PLOS ONE

RESEARCH ARTICLE

A robust anonymous biometric-based authenticated key agreement scheme for multi-server environments

Hua Guo[1☯], Pei Wang[1,2☯], Xiyong Zhang[3☯*], Yuanfei Huang[2], Fangchao Ma[4]

1 Beijing Key Laboratory of Network Technology, Beihang University, Beijing 100191, China, **2** National computer network and information security laboratory, National Computer network Emergency Response technical Team/Coordination Center, Beijing 100029, China, **3** State Key Laboratory of Space-Ground Integrated Information Technology, Beijing 100020, China, **4** Beijing information technology institute, Beijing 100094, China

☯ These authors contributed equally to this work.
* 09384@buaa.edu.cn

Citation: Guo H, Wang P, Zhang X, Huang Y, Ma F (2017) A robust anonymous biometric-based authenticated key agreement scheme for multi-server environments. PLoS ONE 12(11): e0187403. https://doi.org/10.1371/journal.pone.0187403

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

Received: April 12, 2017

Accepted: September 25, 2017

Published: November 9, 2017

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

Funding: This research was supported by the National Natural Science Foundation of China (No. 61300172, 61572027, 61402037), http://www.nsfc.gov.cn/. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

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

Abstract

In order to improve the security in remote authentication systems, numerous biometric-based authentication schemes using smart cards have been proposed. Recently, Moon *et al.* presented an authentication scheme to remedy the flaws of Lu *et al.*'s scheme, and claimed that their improved protocol supports the required security properties. Unfortunately, we found that Moon *et al.*'s scheme still has weaknesses. In this paper, we show that Moon *et al.*'s scheme is vulnerable to insider attack, server spoofing attack, user impersonation attack and guessing attack. Furthermore, we propose a robust anonymous multi-server authentication scheme using public key encryption to remove the aforementioned problems. From the subsequent formal and informal security analysis, we demonstrate that our proposed scheme provides strong mutual authentication and satisfies the desirable security requirements. The functional and performance analysis shows that the improved scheme has the best secure functionality and is computational efficient.

1 Introduction

Nowadays security has becoming an urgent issue for the distributed networks. The remote user authentication scheme allows the transmission of secret data via public channels, thus is an important cryptographic tool for distributed networks. In 1981, Lamport [1] proposed the first password-based authentication scheme. After that, considerable amount of work on password-based authentication schemes have been put forward for different applications [2, 3]. However, passwords are vulnerable to be broken in a short time by using dictionary guessing attack. To solve this problem, smart cards with password-based authentication schemes [4–12] are introduced to enhance the security of user authentication. Unfortunately, there are still some problems when the smart card is stolen and the stored data is leaked [13–15].

The biometric keys, such as fingerprint and iris, are considered to be a unique identifier of a user, thus have many advantages. For example, the biometric keys cannot be forgotten

or lost, are difficult to copy or share, and are not easy to forge or guess. Additionally, one can carry biometric keys at anytime and from anywhere. With the security requirements of the distributed networks and the good security performance and advantages of the biological characteristic, biometrics authentication protocols come to be more crucial and widely deployed [16–36]. In 2002, Lee *et al.* [16] designed the first biometrics-based remote user authentication scheme. In 2004, Lin-Lai [17] demonstrated that Lee *et al.*'s scheme cannot resist impersonation attack and designed a protocol without verification table to fix the flaws of Lee *et al.*'s scheme. In 2007, Khang-Zhang [18] pointed out that Lin-Lai's scheme is insecure against server spoofing attack and illustrated an improved scheme. Rhee [19] demonstrated that Khang-Zhang's scheme is vulnerable to impersonation attack and offline password guessing attack. Later, Li-Wang [20] designed an efficient three-factor remote user authentication scheme which only uses symmetric cryptographic primitive and the hash operation. However, in 2011, Das [21] exhibited that Li-Wang's scheme is insecure against man-in-the-middle attack and does not provide proper certification. Furthermore, he designed a new certification scheme based on biometric characteristics. In 2014, Li *et al.* [25] pointed out that Das *et al.*'s scheme is vulnerable to forgery attack and stolen smart card attack, and put forward a three-factor remote user authentication scheme. After that, Chaturvedi *et al.* [26] demonstrated that Li *et al.*'s scheme doesn't resist known session specific temporary information attack and doesn't protect user's privacy. They also proposed a novel authentication and key agreement protocol to overcome the weaknesses of Li *et al.*'s scheme.

