An authenticated key agreement mechanism should be immune to different types of attacks, such as insider attacks, off-line password guessing attacks, replay attacks, and user impersonation attacks.

Bio-hash function

Recently, three-factor authentication mechanism has frequently been used, which complements the two-factor authentication mechanism using *ID* and *PW* by adding biometric information in order to increase security. In a number of studies on the three-factor authentication mechanism [11, 20, 21, 45, 46], the bio-hash function has been applied to the user's biometric information. In 2004, Jin et al. [41] suggested a fingerprint-based function to identify the user's legitimacy. The bio-hash technique employs the particular tokenized pseudo-random numbers to each of users measuring biometric feature arbitrarily onto two fold strands. The bio-hash function $H(\cdot)$ is a one-way function with a feature that can reduce the probability of denial of service. In order to improve security, our proposed scheme adopts the user's biometric information applied in the bio-hash function. The details are as follows in Section 5.

Description of Li et al.'s scheme

In this section, we briefly review Li et al.'s authentication mechanism [40] in order to cryptanalyze their scheme. Their scheme consists of the following phases: registration, login, verification, and password change. Fig 2 describes Li et al.'s scheme, and Table 1 shows the notations employed in the remainder of this paper.

Registration phase

1. U_i inputs his/her ID_i and PW_i, and U_i generates a random secret number X_u that is only retained by user U_i. U_i computes $RPW_i = h(X_u||PW_i)$ and sends a registration request message $\langle ID_i, RPW_i \rangle$ to S_j through a secure channel.
2. S_j verifies the user's ID_i. If it is valid, S_j computes $v = h(K \oplus ID_i \oplus R)$, where the secret number K is chosen by S_j.

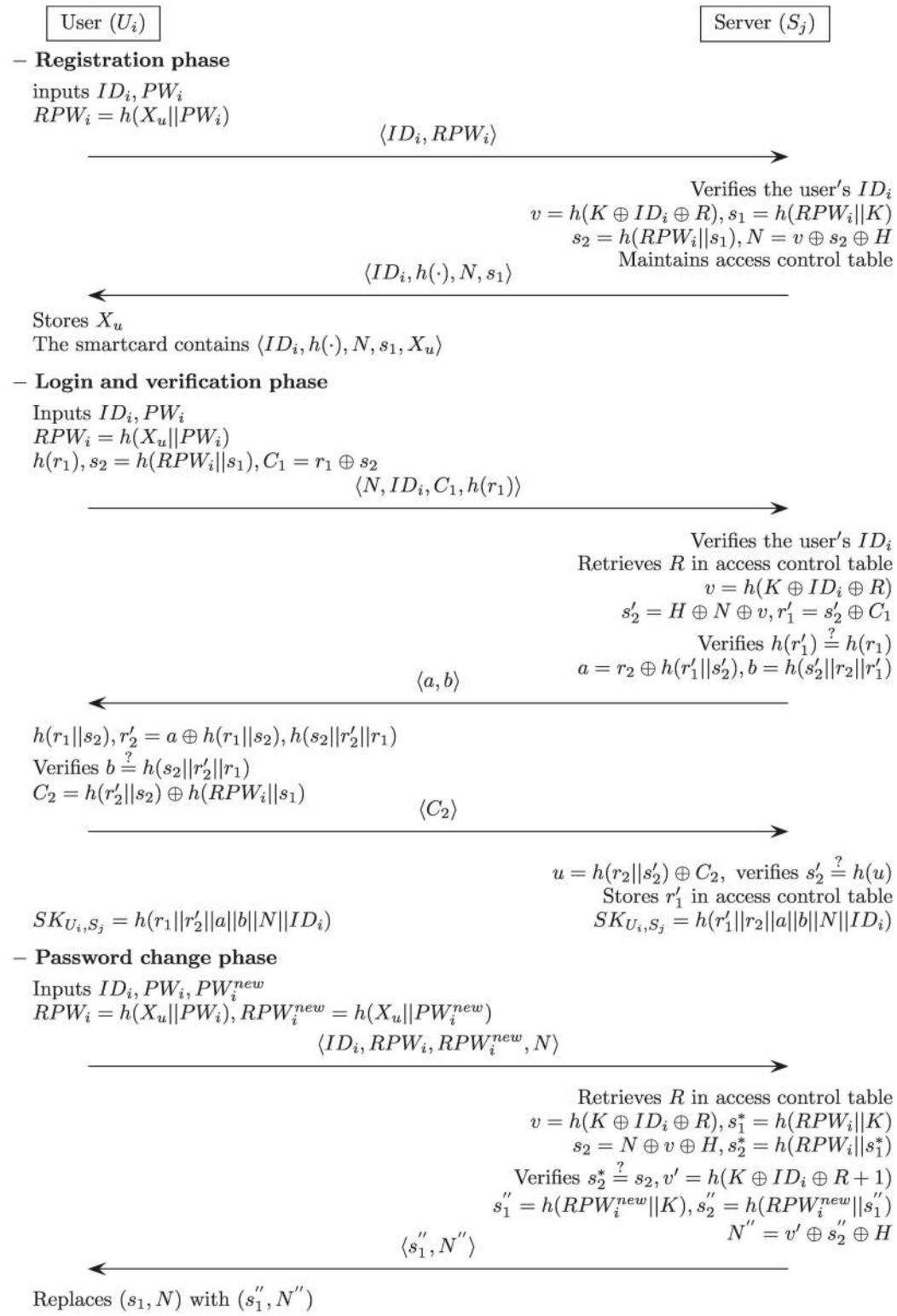

Fig 2. Li et al.'s authentication scheme.

doi:10.1371/journal.pone.0169414.g002

Table 1. Notations.

Notations	Description
U_i	User
S_j	EPR information system server
ID_i	Identity of U_i
PW_i	Password of U_i
PW_i^{new}	New password of U_i
R	Number of times U_i re-registers to S_j
B_i	Biometric information of U_i
X_u	Secret number generated by U_i
H	Secret number generated by S_j
K	Secret key of S_j
r_1	Random number generated by U_i
r_2	Random number generated by S_j
$h(\cdot)$	One-way hash function
$H(\cdot)$	Bio-hash function
$X\|\|Y$	Concatenate operation
$\oplus$	XOR operation

doi:10.1371/journal.pone.0169414.t001

3. S_j computes $s_1 = h(RPW_i||K)$, $s_2 = h(RPW_i||s_1)$ and $N = v \oplus s_2 \oplus H$. S_j then issues a smart card with the parameters $\{ID_i, h(\cdot), N, s_1\}$ and sends it to U_i through a secure channel. At this time, S_j constructs an access control table. This table includes the user's identity ID_i in the first field and in the case of initial registration of U_i, records *null* value and $R = 0$ into the second and third fields, respectively. If it is not an initial registration, S_j records $R = R+1$ in the existing field.
4. Upon receiving the smart card, U_i enters the secret number X_u into its memory, and the smart card includes the information $\{X_u, ID_i, h(\cdot), N, s_1\}$.

Login phase

1. U_i inserts U_i's smart card into a card reader and inputs his/her ID_i and PW_i. The smart card then computes $RPW_i = h(X_u||PW_i)$.
2. The smart card selects a random number r_1 and computes $h(r_1)$, $s_2 = h(RPW_i||s_1)$ and $C_1 = r_1 \oplus s_2$.
3. Finally, U_i sends the login request message $\langle N, ID_i, C_1, h(r_1)\rangle$ to S_j through a public network.

Verification phase

1. S_j verifies the user's ID_i. If it holds, S_j accepts the login request and proceeds with the next step. Otherwise, S_j rejects the login request and this phase is terminated.
2. S_j retrieves the R stored in the access control table and computes $v = h(K \oplus ID_i \oplus R)$, $s_2' = H \oplus N \oplus v$ and $r_1' = s_2' \oplus C_1$. S_j then verifies whether $h(r_1') = h(r_1)$. If this holds, S_j proceeds to the next step. Otherwise, this phase is terminated.
3. S_j selects a random number r_2 and computes $a = r_2 \oplus h(r_1'||s_2')$ and $b = h(s_2'||r_2||r_1')$. S_j then sends an authentication request message $\langle a, b\rangle$ to U_i through a public network.

4. U_i computes $h(r_1||s_2)$, $r'_2 = a \oplus h(r_1||s_2)$ and $h(s_2||r'_2||r_1)$. U_i then verifies whether $b = h(s_2||r'_2||r_1)$. If this holds, U_i successfully authenticates S_j. Subsequently, U_i computes $C_2 = h(r'_2||s_2) \oplus h(RPW_i||s_1)$ and sends the acknowledgement message $\langle C_2 \rangle$ to S_j through a public network.
5. S_j computes $u = h(r_2||s'_2) \oplus C_2$ and verifies whether $s'_2 = h(u)$. If this holds, S_j successfully authenticates the U_i and stores r'_1 in its access control table.
6. Finally, S_j computes a shared session key $SK_{U_i,S_j} = h(r'_1||r_2||a||b||N||ID_i)$ and the U_i also computes the same shared session key $SK_{U_i,S_j} = h(r_1||r'_2||a||b||N||ID_i)$ successfully.