In 2014, Chuang-Chen [27] proposed an efficient lightweight three-factor authentication protocol for multi-server environment which requires only the hash operation. After that, Mishra *et al.* [28] showed that Chuang-Chen's scheme is insecure against the denial-of-service attack, smart card stolen attack, server spoofing attack and impersonation attack. In addition, they proposed a new biometric-based multi-server authentication protocol so as to overcome the weaknesses of Chuang-Chen's scheme. In 2015, Lu *et al.* [29] illustrated that Mishra *et al.*'s scheme is insecure against server spoofing attack and impersonation attack, and can not provide forward secrecy. They introduced two independent three-factor authentication schemes [29, 31] for multi-server architecture, and claimed that the improved scheme has strong security. Unfortunately, Moon *et al.* [30] showed that Lu *et al.*'s scheme [29] is vulnerable to outsider attack and user impersonation attack, and put forward an enhanced protocol which fixes the flaws of Lu *et al.*'s scheme.

Unfortunately, we found that Moon *et al.*'s biometric-based remote user authentication scheme still has some flaws. In this paper, we firstly showed that Moon *et al.*'s scheme is vulnerable to insider attack, server spoofing attack, user impersonation attack and guessing attack. Moreover, we exhibited that their scheme is not anonymous for the user. Then we proposed an improved authentication scheme for multi-server environment to fix their design flaws. After that, we show that our scheme is robust against all known attacks through the formal and informal security analysis. Finally we demonstrate that the improved scheme has the best secure functionality and is computational efficient.

The rest of the paper is organized as follows. In section 2, we introduce some preliminary knowledge. Section 3 briefly reviews Moon *et al.*'s biometric-based remote user authentication scheme. Section 4 shows the design flaws in Moon *et al.*'s scheme. In order to eliminate the shortcomings discussed in section 4, we propose an enhancement authentication protocol in section 5. Section 6 analyzes the security of the proposed scheme, and Section 7 compares the performance of the enhanced scheme with other related schemes. Finally, we conclude in section 8.

2 Preliminaries

This section elaborates the definitions of one-way hash function and BioHashing, and the security model.

2.1 Definition

One-way hash function. A one-way hash function h: $\{0, 1\}^* \rightarrow \{0, 1\}^n$ takes an arbitrary-length input $x \in \{0, 1\}^*$, and produces a fixed-length output $h(x) \in \{0, 1\}^n$, called the message digest. The hash function has the following attributes:

- Computationally, it is easy to compute $y = h(x)$ if x and $h(\cdot)$ are specified.
- It is almost impossible through polynomial time t to know two inputs x_1 and x_2, such that $h(x_1) = h(x_2)$.

BioHashing. BioHashing technique [37] is designed to reduce the probability of denial of access while keeping the false acceptation performance. Inputing the biometric feature set and a seed which represents the "Hash key", BioHashing generates a vector of bits. More precisely, with the help of a uniform distributed pseudo-random numbers generated by giving a secret seed, the biometric vector data $x \in R^n$ is reduced down to a bit vector $b \in \{0, 1\}^l$ with l the length of the bit string ($l \leq n$) through BioHashing.

2.2 Security model

In this paper, we adopt the security model proposed by Abdalla et al. [38] to prove the security of our protocol.

- **Participants.** An oracle $\pi_{S_j}^t$ denotes an instance t of a party S_j, $\pi_{U_i}^u$ denotes the instance u of U_i, and π_{RS}^v denotes the instance v of RS.
- **Partnering.** The partner of an instance $\pi_{U_i}^u$ of U_i is the instance $\pi_{S_j}^t$ of S_j and conversely. The partial transcript of all exchanged messages between U_i and S_j is unique, and is said as a session ID sid_{Ui}^u for the present session in which $\pi_{U_i}^u$ participates.
- **Freshness.** $\pi_{S_j}^t$ or $\pi_{U_i}^u$ is fresh, only if the session key SK is not leaked to $\mathcal{A}$.
- **Adversary.** In the ROR model, $\mathcal{A}$ models the real attack via the following oracle queries. To breach the security of the authentication protocol, $\mathcal{A}$ is able to access the queries given below:
  - $Execute(\pi^t, \pi^u)$: The *Execute* query helps $\mathcal{A}$ obtain the messages transmitted between two honest participants; this query models an eavesdropping attack.
  - $Send(\pi^t; x)$: The *Send* query corresponds to an active attack. π^t executes the protocol and responds with an outgoing message after receiving a message x from $\mathcal{A}$.
  - $Reveal(\pi^t)$: The $\mathcal{A}$ executes *Reveal* query to reveal of session keys. If the session has been accepted, π^t returns the session key SK as its response that is computed between π^t and its partner, otherwise returns a null value.
  - $CorruptSC(\pi^t)$: It is about modeling smart card loss attack and outputs the information stored in SC_i.
  - $Test(\pi^t)$: At some point, the adversary $\mathcal{A}$ can make a Test query to an oracle Π^t. Π^t flips an unbiased coin b and responds with the real agreed session key SK if SK is established