Password change phase

1. U_i inserts U_i's smart card into a card reader and inputs ID_i, old password PW_i and new password PW_i^{new}. The smart card computes the old masked password $RPW_i = h(X_u||PW_i)$ and the new masked password $RPW_i^{new} = h(X_u||PW_i^{new})$. U_i then sends the password change request message $\langle ID_i, RPW_i, RPW_i^{new}, N \rangle$ to S_j through a secure channel.
2. S_j retrieves R stored in the access control table and computes $v = h(K \oplus ID_i \oplus R)$, $s_1^* = h(RPW_i||K)$, $s_2 = N \oplus v \oplus H$, and $s_2^* = h(RPW_i||s_1^*)$. S_j then verifies whether $s_2^* = s_2$. If this holds, S_j accepts the password change request and proceeds with the next step. Otherwise, the password change request is rejected and this phase is terminated.
3. S_j computes $v' = h(K \oplus ID_i \oplus R+1)$, $s''_1 = h(RPW_i^{new}||K)$, $s''_2 = h(RPW_i^{new}||s''_1)$ and $N'' = v' \oplus s''_2 \oplus H$. S_j then sends the password change update message $\langle s''_1, N'' \rangle$ to U_i through a secure channel.
4. Upon receiving the password change update message, the smart card replaces the existing values s_1 and N with the new values s''_1 and N'', respectively. Finally, the smart card contains the information $\{ID_i, h(\cdot), N'', s''_1, X_u\}$.

Security pitfalls of Li et al.'s scheme

In this section, we show that Li et al.'s scheme [40] possesses some security vulnerabilities. The following attacks are based on the assumptions that an attacker can extract all of the parameters stored in the smart card by physically monitoring its power consumption [35] and that the attacker can intercept or modify any messages in the public channel [11, 14, 15]. Under these two assumptions, the following problems have been observed and their detailed descriptions are given below.

Off-line password guessing attack

This attack is an attempt to identify a password until the correct password is found, and is carried out due to the tendency of many users to create simple, brief passwords for the sake of convenience. For this reason, authentication schemes for all password-based users should be designed to prevent a guessing attack; however, Li et al.'s scheme [40] does not provide sufficient protection against such an attack. We therefore propose a scenario for an off-line password-guessing attack using Li et al.'s scheme. The following is a detailed description of this scenario:

Step.1. An attacker extracts $\{X_u, ID_i, h(\cdot), N, s_1\}$ from U_i's lost smart card [35].

Step.2. The attacker collects a valid login request message $\langle N, ID_i, C_1, h(r_1)\rangle$ from the previous session.

Step.3. The attacker selects a password candidate PW_i^*.

Step.4. The attacker computes $h(r_1) = h(C_1 \oplus s_2)$ using $r_1 = C_1 \oplus s_2$. Note that $s_2 = h(RPW_i||s_1)$, and $RPW_i = h(X_u||PW_i)$; therefore, the attacker can write $h(r_1) = h(C_1 \oplus h(h(X_u||PW_i)||s_1))$.

Step.5. The attacker then computes $h(r_1)^* = h(C_1 \oplus h(h(X_u||PW_i^*)||s_1))$ using the password candidate PW_i^*.

Step.6. The attacker repeats the above steps from 3 to 5 until the computed result $h(r_1)^*$ is equal to the breached secret $h(r_1)$.

Step.7. If $h(r_1)^*$ and $h(r_1)$ correspond, PW_i^* would be an accurate password. If not, the attacker repeats the above steps until the correct password is found.

Therefore, Li et al's scheme [40] is vulnerable to the off-line password guessing attack. In addition, if an attacker obtains the user's password, they can successfully launch a user impersonation attack.

Server spoofing attack

The security of the password-based user authentication mechanism is based on the intelligence of the password. Thus, if an attacker gains a password, they can pretend to be a legal server. Unfortunately, Li et al.'s scheme allows an attacker to disguise a legal server if the attacker obtains the user's password PW_i through the guessing attack. The following is a detailed description of this scenario:

Step.1. An attacker extracts $\{X_u, ID_i, h(\cdot), N, s_1\}$ from U_i's lost smart card [35].

Step.2. The attacker collects a valid login request message $\langle N, ID_i, C_1, h(r_1)\rangle$ from the previous session.

Step.3. The attacker obtains the PW_i through the off-line password guessing attack.

Step.4. The attacker computes $RPW_i = h(X_u||PW_i)$, $s_2 = h(RPW_i||s_1)$, and $r_1 = s_2 \oplus C_1$. The attacker then selects a random number r_2^* and computes $a^* = r_2^* \oplus h(r_1||s_2)$ and $b^* = h(s_2||r_2^*||r_1)$.

Step.5. The attacker then sends an authentication request $\langle a^*, b^*\rangle$ to U_i through a public network.

Step.6. After receiving the $\langle a^*, b^*\rangle$, U_i computes $h(r_1||s_2)$, $r_2' = a^* \oplus h(r_1||s_2)$ and $h(s_2||r_2'||r_1)$.

Step.7. U_i verifies whether $b^* = h(s_2||r_2'||r_1)$. Finally, U_i successfully finishes the verification process because b^*, which is created by the attacker, is equal to $h(s_2||r_2'||r_1)$.

Through the aforementioned descriptions, it is demonstrated that the attacker can successfully disguise a legal server in Li et al's scheme [40].

Lack of user's anonymity

In modern networks environments, the leakage of user-related information can expedite an outside attacker to identify every specific user. In such a case, the user's privacy data is at risk of being disclosed to an untrusted third party who disobeys the user's will. Therefore, user

anonymity is highly considered as a essential property for user authentication mechanism. However, in Li et al.'s scheme [40], an attacker can easily obtain the user's identity through monitoring the public channels [11, 14, 15] because the user's ID_i is in plain text form during the login phase. In addition, if the attacker obtains the smart card, the user's identity ID_i can also be easily exposed through physically monitoring the smart card's power consumption [35]. With this information, the attacker can also try to launch various types of attacks, which lead to many malicious scenarios. For this reason, user anonymity cannot be preserved in Li et al.'s authentication scheme.

Absence of password verification process

During the Login phase of Li et al.'s scheme [40], if a user enters his/her identity and password, the smart card does not verify the validity of the password itself. This situation leads to other drawbacks as given below.

Case.1. Assume that the U_i inputs the ID_i and incorrect password PW_i^* during the login phase; the smart card then computes $RPW_i^* = h(X_u||PW_i^*)$, $h(r_1)$, $s_2^* = h(RPW_i^*||s_1)$ and $C_1^* = r_1 \oplus s_2^*$. U_i then sends a login request $\langle N, ID_i, C_1^*, h(r_1)\rangle$ to S_j through a public channel. Upon receiving the login request, S_j first checks the validity of the user's identity ID_i. Then, S_j retrieves the stored R in the access control table and computes $v = h(K \oplus ID_i \oplus R)$, $s_2' = H \oplus N \oplus v$ and $r_1' = s_2' \oplus C_1^*$. S_j then verifies whether $h(r_1') = h(r_1)$. If it holds, S_j accepts the login request. Otherwise, the login request is rejected. However, it is obvious that $h(r_1') \neq h(r_1)$, since s_2' is not equal to s_2^*. Therefore, S_j belatedly realizes that the entered password PW_i^* is an incorrect value and S_j rejects U_i's login request. Consequently, if U_i inputs an incorrect password by mistake, the login and verification phases are continued until they have been checked by S_j, leading to unnecessary costs in communication and computation, If the password can be quickly verified at the beginning of the login phase, this situation would not occur and the unnecessary waste would be avoided.

Case.2. Assume that an attacker steals a user's smart card, and logs in to their system using the user's real ID_i and fake password PW_i^*. The smart card computes a fake login request message $\langle N, ID_i, C_1^*, h(r_1)\rangle$ with a fake password PW_i^*. If the attacker sends a large number of such fake login request messages, S_j will be busy processing these messages, and will meanwhile reject other legitimate users. Therefore, without a password verification process, the system suffer from a clogging attack, which is a type of denial of service (DoS) attack.

Inconvenient password change phase

The ideal authentication scheme based on a password should be designed without restricting the user's ability to alter their password stored in smart cards [11, 15]. Moreover, after carrying out a verification process on an existing password, the self-alteration procedure should be performed within the smart card, without extra communication to server S_j, in order to enhance efficiency. However, since a password verification procedure functions with assistance from S_j in Li et al.'s scheme, it is considered to be inefficient and not user-friendly.

Scalability problem

Li et al.'s scheme [40], in order to strengthen security, it is suggested that the server comprises an access control table to save the information such as the user's identity, latest login number,

and registration times. Accordingly, the server needs to retain each user's access control table. However, the increased amount of user information needing to be retained places more burden on the server, since the number of access control table will increase as the number of users' increases. Moreover, the use of the access control table is inefficient in terms of computation time, since the changed values at each phase need to be recorded in the access control table.

The proposed scheme

In this section, we suggest a refined version of the authentication protocol to offer enhanced security by resolving the vulnerabilities of Li et al.'s [40] scheme. In the proposed scheme, in order to resist an off-line password guessing attack and server spoofing attack, we use biometrics information with Biohashing $H(\cdot)$ [41] to securely conceal the password. We also include a password verification process at the beginning of the login phase to ensure high efficiency and resist a denial of service (DoS) attack. In addition, we remove the access control table to reduce the server load and modify the password change process to provide user-friendly access. Our proposed protocol also consists of four phases: registration, login, verification and password change. Fig 3 describes our proposed scheme, and Table 1 lists the notations employed in the proposed scheme.