and fresh, if $b = 1$; otherwise it returns a random sample generated according to the distribution of the session key. Otherwise, it returns $\perp$.

Semantic security of the session key. In an experiment, the adversary $\mathcal{A}$ is challenged to differentiate between an instance's real session key SK and a random key. $\mathcal{A}$ can continue querying *Test* queries to either the server instance or the user instance. The outcome of *Test* query must be consistent with the random bit b. Eventually, $\mathcal{A}$ terminates the game simulation and outputs a bit b' for b. we say $\mathcal{A}$ wins if the adversary guesses the correct b.

Let E denotes the event that $\mathcal{A}$ wins the game. Then, the advantage of $\mathcal{A}$ breaches the semantic security of our proposed authenticated key-agreement (AKE) protocol, say $\mathcal{P}$, is computed as $Adv_{\mathcal{P}}^{ake}(\mathcal{A}) = |2pr[E_0] - 1|$. We say that the protocol $\mathcal{P}$ is a secure multi-server authentication and key agreement protocol in the ROR sense if $Adv_{\mathcal{P}}^{ake}$ is negligible.

Random oracle. To prove the security of the proposed protocol, the one-way hash function $h(\cdot)$ is treated as a random oracle(say Hash oracle), and is provided to the adversary $\mathcal{A}$ and every participant. The Hash oracle is simulated by a two-tuple (u, v) table of binary strings. When a hash query $h(u)$ is made, the *Hash* oracle returns v if u is found in the table; otherwise, it returns a uniformly random string v and stores the pair (u, v) in the table.

3 Review of Moon *et al.*'s scheme

In this section, we briefly review Moon *et al.*'s scheme, which consists of four phases: registration phase, login phase, authentication phase and password change phase. Table 1 summarizes the notations used in this paper.

3.1 Registration phase

The registration and authentication phases are shown in Fig 1. In order to get the access to different services provided by the servers, a user must register himself through the registration server. U_i firstly selects an identity ID_i and password PW_i and inputs biometrics BIO_i.

Table 1. Notations.

Notations	Description
U_i	An i_{th} user
AS	Application server
RS	Registration server
ID_i	Identity of U_i
PW_i	Password of U_i
SC	smart card
SID_j	Identity of AS
PSK	Secret keys chosen by RS for AS
$E\{\}$, $D\{\}$	Encryption and decryption operations
P_{ub_s}, P_{ri_s}	Public and private keys of AS
n_1, n_2	Random numbers chosen by U_i and AS
$h(\cdot)$	A secure one-way hash function
$H(\cdot)$	A bio-hash function
$\oplus$	An exclusive-OR operation
$\|\|$	The concatenation operation

Table 1 summarizes the notations used in this paper.

https://doi.org/10.1371/journal.pone.0187403.t001

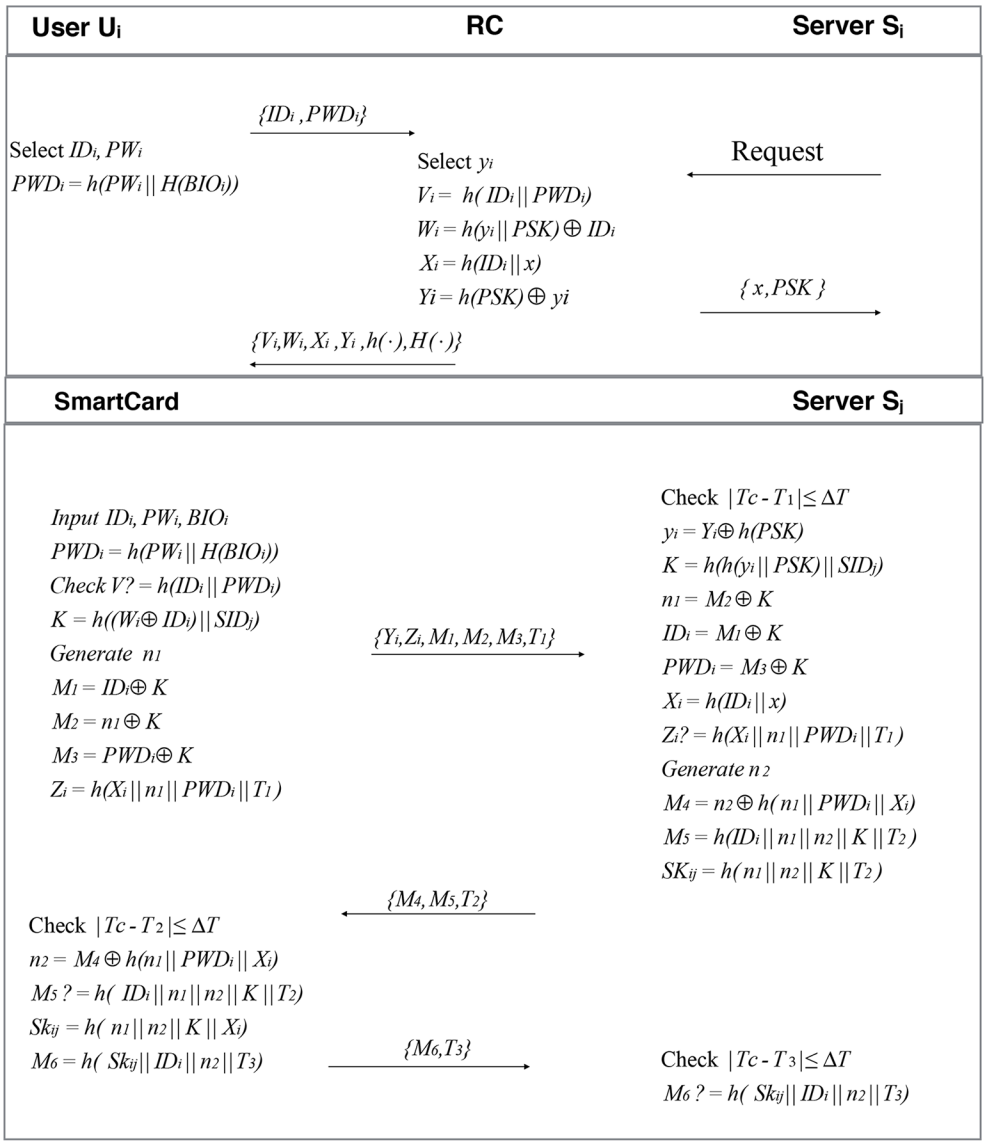

Fig 1. Registration and authentication phases of Moon *et al.*'s scheme. Registration and authentication phases of Moon *et al.*'s scheme.

https://doi.org/10.1371/journal.pone.0187403.g001

1. Using the password and the biometrics, the smart card computes $PWD_i = h(PW_i||H(BIO_i))$ and sends $< ID_i, PWD_i >$ to the registration server through a secure channel.
2. Upon receiving the message $< ID_i, PWD_i >$, the registration server computes $V_i = h(ID_i||PWD_i)$, $W_i = h(y_i||PSK) \oplus ID_i$, $X_i = h(ID_i||x)$, $Y_i = y_i \oplus h(PSK)$. Then RS stores $< V_i, W_i, X_i, Y_i, h(\cdot), H(\cdot) >$ onto a smart card and sends the smart card to U_i.

3.2 Login phase

During the login phase, the user U_i inserts his smart card into the smart card reader, inputs his identity ID_i and password PW_i, and imprints biometric information BIO_i. Upon receiving an input, the smart card uses the following steps to perform a login session:

1. The smart card computes $PWD_i = h(PW_i||H(BIO_i))$ and verifies $V_i? = h(ID_i||\ PWD_i)$. If succeeds, it executes the next step. Otherwise the session aborts.
2. The smart card generates a random number n_1 and computes $K = h((W_i \oplus ID_i)||\ SID_j)$, $M_1 = ID_i \oplus K$, $M_2 = n_1 \oplus K$, $M_3 = PWD_i \oplus K$, $Z_i = h(X_i||n_1||\ PWD_i||T_i)$.
3. The smart card transmits the login request message $< Y_i, Z_i, M_1, M_2, M_3, T_1 >$ to the server S_j through a public channel, where T_1 is the current timestamp.