Registration phase

1. U_i inputs ID_i and PW_i, and imprints his/her biometrics B_i. Then U_i computes $RPW_i = h(PW_i||H(B_i))$ and sends a registration request $\langle ID_i, RPW_i \rangle$ to S_j through a secure channel.
2. S_j verifies the user's ID_i. If it is valid, S_j computes $N = h(ID_i||RPW_i)$ and $v = N \oplus K$.
3. S_j issues a smart card with the $\{v, h(\cdot), H(\cdot)\}$ and sends it to U_i through a secure channel.
4. Upon receiving the smart card, U_i computes $e = h(ID_i||PW_i||H(B_i))$ and enters it into the smart card's memory. Finally, the smart card includes the information $\{v, h(\cdot), H(\cdot), e\}$.

Login phase

1. U_i inserts U_i's smart card into a card reader and inputs his/her ID_i, PW_i, and imprints biometric B_i. The smart card then computes $e' = h(ID_i||PW_i||H(B_i))$, and compares it with the stored e in the smart card. If this holds, the smart card acknowledges the legitimacy of user U_i, and proceeds with the next step. Otherwise, it terminates the login process.
2. The smart card selects a random number r_1, and computes $RPW_i = h(PW_i||H(B_i))$, $N = h(ID_i||RPW_i)$, $DID_i = ID_i \oplus N$, $C_1 = ID_i \oplus r_1$ and $C_2 = h(ID_i||N||r_1)$.
3. Finally, U_i sends the login request message $\langle DID_i, v, C_1, C_2 \rangle$ to S_j through a public network.

Verification phase

1. S_j computes $ID_i' = DID_i \oplus v \oplus K$. S_j verifies the user's ID_i'. If this holds, S_j accepts the login request and proceeds with the next step. Otherwise, S_j rejects the login request and this phase is terminated.
2. S_j computes $r_1' = C_1 \oplus ID_i'$, $C_2' = h(ID_i'||v \oplus K||r_1')$, and verifies whether $C_2' = C_2$. If this holds, S_j proceeds to the next step. Otherwise, this phase is terminated.

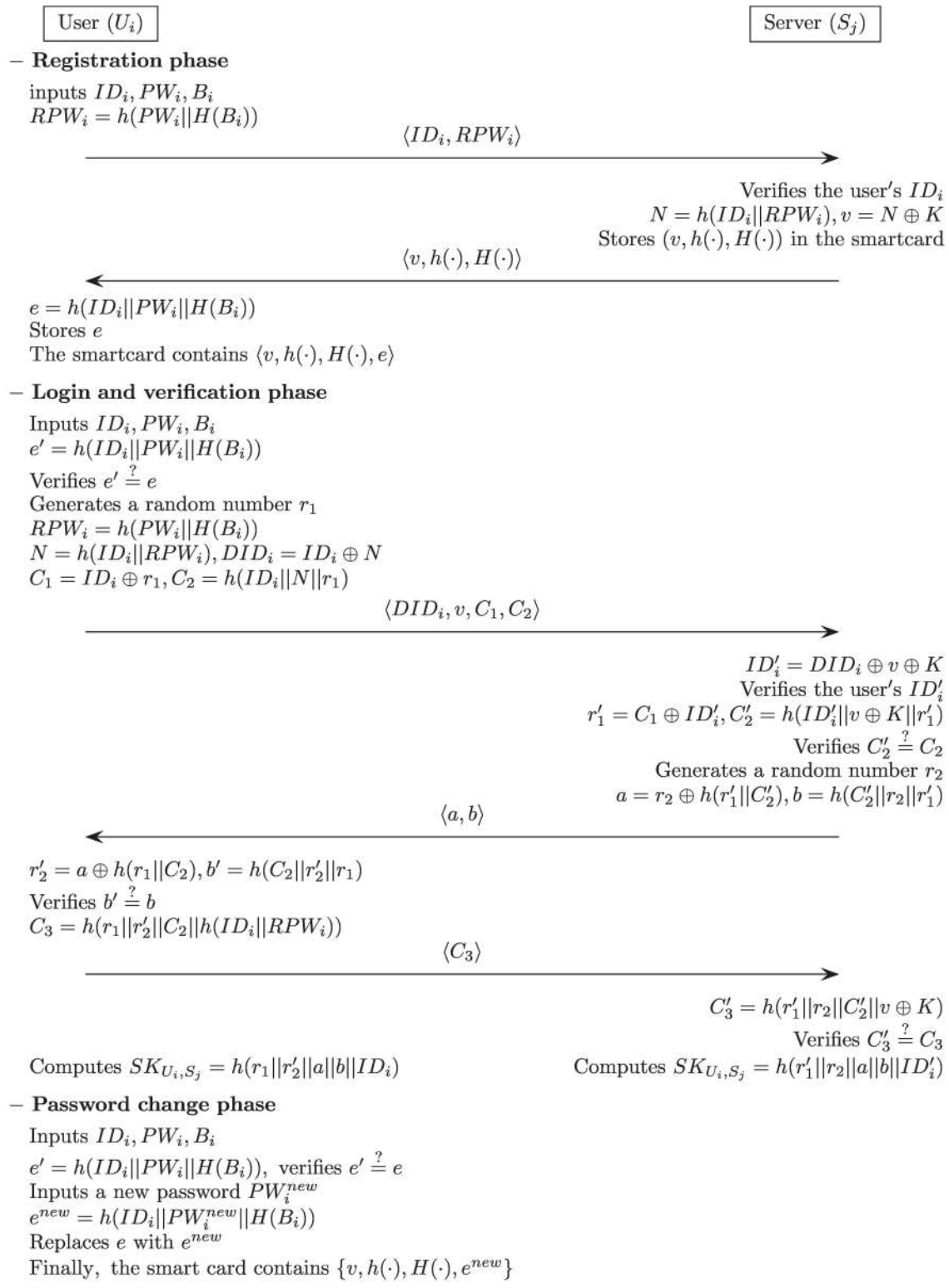

Fig 3. Our proposed scheme.

doi:10.1371/journal.pone.0169414.g003

3. S_j selects a random number r_2 and computes $a = r_2 \oplus h(r_1'||C_2')$ and $b = h(C_2'||r_2||r_1')$. S_j then sends an authentication request message $\langle a, b \rangle$ to U_i through a public network.
4. U_i computes $r_2' = a \oplus h(r_1||C_2)$ and $b' = h(C_2||r_2'||r_1)$. U_i then verifies whether $b' = b$. If this holds, U_i successfully authenticates S_j. Then, U_i computes $C_3 = h(r_1||r_2'||C_2||h(ID_i||RPW_i))$ and sends the acknowledgement message $\langle C_3 \rangle$ to S_j through a public network.
5. S_j computes $C_3' = h(r_1'||r_2||C_2'||v \oplus K)$ and verifies whether $C_3' = C_3$. If this holds, S_j successfully authenticates U_i.
6. Finally, S_j computes a shared session key $SK_{U_i,S_j} = h(r_1'||r_2||a||b||ID_i')$ and U_i also computes the same shared session key $SK_{U_i,S_j} = h(r_1||r_2'||a||b||ID_i)$ successfully.

Password change phase

1. U_i inserts U_i's smart card into a card reader and inputs ID_i, and the old password PW_i, and then imprints biometric B_i. The smart card computes $e' = h(ID_i||PW_i||H(B_i))$. The smart card then verifies whether $e_i' = e_i$, where e_i is stored in the smart card. If this condition is not satisfied, it terminates this phase. Otherwise, the smart card proceeds to the next step.
2. U_i inputs a new password PW_i^{new}, and the smart card then computes $e^{new} = h(ID_i||PW_i^{new}||H(B_i))$,
3. The smart card replaces the existing value e with the new value e^{new}. Finally, the smart card contains the information $\{v, h(\cdot), H(\cdot), e^{new}\}$.

Security analysis of the proposed scheme

In this section, we first adopt Burrows-Abadi-Needham (BAN) logic [47] to prove that a session key can be correctly generated between U_i and S_j. We then verify that our proposed scheme is secure against various attacks through both informal and formal security analysis.

Authentication proof with BAN logic

In this subsection, based on the BAN logic technique, we verify the legitimacy of session keys distributed to the participants who communicate in our proposed protocol. BAN logic [47] is applied, which is a well-known formal logic technique used to analyze the security of cryptographic protocols. Table 2 shows the basic notations to determine BAN logic.

In order to determine the reasonable postulates of BAN logic, five logic rules are also offered as follows.

- Message-meaning rule: $\frac{U|\equiv U \overset{K}{\leftrightarrow} S, U \triangleleft <C>_K}{U|\equiv S|\sim C}$: If U trusts that the key K is shared with S, U sees the C combined with K, then U trusts S once said C.
- Nonce-verification rule: $\frac{U|\equiv \#(C), U|\equiv S|\sim C}{U|\equiv S|\equiv C}$: If U trusts that C's freshness and U trusts S once said C, then U trusts that S trusts C.
- Believe rule: $\frac{U|\equiv C, U|\equiv M}{A|\equiv (C,M)}$: If U trusts C and M, (C, M) are also trusted by U.
- Freshness-conjuncatenation rule: $\frac{U|\equiv \#(C)}{A|\equiv \#(C,M)}$: If freshness of C is trusted by U, then U can trust the freshness of full condition.

Table 2. Notations.

Notations	Description
$U \mid\equiv C$	Condition C is believed by U
$U \lhd C$	U sees condition C
$U \mid\sim C$	U expresses the condition C
$U \Rightarrow C$	Condition C is handled by U
$\sharp(C)$	It makes a fresh C
$U \overset{K}{\longleftrightarrow} S$	U and S share a secret key K
$< C >_K$	Combine condition C using K
$(C)_K$	Perform the hash operation on C using K

doi:10.1371/journal.pone.0169414.t002

- Jurisdiction rule: $\frac{U\mid\equiv S\mid\Rightarrow C, U\mid\equiv S\mid\equiv C}{U\mid\equiv C}$: If U trusts that S has jurisdiction over C, and U trusts that S trusts a condition C, then U also trusts C.

Through our analysis, we intend to meet the following two goals.

- Goal 1. $U \mid\equiv (U \overset{sk}{\longleftrightarrow} S)$
- Goal 2. $S \mid\equiv (U \overset{sk}{\longleftrightarrow} S)$

Next, all transmitted messages in our authentication scheme can be transmuted into an idealized form as follows.

- Message 1. $U \rightarrow S :< ID_i >_v, < ID_i >_{r_1}, (r_1)_{ID_i}$
- Message 2. $S \rightarrow U :< r_2 >_{(r_1)_{ID_i}}, (r_1, r_2)_{ID_i}$
- Message 3. $U \rightarrow S$: $(r_1, r_2, ID_i)_v$

In order to analyze our mechanism, we define the following assumptions.

- A1: $U \mid\equiv \sharp(r_1)$
- A2: $U \mid\equiv \sharp(r_2)$
- A3: $U \mid\equiv (U \overset{v}{\longleftrightarrow} S)$
- A4: $S \mid\equiv (U \overset{v}{\longleftrightarrow} S)$
- A5: $U \mid\equiv S \mid\equiv (U \overset{v}{\longleftrightarrow} S)$
- A6: $S \mid\equiv U \mid\equiv (U \overset{v}{\longleftrightarrow} S)$

Now, we describe our main proof as follows. In order to describe our proof, we use predefined information, including five logic rules, three messages and six assumptions.

According to Message 1, we could derive the following:

- V1: $S \lhd < ID_i >_v, < ID_i >_{r_1}, (r_1)_{ID_i}$

Based on assumption A4 and the message meaning rule, we derive:

- V2: $S \mid\equiv U \mid\sim v$

Based on V2 and the message meaning rule, we derive:

- V3: $S \mid\equiv U \mid\sim ID_i$

Based on V3 and the message meaning rule, we derive:

- V4: $S \mid\equiv U \mid\sim r_1$
  Based on assumption A1, V4 and the freshness conjuncatenation rule, we derive:
- V5: $S \mid\equiv \sharp < ID_i >_{r_1}$
  Based on V4, V5 and the nonce verification rule, we derive:
- V6: $S \mid\equiv U \mid\equiv < ID_i >_{r_1}$
  Based on V3, V6 and the jurisdiction rule, we derive:
- V7: $S \mid\equiv r_1$
  Based on Message 2, we derive:
- V8: $U \triangleleft < r_2 >_{(r_1)_{ID_i}}, (r_1, r_2)_{ID_i}$
  Based on assumption A2, V3 and the message meaning rule, we derive:
- V9: $U \mid\equiv S \mid\sim r_2$
  Based on assumption A2 and the freshness conjuncatenation rule, we derive:
- V10: $U \mid\equiv \sharp < r_2 >_{(r_1)_{ID_i}}$
  Based on V9, V10 and the nonce verification rule, we derive:
- V11: $U \mid\equiv S \mid\equiv < r_2 >_{(r_1)_{ID_i}}$
  Based on V9, V11 and the jurisdiction rule, we derive:
- V12: $U \mid\equiv r_2$
  Based on *sk*, V12 and assumption A1, we derive:
- V13: $U \mid\equiv (U \overset{sk}{\longleftrightarrow} S)$ **(Goal 1.)**
  Based on Message 3, we derive:
- V14: $S \triangleleft (r_1, r_2, ID_i)_v$
  Based on *sk*, V3, V7 and assumption A2, we derive:
- V15: $S \mid\equiv (U \overset{sk}{\longleftrightarrow} S)$ **(Goal 2.)**

The above description clearly shows that U_i and S_j achieve mutual authentication, and based on (Goal.1) and (Goal.2), U_i and S_j trust that session key *sk* is securely shared between them.

Informal security analysis

In this subsection, we examine the security our proposed scheme against various attacks, and the suitability of the basic requirements, as described in Section 2.1, are evaluated. Also, we perform a comparative analysis of previous schemes [9, 10, 27, 28, 33, 38, 40], which is illustrated in Table 3. The results are described as follows.

Proposition 1. Resistant to insider attack. **Proof.** Insider attack means that an insider can directly obtain the user's password from the server and can then access the user's account in another server by using the same password. This attack occurs due to the disclosure of U_i's password PW_i during the registration phase. However, in our scheme, PW_i is transmitted not as a revealed condition but as a form of $RPW_i = h(PW_i||H(B_i))$ with a value of biometrics, $H(B_i)$, when U_i sends a registration request $\langle ID_i, RPW_i \rangle$ to S_j to prohibit insider attack. Thus, our scheme is secure against insider attack.

Proposition 2. Preserve user anonymity. **Proof.** User anonymity is valuable information for the user authentication mechanism, because the disclosure of a user's identity can allow an unauthorized party to track the user's login pattern. Our scheme shields user's identity ID_i

Table 3. Security comparison of the proposed scheme and other related schemes.

Features	Hao's [9]	Jiang's [10]	Chaudhry's [27]	Irshad's [28]	Wu's [33]	Das's [38]	Li's [40]	Proposed Scheme
Proposition 1.	√	√	√	√	×	√	√	√
Proposition 2.	√	√	√	√	×	×	×	√
Proposition 3.	√	√	√	√	√	√	√	√
Proposition 4.	×	×	√	√	×	×	×	√
Proposition 5.	×	×	√	√	×	×	×	√
Proposition 6.	×	×	√	√	×	×	×	√
Proposition 7.	√	√	√	√	√	√	√	√
Proposition 8.	×	×	√	√	×	×	×	√
Proposition 9.	×	√	√	×	×	×	×	√
Proposition 10.	√	√	√	√	√	√	√	√
Proposition 11.	×	×	×	×	√	√	√	√
Proposition 12.	√	√	√	√	√	√	×	√
Formal security proof	×	×	√	√	×	√	√	√

doi:10.1371/journal.pone.0169414.t003

transmitted by messages from the potential risks of exposure in order to fulfill user anonymity. Even if an attacker obtains C_1 by snatching login request $\langle DID_i, v, C_1, C_2\rangle$, it is impossible to calculate ID_i since the random number r_1 is not known.

Proposition 3. Provide mutual authentication. **Proof.** In the verification phase of our proposed scheme, U_i and S_j can authenticate each other by a number of checking processes. In detail, S_j first verifies the login request by checking whether C_2 is correct. U_i also verifies the authentication request by checking whether b is correct. Lastly, S_j verifies the acknowledgement message by checking whether or not C_3 is valid. If all these verification processes are successfully completed, mutual authentication has been executed properly.

Proposition 4. Resistant to off-line password guessing attack. **Proof.** An off-line password guessing attack occurs when an attacker attempts to guess a password and eventually finds the exact user's password in an off-line environment by using the information stored in the smart card or intercepted packets. In our scheme, an attacker can obtain $\{v, h(\cdot), H(\cdot), e\}$ from the stolen smart card and intercept the login request $\langle DID_i, v, C_1, C_2\rangle$. Using these values, the attacker may try to guess the correct password PW_i. However, without knowing ID_i and $H(B_i)$, the attacker cannot guess PW_i. In addition, $H(B_i)$ is hashed biometric information, which is only known by U_i. Therefore, our proposed scheme can withstand an off-line password guessing attack.

Proposition 5. Resistant to user impersonation attack. **Proof.** An user impersonation attack occurs when an attacker pretends to be the legal user with the counterfeited login request by using the information that has accumulated from the smart cards and the intercepted packets. Suppose that an attacker obtains $\{v, h(\cdot), H(\cdot), e\}$ from the stolen smart card. The attacker then generates a random number r_1^* and attempts to compute $DID_i = ID_i \oplus N$, $C_1^* = ID_i \oplus r_1^*$ and $C_2^* = h(ID_i||N||r_1^*)$. In order to compute DID_i, C_1^*, and C_2^*, the attacker must obtain S_j's secret key K. However, it is impossible to obtain K in our scheme. Therefore, the attacker cannot generate a sufficient login request $\langle DID_i, v, C_1^*, C_2^*\rangle$ to deceive S_j.

Proposition 6. Resistant to server spoofing attack. **Proof.** A server spoofing attack occurs when an attacker masquerades as a legal server with the counterfeited authentication request by using the information stored in the smart card and the intercepted packets. Suppose that an attacker obtains $\{v, h(\cdot), H(\cdot), e\}$ from the stolen smart card and intercepts the login request $\langle DID_i, v, C_1, C_2\rangle$. The attacker then generates a random number r_2^* and attempts to

modify the authentication request $\langle a, b\rangle$ to impersonate a legal S_j. However, the attacker cannot compute a sufficient authentication request to deceive U_i because it is impossible to obtain r_1. In addition, as mentioned above, our scheme can guarantee protection against off-line password guessing attacks. Therefore, our proposed scheme can withstand a sever spoofing attack.

Proposition 7. Resistant to replay attack. **Proof.** A replay attack occurs when an attacker deceives a legitimate participant by recycling of the same packets acquired in previous sessions. Suppose that an attacker intercepts the previous login request $\langle DID_i, v, C_1, C_2\rangle$. The attacker might then pretend to be a valid user to login to the server by sending the message. However, if the attacker sends the previous login request, the server would obviously reject the request, because our scheme can detect an invalid random number by comparing it to the C_2 value. In addition, our scheme uses different random numbers in each session. Therefore, our proposed scheme can withstand a replay attack.

Proposition 8. Provide a password verification process. **Proof.** It is a possible for a user to input an incorrect password by mistake. However, for the password verification procedure, an incorrect password will be detected after performing the authentication phase. Our scheme overcomes this type of inefficiency problem, evaluating the correctness of the password by verifying the value e in the early login phase.

Proposition 9. Provide an efficient password change phase. **Proof.** In general, the smart card should carry out the verification process by itself when password alteration occurs. The efficiency of a security scheme can be increased through its own process without communicating to the server S_j. Our proposed scheme performs existing password checks in a self-verification process within the smart card. After examining, it will switch the computed value e^{new} from the new password with the existed value e in a convenient and efficient way.

Proposition 10. Provide session key agreement. **Proof.** In our scheme, U_i and S_j compute the session key SK_{U_i,S_j} after the mutual authentication process. In addition, the session key is generated by the random number and a one-way hash function. Hence, this session key is different in each session, and it is not possible to derive the session key from the intercepted messages and stolen smart card. Thus, our scheme guarantees that each session key is generated and distributed in a secure way.

Proposition 11. No time synchronization. **Proof.** In timestamp-based authentication protocols, when transmitting a message between a user and a server, the clocks of all devices should be synchronized. Therefore, the possibility that an error has occurred is high. However, to avoid this problem, our scheme uses a random numbers r_1, r_2 based authentication mechanism instead of a time-stamp technique.

Proposition 12. No access control table. We analyzed the inefficiency risk of Li et al.'s scheme with respect to the access control table. In order to overcome this weakness, our scheme removes the access control table without loss of protection from various types of attacks.

Formal security analysis

In this subsection, through the formal proof method, we demonstrate that our proposed scheme is secure. First, we specify a hash function as follows.

Definition 1 A hash function f: $\{0, 1\}^* \rightarrow \{0, 1\}^n$ is a one-direction function [48, 49] that takes the input $x \in \{0, 1\}^*$ of arbitrary length and outputs a bit string with a fixed-length $f(x) \in \{0, 1\}^n$ that is referred to as the "message digest" or "hash value". When using cryptographic hash functions, the following three common levels of security must be considered:

- It is impossible to acquire input x under the conditions of the hash value $y = h(x)$ and the given hash function $h(\cdot)$.

- It is impossible to acquire other input x', when given the input x and $f(x') = f(x)$.
- It is impossible to acquire the inputs (x, x'), where $x \neq x'$, when given $f(x) = f(x')$.

Theorem 1 It is assumed that the one-way hash function $h(\cdot)$ performs like an oracle. Under this assumption, our proposed mechanism is provably secure against an adversary $\mathcal{A}$ for protecting the user's personal information, such as identity ID_i, password PW_i, biometrics B_i, and the server's secret key K.

Reveal Given the hash result $y = h(x)$, this random oracle will unconditionally output the input x.

Proof The similar method used in [20] is applied in our authentication mechanism for verification. In order to prove it, we assume that an attacker $\mathcal{A}$ is able to extract the U_i's identity ID_i, password PW_i, biometrics B_i, and the server's secret key K. For this, $\mathcal{A}$ runs the experimental algorithm that is shown in Table 4, $EXP_{HASH,A}^{ABUAKAS}$ for our anonymous biometric-based user authentication with key agreement scheme, called ABUAKAS.

We define the success probability for $EXP_{HASH,A}^{ABUAKAS}$ as $Success_{HASH,A}^{ABUAKAS} = |Pr[EXP_{HASH,A}^{ABUAKAS} = 1] - 1|$, where $Pr()$ means the probability of $EXP_{HASH,A}^{ABUAKAS}$. The advantage function for this experiment becomes $Adv_{HASH,A}^{ABUAKAS}(t, q_R) = max_A\ Success_{HASH,A}^{ABUAKAS}$ in which the maximum is determined by three factors; all of $\mathcal{A}$, the execution time t, and the number of queries q_R derived from Reveal oracle. If the attacker $\mathcal{A}$ is assumed that $\mathcal{A}$ is able to resolve the hash function problem, $\mathcal{A}$ could directly obtain U_i's identity ID_i, password PW_i, biometrics B_i, and the server's secret key K. Refer to attack experiment described in Table 4. In this case, $\mathcal{A}$ will discover the complete connections between U_i and S_j. However, it is a computationally infeasible problem

Table 4. Algorithm $EXP_{HASH,A}^{ABUAKAS}$.

1. Eavesdrop login request message $\langle DID_i, v, C_1, C_2 \rangle$
2. Call the Reveal oracle. Let $(ID_i', N', r_1') \leftarrow Reveal(C_2)$
3. Eavesdrop login request message $\langle a, b \rangle$
4. Call the Reveal oracle. Let $(C_2', r_2', r_1'') \leftarrow Reveal(b)$
5. Computes $C_1' = ID_i' \oplus r_1'$
6. **if** $(C_1' = C_1)$ **then**
7. Accepts ID_i' as the correct ID_i of user
8. Call the Reveal oracle. Let $(ID_i'', RPW_i'') \leftarrow Reveal(N')$
9. **if** $(ID_i'' = ID_i')$ **then**
10. Call the Reveal oracle. Let $(PW_i', B_i') \leftarrow Reveal(RPW_i')$
11. Computes $C_2'' = h(ID_i''||N'||r_1')$
12. **if** $(C_2'' = C_2')$ **then**
13. Accept ID_i'', PW_i', B_i' as the correct information of U_i
K as the correct key of S_j
14. **return** 1
15. **else**
16. **return** 0
17. **end if**
18. **else**
19. **return** 0
20. **end if**
21. **else**
22. **return** 0
23. **end if**

doi:10.1371/journal.pone.0169414.t004

to invert a one-way hash function $h(\cdot)$, i.e., $Adv_{HASH,A}^{ABUAKAS}(t) \leq \epsilon, \forall \epsilon > 0$. Then, we have $Adv_{HASH,A}^{ABUAKAS}(t, q_R) \leq \epsilon$, since $Adv_{HASH,A}^{ABUAKAS}(t, q_R)$ depends on $Adv_{HASH,A}^{ABUAKAS}(t)$. Therefore, our proposed scheme is provably secure against the attacker $\mathcal{A}$ for deriving ID_i, PW_i, B_i and K.

Formal security verification using AVISPA tool

In this subsection, we simulate our scheme for the formal security verification using the Automated Validation of Internet Security Protocols and Applications(AVISPA) tool [50] in order to prove that our scheme can withstand both passive and active attacks. We first overview the structure of the AVISPA tool, and then specify our authentication mechanism using High Level Protocols Specification Language(HLPSL) [51]. Lastly, we conduct a simulation using the AVISPA tool and show that our scheme guarantees safety.

Overview of AVISPA tool. AVISPA is a well-known formal method, which has been used to verify the security of protocols. AVISPA provides the specification of security protocols and properties by using a modular and expressive specification language. A number of authentication studies have been conducted [12, 19, 26, 30–32, 39] based on the AVISPA simulation tool. The AVISPA tool can be utilized by external users because the web-interface is accessible from the website [52]. It is also provided as a package(SPAN) that can be installed on the Linux and Mac OS operating systems. The architecture of the AVISPA tool is illustrated in Fig 4.

As shown in Fig 4, the tool takes a specification, as an input, written in HLPSL and produces the results of the test, as an output, from the different back-ends. More specifically, specifications of protocols written in HLPSL are automatically translated by the HLPSL2IF into

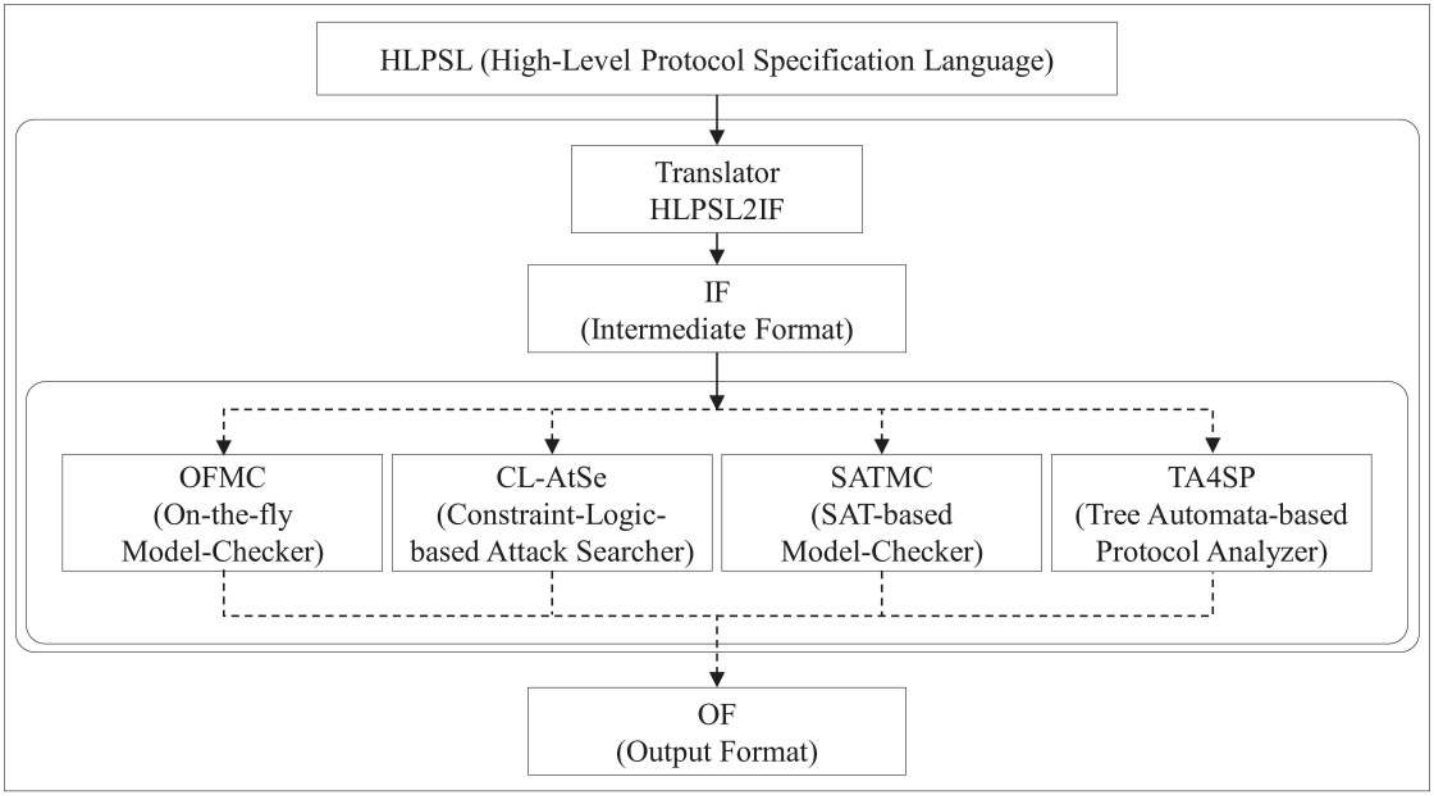

Fig 4. Architecture of AVISPA tool.

doi:10.1371/journal.pone.0169414.g004

Intermediate Format (IF) specifications, which are then given as input to the different back-ends. The back-ends consist of four parts [50]: On-the-fly Model-Checker(OFMC), Constraint-Logic-based Attack Searcher(CL-AtSe), SAT based Model Checker(SATMC), and Tree Automata-based on Automatic Approximations for the Analysis of Security Protocols (TA4SP). A detailed description of each part is as follows:

- On-the-fly Model-Checker(OFMC): uses different symbolic techniques to explore the state space in a demand-driven way.
- Constraint-Logic-based Attack Searcher(CL-AtSe): uses simplification heuristics and redundancy elimination techniques.
- SAT based Model Checker(SATMC): uses SAT-solvers in order to find a proposition leading to a fail in the model.
- Tree Automata-based on Automatic Approximations for the Analysis of Security Protocols (TA4SP): uses regular tree languages in order to evaluate the intruder knowledge.

Specifying the proposed scheme. This section provides descriptions of the specifications of our scheme in HLPSL. We first implement the basic roles for a user U_i and a server S_j during the registration, login, and verification phase, and then specify the other roles for the session, environment, and goal. In our specifications, the type declaration *channel*(*dy*) indicates that the channel is influenced by the Dolev-Yao threat model [53]. The attacker has full manage over the public channel, such that the attacker can intercept or eavesdrop on all messages sent by agents. In addition, the declaration *secret*({*K*}, *subs*1, S_j) indicates that the secret key K is only known to S_j and *secret*({*IDi*, *PWi*, *Bi*}, *subs*2, U_i) indicates that $\langle ID_i, PW_i, B_i \rangle$ are only known to U_i. In our implementation, we assume that the bio-hash function $H(\cdot)$ is the same as the one-way hash function $h(\cdot)$. The role specification of the user U_i is shown in Fig 5.

During the registration phase, U_i sends a registration request message $\langle ID_i, RPW_i \rangle$ to server S_j through a secure channel using *Snd*() operation, and receives the information $\{v, h(\cdot), H(\cdot)\}$ stored in the smart card from S_j using *Rcv*() operation securely. During the login and verification phase, U_i generates a random number r_1 using *new*() operation and sends the login request message $\langle DID_i, v, C_1, C_2 \rangle$ to S_j through a public channel. U_i then receives the authentication request message $\langle a, b \rangle$ from S_j through a public channel. Finally, U_i sends the acknowledgement message $\langle C_3 \rangle$ to S_j through a public channel. The declaration *witness*(*Ui*, *Sj*, *alice_bob_r*1, *R*1′) indicates that U_i has freshly generated the random number *R*1 for S_j. The declaration *request*(*Sj*, *Ui*, *bob_alice_r*2, *R*2′) denotes that U_i authenticates the server S_j. We similarly implemented the role for the server S_j as shown in Fig 6.

During the registration phase, S_j receives the registration request message $\langle ID_i, RPW_i \rangle$ from the user U_i through a secure channel. After receiving the registration request message, the S_j issues a smart card with the $\{v, h(\cdot), H(\cdot)\}$ and sends it to U_i through a secure channel using the *Snd*() operation. During the login and verification phase, S_j receives the login request message $\langle DID_i, v, C_1, C_2 \rangle$ from the user U_i through a public channel. The S_j then generates a random number r_2 and sends the authentication request message $\langle a, b \rangle$ to U_i through a public channel. Finally, S_j receives the acknowledgement message $\langle C_3 \rangle$ from the user U_i through a public channel using the *Rcv*() operation. The declaration *witness*(*Sj*, *Ui*, *bob_alice_r*2, *R*2′) indicates that S_j has freshly generated the random number *R*2 for U_i. The declaration *request*(*Ui*, *Sj*, *alice_bob_r*1, *R*1′) indicates that S_j authenticates the user U_i.

We have therefore provided the specification in HLPSL for the roles including the session, environment, and gaol. The detailed specification of each role is described in Fig 7.

```
role alice (Ui,Sj : agent,
            SKuisj: symmetric_key,
            H : hash_func,
            Snd,Rcv: channel(dy))
played_by Ui
def=
   local State : nat,
            IDi, PWi, RPWi, Bi, K, R1, R2, N, V, DIDi, C1, C2, C3, A, B : text
   const alice_bob_r1, bob_alice_r2, subs1, subs2 : protocol_id
   init State := 0
   transition
% Registration phase
    1. State = 0 /\ Rcv(start) =|>
          State' := 1 /\ RPWi' := H(PWi.H(Bi))
           /\ Snd({IDi.RPWi'}_SKuisj)
           /\ secret({K}, subs1, Sj)
           /\ secret({IDi, PWi, Bi}, subs2, Ui)
    2. State = 1 /\ Rcv({H.xor(H(IDi.H(PWi.H(Bi))), K)}_SKuisj) =|>
% Login phase
         State' := 2 /\ R1' := new()
         /\ RPWi' := H(PWi.H(Bi))
         /\ V' := xor(H(IDi.H(PWi.H(Bi))), K)
         /\ N' := H(IDi.H(PWi.H(Bi)))
         /\ DIDi' := xor(IDi, H(IDi.H(PWi.H(Bi))))
         /\ C1' := xor(IDi, R1')
         /\ C2' := H(IDi.H(IDi.H(PWi.H(Bi))).R1')
         /\ Snd(DIDi'.V'.C1'.C2')
         /\ witness(Ui, Sj, alice_bob_r1, R1')
% Verification phase
    3. State = 2 /\ Rcv(xor(R2', H(R1'.H(IDi.H(IDi.H(PWi.H(Bi))).R1')))
.H(H(IDi.H(IDi.H(PWi.H(Bi))).R1').R2'.R1')) =|>
         State' := 3 /\ C3' := H(R1'.R2'.H(IDi.H(IDi.H(PWi.H(Bi))).R1')
.H(IDi.H(PWi.H(Bi))))
         /\ Snd(C3')
         /\ request(Sj, Ui, bob_alice_r2, R2')
end role
```

Fig 5. Role specification in HLPSL for the user U_i.

doi:10.1371/journal.pone.0169414.g005

The session part involves the starting parameters, local variables, and composition of agents. The environment part involves the global constants, attacker knowledge, security goals, and the composition of more than one session run in parallel. In our simulation, the following two secrecy goals and two authentications are verified:

- secrecy_of subs1: indicates that the secret key K is only known to the legal server S_j.
- secrecy_of subs2: indicates that the information including ID_i, PW_i and B_i is only known to the legal user U_i.

```
role bob (Ui,Sj : agent,
          SKuisj: symmetric_key,
          H : hash_func,
          Snd,Rcv: channel(dy))
played_by Sj
def=
  local State : nat,
          IDi, PWi, RPWi, Bi, K, R1, R2, N, V, DIDi, C1, C2, C3, A, B : text
  const alice_bob_r1, bob_alice_r2,
          subs1, subs2 : protocol_id
  init State := 0
  transition
% Registration phase
    1. State = 0 /\ Rcv({IDi.H(PWi.H(Bi))}_SKuisj) =|>
       State' := 1 /\ secret({K}, subs1, Sj)
          /\ secret({IDi, PWi, Bi}, subs2, Ui)
          /\ Snd({H.xor(H(IDi.H(PWi.H(Bi))), K)}_SKuisj)
% Login phase
    2. State = 1 /\ Rcv(xor(IDi, H(IDi.H(PWi.H(Bi)))).xor(H(IDi.H(PWi.H(Bi))), K)
.xor(IDi, R1').H(IDi.H(IDi.H(PWi.H(Bi))).R1')) =|>
% Verification phase
       State' := 2 /\ R2' := new()
       /\ IDi' := xor(xor(IDi, H(IDi.H(PWi.H(Bi)))), xor(H(IDi.H(PWi.H(Bi))), K), K)
       /\ R1' := xor(xor(IDi, R1'), IDi')
       /\ C2' := H(IDi'.xor(xor(H(IDi'.H(PWi.H(Bi))), K), K).R1')
       /\ A' := xor(R2', H(R1'.C2'))
       /\ B' := H(C2'.R2'.R1')
       /\ Snd(A'.B')
       /\ witness(Sj, Ui, bob_alice_r2, R2')
    3. State = 2 /\ Rcv(H(R1'.R2'.H(IDi.H(IDi.H(PWi.H(Bi))).R1')
.xor(xor(H(IDi'.H(PWi.H(Bi))), K), K))) =|>
       State' := 3 /\ request(Ui, Sj, alice_bob_r1, R1')
end role
```

Fig 6. Role specification in HLPSL for the server S_j.

doi:10.1371/journal.pone.0169414.g006

- authentication_on alice_bob_r1: indicates that U_i generates a random number r_1, where r_1 is only taken to U_i. If the server S_j securely receives it from the message, S_j then authenticates U_i.
- authentication_on bob_alice_r2: indicates that S_j generates a random number r_2, where r_2 is only taken to S_j. If the user U_i securely receives it from the message, U_i also authenticates S_j.

Simulation results. We simulated our proposed scheme using the AVISPA tool in order to check that our scheme can guarantee safety. The simulation results under the OFMC and CL-AtSe back-ends are shown in Fig 8. The results clearly demonstrate that our scheme is *SAFE* under each bank-end. Therefore, we conclude that our proposed scheme can guarantee protection against passive and active attacks such as replay and man-in-the middle attacks.

```
role session(Ui, Sj: agent,
        SKuisj : symmetric_key,
        H : hash_func)
def=
   local SI, SJ, RI, RJ: channel (dy)
   composition
        alice(Ui, Sj, SKuisj, H, SI, RI)
        /\ bob(Ui, Sj, SKuisj, H, SJ, RJ)
end role
```

```
role environment()
def=
   const ui, sj: agent,
        skuisj : symmetric_key,
        h : hash_func,
        %H : one-way/bio hash function
        idi, pwi, bi, k, r1, r2: text,
        alice_bob_r1, bob_alice_r2,
        subs1, subs2 : protocol_id
   intruder_knowledge = {ui, sj, h}
   composition
        session(ui, sj, skuisj, h)
        /\ session(ui, sj, skuisj, h)
end role

goal
   secrecy_of subs1
   secrecy_of subs2
   authentication_on alice_bob_r1
   authentication_on bob_alice_r2
end goal
environment()
```

Fig 7. Role specification in HLPSL for the session, environment and goal.

doi:10.1371/journal.pone.0169414.g007

```
SUMMARY
   SAFE
DETAILS
   BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
   /home/jaewook/Desktop/span/testsuite/results
   /PLOS_ONE.if
GOAL
   as specified
BACKEND
   OFMC
STATISTICS
   TIME 28 ms
   parseTime 0 ms
   visitedNodes: 13 nodes
   depth: 4 plies
```

```
SUMMARY
   SAFE
DETAILS
   BOUNDED_NUMBER_OF_SESSIONS
   TYPED_MODEL
PROTOCOL
   /home/jaewook/Desktop/span/testsuite/results
   /PLOS_ONE.if
GOAL
   As specified
BACKEND
   CL-AtSe
STATISTICS
   Analysed : 2 states
   Reachable : 0 state
   Translation: 0.02 seconds
   Computation: 0.00 seconds
```

Fig 8. Simulation results under the OFMC and CL-AtSe back-ends.

doi:10.1371/journal.pone.0169414.g008

Table 5. Performance comparison of the proposed scheme and other related schemes.

Phases/Schemes	Lee et al. [34]	Das [38]	Mir et al. [39]	Li et al. [40]	Proposed Scheme
Registration phase	$4T_H$	$4T_H$	$5T_H$	$4T_H$	$4T_H$
Login phase	$2T_H$	$3T_H$	$7T_H$	$3T_H$	$5T_H$
Verification phase	$10T_H$	$11T_H$	$10T_H$	$11T_H$	$9T_H$
Password change phase	$3T_H$	$7T_H$	$5T_H$	$8T_H$	$3T_H$
Total cost	$19T_H$	$25T_H$	$27T_H$	$26T_H$	$21T_H$
Execution time	$\approx$ 3.8ms	$\approx$ 5.0ms	$\approx$ 5.4ms	$\approx$ 5.2ms	$\approx$ 4.2ms

doi:10.1371/journal.pone.0169414.t005

Performance analysis of the proposed scheme

In this section, we have conducted the comparison of the computational costs and execution time for the proposed scheme with other hash-based schemes [34, 38–40]. Generally, the computational cost is examined based on the respective operations in authentication protocol. Accordingly, this analysis of computational cost concentrates on the operations that are conducted by the members, such as user and server. For the evaluation of the computational costs, we define the computational parameter T_H as the time taken to execute a one-way hash function/bio-hash function.

Table 5 provides a summary of the comparison of the computational overheads. Table 5 shows that Lee et al. [34], Das [38], Mir et al. [39], Li et al. [40] and our proposed scheme require the total computational overheads of $19T_H$, $25T_H$, $27T_H$, $26T_H$, and $21T_H$, respectively.

The results show that our proposed scheme is relatively superior to that proposed in a number of related studies [38–40]. In addition, as is shown in Table 3, our proposed scheme guarantees safety against a variety of existing attacks. According to [40], the actual execution times for the one-way hash function T_H is 0.2ms. In Table 5, we also listed the time consumption of our proposed scheme and of the schemes presented in the other related studies [34, 38–40]. Table 5 shows that the execution time of our proposed scheme requires only 4.2 ms ($\approx 21 \times 0.2$ ms); it can therefore be considered as of minor significance. On the other hand, the execution time of Das's scheme [38], Mir et al.'s scheme [39] and Li et al.'s scheme [40] require 5.0ms ($\approx 25 \times 0.2$ ms), 5.4 ms ($\approx 27 \times 0.2$ ms) and 5.2 ms ($\approx 26 \times 0.2$ ms), respectively; these schemes are therefore proven to be slightly ineffective compared to our scheme. Table 5 demonstrates that our proposed mechanism considers efficiency.

Conclusions

In this paper, we demonstrate that Li et al.'s scheme has a number of critical vulnerabilities and we propose an extended authentication scheme to overcome these defects. Our proposed scheme has been thoroughly verified in terms of a variety of security features, and the proof result demonstrates that a session key can be correctly generated between the communicating parties. In addition, a performance comparison for the proposed scheme in relation to the schemes proposed in other studies was carried out, and we consider that our proposed scheme has sufficient efficiency and robustness for an integrated EPR information system. In the future, we will propose a new authentication scheme applying the fuzzy extractor technique instead of the biohashing method and analyze the new scheme not only in terms of computation cost, but also in terms of communication and smart card storage cost.

Acknowledgments

All authors, especially the corresponding author Dongho Won, would like to thank the anonymous reviewers for their time and invaluable comments and suggestions on this paper.

Author Contributions

Conceptualization: JJ DK DL DW.

Data curation: JJ DK DL.

Formal analysis: JJ DK.

Funding acquisition: JJ DW.

Investigation: JJ DK DL.

Methodology: JJ DK.

Project administration: JJ DW.

Resources: JJ DW.

Software: JJ DK DL.

Supervision: JJ DW.

Validation: JJ DW.

Visualization: JJ DK DL.

Writing – original draft: JJ DW.

Writing – review & editing: JJ DK DL DW.

References

1. Takeda H, Matsumura Y, Kuwata S, Nakano H, Sakamoto N, Yamamoto R. Architecture for networked electronic patient record systems. International journal of medical informatics. 2000; 60(2):161–167. doi: 10.1016/S1386-5056(00)00116-7 PMID: 11154967
2. Gritzalis S, Lambrinoudakis C, Lekkas D, Deftereos S. Technical guidelines for enhancing privacy and data protection in modern electronic medical environments. IEEE Transactions on Information Technology in Biomedicine. 2005; 9(3):413–423. doi: 10.1109/TITB.2005.847498 PMID: 16167696
3. Chan AT, Cao J, Chan H, Young G. A web-enabled framework for smart card applications in health services. Communications of the ACM. 2001; 44(9):76–82. doi: 10.1145/383694.383710
4. Wang DW, Liu DR, Chen YC. A mechanism to verify the integrity of computer-based patient records. J China Assoc Med Inform. 1999; 10:71–84.
5. Lamport L. Password authentication with insecure communication. Communications of the ACM. 1981; 24(11):770–772. doi: 10.1145/358790.358797
6. Lee CC, Hwang MS, Liao IE. Security enhancement on a new authentication scheme with anonymity for wireless environments. IEEE Transactions on Industrial Electronics. 2006; 53(5):1683–1687. doi: 10.1109/TIE.2006.881998
7. Wu ZY, Lee YC, Lai F, Lee HC, Chung Y. A secure authentication scheme for telecare medicine information systems. Journal of medical systems. 2012; 36(3):1529–1535. doi: 10.1007/s10916-010-9614-9 PMID: 20978928
8. He D, Jianhua C, Rui Z. A more secure authentication scheme for telecare medicine information systems. Journal of Medical Systems. 2012; 36(3):1989–1995. doi: 10.1007/s10916-011-9658-5
9. Hao X, Wang J, Yang Q, Yan X, Li P. A Chaotic Map-based Authentication Scheme for Telecare Medicine Information Systems. Journal of Medical Systems. 2013; 37(2):9919. doi: 10.1007/s10916-012-9919-y PMID: 23334801

10. Jiang Q, Ma J, Lu X, Tian Y. Robust chaotic map-based authentication and key agreement scheme with strong anonymity for telecare medicine information systems. Journal of medical systems. 2014; 38 (2):1–8. doi: 10.1007/s10916-014-0012-6
11. Moon J, Choi Y, Kim J, Won D. An Improvement of Robust and Efficient Biometrics Based Password Authentication Scheme for Telecare Medicine Information Systems Using Extended Chaotic Maps. Journal of medical systems. 2016; 40(3):1–11. doi: 10.1007/s10916-015-0422-0
12. Amin R, Islam SH, Biswas GP, Khan MK, Li X. Cryptanalysis and enhancement of anonymity preserving remote user mutual authentication and session key agreement scheme for e-health care systems. Journal of medical systems. 2015; 39(11):1–21. doi: 10.1007/s10916-015-0318-z
13. He D, Zeadally S, Kumar N, Lee JH. Anonymous authentication for wireless body area networks with provable security. IEEE Systems Journal. 2016; PP(99):1–12. doi: 10.1109/JSYST.2016.2544805
14. Kim J, Lee D, Jeon W, Lee Y, Won D. Security analysis and improvements of two-factor mutual authentication with key agreement in wireless sensor networks. Sensors. 2014; 14(4):6443–6462. doi: 10.3390/s140406443 PMID: 24721764
15. Choi Y, Lee D, Kim J, Jung J, Nam J, Won D. Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography. Sensors. 2014; 14(6):10081–10106. doi: 10.3390/s140610081 PMID: 24919012
16. Nam J, Kim M, Paik J, Lee Y, Won D. A provably-secure ECC-based authentication scheme for wireless sensor networks. Sensors. 2014; 14(11):21023–21044. doi: 10.3390/s141121023 PMID: 25384009
17. Khan MK. Fingerprint biometric-based self-authentication and deniable authentication schemes for the electronic world. IETE Technical Review. 2014; 26(3):191–195.
18. Chaudhry SA, Naqvi H, Farash MS, Shon T, Sher M. An improved and robust biometrics-based three factor authentication scheme for multiserver environments. The Journal of Supercomputing. 2015:1–17.
19. Amin R, Biswas GP. A novel user authentication and key agreement protocol for accessing multi-medical server usable in tmis. Journal of medical systems. 2015; 39(3):1–17. doi: 10.1007/s10916-015-0217-3
20. Moon J, Choi Y, Jung J, Won D. An Improvement of Robust Biometrics-Based Authentication and Key Agreement Scheme for Multi-Server Environments Using Smart Cards. PloS one. 2015; 10(12): e0145263. doi: 10.1371/journal.pone.0145263 PMID: 26709702
21. Mishra D, Das AK, Mukhopadhyay S. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Expert Systems with Applications. 2014; 41 (18):8129–8143. doi: 10.1016/j.eswa.2014.07.004
22. He D, Kumar N, Shen H, Lee JH. One-to-many authentication for access control in mobile pay-TV systems. Science China Information Sciences, 2015:1–14.
23. Khan I, Chaudhry SA, Sher M, Khan JI, Khan MK. An anonymous and provably secure biometric-based authentication scheme using chaotic maps for accessing medical drop box data. The Journal of Supercomputing. 2016:1–19.
24. Lee CC, Hsu CW. A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps. Nonlinear Dynamics. 2013; 71(1):201–211. doi: 10.1007/s11071-012-0652-3
25. Giri D, Maitra T, Amin R, Srivastava PD. An efficient and robust rsa-based remote user authentication for telecare medical information systems. Journal of medical systems. 2015; 39(1):1–9. doi: 10.1007/s10916-014-0145-7
26. Amin R, Biswas GP. An improved rsa based user authentication and session key agreement protocol usable in tmis. Journal of Medical Systems. 2015; 39(8):1–14. doi: 10.1007/s10916-015-0262-y
27. Chaudhry SA, Mahmood K, Naqvi H, Khan MK. An improved and secure biometric authentication scheme for telecare medicine information systems based on elliptic curve cryptography. Journal of Medical Systems. 2015; 39(11):1–12. doi: 10.1007/s10916-015-0335-y
28. Irshad A, Sher M, Nawaz O, Chaudhry SA, Khan I, Kumari S. A secure and provable multi-server authenticated key agreement for TMIS based on Amin et al. scheme. Multimedia Tools and Applications. 2016:1–27.
29. Islam SH, Khan MK. Cryptanalysis and improvement of authentication and key agreement protocols for telecare medicine information systems. Journal of medical systems. 2014; 38(10):1–16. doi: 10.1007/s10916-014-0135-9
30. Amin R, Biswas GP. A secure three-factor user authentication and key agreement protocol for tmis with user anonymity. Journal of medical systems. 2015; 39(8):1–19.

31. Amin R, Islam SH, Biswas GP, Khan MK, Obaidat MS. Design and analysis of an enhanced patient-server mutual authentication protocol for telecare medical information system. Journal of medical systems. 2015; 39(11):1–20. doi: 10.1007/s10916-015-0307-2

32. Amin R, Islam SH, Biswas GP, Khan MK, Kumar N. An efficient and practical smart card based anonymity preserving user authentication scheme for TMIS using elliptic curve cryptography. Journal of medical systems. 2015; 39(11):1–18. doi: 10.1007/s10916-015-0351-y

33. Wu ZY, Chung Y, Lai F, Chen TS. A password-based user authentication scheme for the integrated EPR information system. Journal of medical systems. 2012; 36(2):631–638. doi: 10.1007/s10916-010-9527-7 PMID: 20703670

34. Lee TF, Chang IP, Lin TH, Wang CC. A secure and efficient password-based user authentication scheme using smart cards for the integrated epr information system. Journal of medical systems. 2013; 37(3):1–7.

35. Kocher P, Jaffe J, Jun B. Differential power analysis. In: Annual International Cryptology Conference. Springer; 1999. p. 388–397.

36. Wen F. A more secure anonymous user authentication scheme for the integrated EPR information system. Journal of medical systems. 2014; 38(5):1–7. doi: 10.1007/s10916-014-0042-0

37. Li CT, Weng CY, Lee CC, Wang CC. Secure user authentication and user anonymity scheme based on quadratic residues for the integrated EPRIS. Procedia Computer Science. 2015; 52:21–28. doi: 10.1016/j.procs.2015.05.008

38. Das AK. A secure and robust password-based remote user authentication scheme using smart cards for the integrated epr information system. Journal of medical systems. 2015; 39(3):1–14. doi: 10.1007/s10916-015-0204-8

39. Mir O, van der Weide T, Lee CC. A secure user anonymity and authentication scheme using AVISPA for telecare medical information systems. Journal of Medical Systems. 2015; 39(9):1–16. doi: 10.1007/s10916-015-0265-8

40. Li CT, Weng CY, Lee CC, Wang CC. A hash based remote user authentication and authenticated key agreement scheme for the integrated EPR information system. Journal of medical systems. 2015; 39 (11):1–11. doi: 10.1007/s10916-015-0322-3

41. Jin ATB, Ling DNC, Goh A. Biohashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern recognition. 2004; 37(11):2245–2255. doi: 10.1016/j.patcog.2004.04.011

42. Chung Y, Choi S, Lee Y, Park N, Won D. An Enhanced Lightweight Anonymous Authentication Scheme for a Scalable Localization Roaming Service in Wireless Sensor Networks. Sensors. 2016; 16 (10):1653. doi: 10.3390/s16101653

43. Kang D, Jung J, Mun J, Lee D, Choi Y, Won D. Efficient and robust user authentication scheme that achieve user anonymity with a Markov chain. Security and Communication Networks. 2016; 9 (11):1462–1476. doi: 10.1002/sec.1432

44. Jung J, Kim J, Choi Y, Won D. An Anonymous User Authentication and Key Agreement Scheme Based on a Symmetric Cryptosystem in Wireless Sensor Networks. Sensors. 2016; 16(8):1299. doi: 10.3390/s16081299

45. Choi Y, Lee Y, Won D. Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction. International Journal of Distributed Sensor Networks. 2016; 2016:2. doi: 10.1155/2016/8572410

46. Das AK, Goswami A. A secure and efficient uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care. Journal of medical systems. 2013; 37(3):1–16. doi: 10.1007/s10916-013-9948-1

47. Burrows M, Abadi M, Needham RM. A logic of authentication. In: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. vol. 426. The Royal Society; 1989. p. 233–271.

48. Stallings W. Cryptography and network security: principles and practices. Pearson Education India; 2006.

49. FIPS P. 180–1. Secure hash standard. National Institute of Standards and Technology. 1995; 17:45.

50. AVISPA, Automated validation of internet security protocols and applications. http://www.avispa-project.org (accessed October 2014).

51. Von Oheimb, D. (2005, September). The high-level protocol specification language HLPSL developed in the EU project AVISPA. In Proceedings of APPSEM 2005 workshop (pp. 1–17).

52. AVISPA, AVISPA web tool. http://www.avispa-project.org/web-interface/expert.php (accessed on October 2014).

53. Dolev D, Yao AC. On the security of public key protocols. IEEE Transactions on information theory. 1983; 29(2):198–208. doi: 10.1109/TIT.1983.1056650