3.3 Authentication phase

After receiving the authentication request $< Y_i, Z_i, M_1, M_2, M_3, T_1 >$ from the user U_i, the server S_j executes the following steps to authenticate each other.

1. The server S_j firstly checks whether $|T_c - T_1| < \Delta T$, then uses its pre-shared key PSK and achieves $y_i = Y_i \oplus h(PSK)$. The server also retrieves $K = h(h(y_i||PSK)||SID_j)$, $n_1 = M_2 \oplus K$, $ID_i = M_1 \oplus K$, $PWD_i = M_3 \oplus K$, $X_i = h(ID_i||x)$ and verifies $Z_i? = h(X_i||n_1||PWD_i||T_1)$. If they are not equal, S_j rejects the login request and terminates the session. Otherwise, the server generates a random number n_2 and computes $M_4 = n_2 \oplus h(n_1||PWD_i||X_i)$, $M_5 = h(ID_i||n_1||n_2||K||T_2)$, $SK_{ij} = h(n_1||n_2||K||T_2)$ and then responds with the message $< M_4, M_5, T_2 >$ to the smart card (user U_i) over a public channel.
2. Upon receiving the message $< M_4, M_5, T_2 >$ and checking the freshness of T_2, the smart card retrieves the value $n_2 = M_4 \oplus h(n_1||PWD_i||X_i)$. Then it verifies $M_5? = h(ID_i||n_1||n_2||K||T_2)$. If the verification holds, it computes the session key $SK_{ij} = h(n_1||n_2||K||X_i)$, which would be shared between U_i and S_j. Finally, the smart card computes $M_6 = h(SK_{ij}||ID_i||n_2||T_3)$ and sends the message $< M_6, T_3 >$ to S_j via a public channel.
3. Upon receiving the message $< M_6, T_3 >$, S_j checks the freshness of T_3 and verifies $h(SK_{ij}||ID_i||n_2||T_3)? = M_6$. If the equation holds, the server ensures the identity of U_i. Otherwise, the server aborts the session.

3.4 Password updating

In this phase, U_i can change his password any time when he wants. In order to change password, the user performs the following steps:

1. U_i inserts his smart card into the smart card reader and then inputs ID_i and PW_i and biometrics BIO_i.
2. The smart card SC_i computes $PWD_i = h(PW_i||H(BIO_i))$, then checks if $Vi' = h(ID_i||PWD_i)$ is the same as the stored V_i. If they are the same, SC_i accepts U_i to enter a new password $PW_{i_{new}}$.
3. SC_i computes $PWD_{i_{new}} = h(PW_{i_{new}}||H(BIO_i))$ and $V_{i_{new}} = h(ID_i||PWD_{i_{new}})$, and replaces V_i with $V_{i_{new}}$.

4 Security analysis of Moon *et al.*'s scheme

Although Moon *et al.* claimed that their scheme satisfies the required security requirements, we found that their scheme still has some weakness, i.e., fail to resist the insider attack, server spoofing attack, guessing attack and impersonation attack. Moreover, their scheme is not anonymous for users.

4.1 Lack of user anonymity

User anonymity means that the adversary cannot obtain or track the identity of the user according to the message transmitted via the public channel, which is an important property to protect the privacy of users. In Moon *et al.*'s scheme, during authentication phase, U_i sends $< Y_i, Z_i, M_1, M_2, M_3, T_1 >$ as authentication request message to S_j. Note that all the information transmitted in public channel can be intercepted by the adversary. The parameter $M_1 = K \oplus ID_i$ where $K = h((W_i \oplus ID_i)||SID_j))$ in the message $< Y_i, Z_i, M_1, M_2, M_3, T_1 >$, is unique and static for each user during all logins to the same server. Thus anyone has ability to track the activities of a legal user, if he captures the value of M_1.

4.2 Insider attack

Insider attack means that an insider can get the sensitive credentials from the information stored in *RS*. In Moon *et al.*'s scheme, during user registration phase, U_i submits his identity ID_i and PWD_i to *RS*. In order to prevent duplicate user registration, *RS* has to store the user's *ID*. If an adversary obtains the list of *ID*, it would cause great devastation. The adversary can impersonate himself as U_i as described in the following user impersonation attack.

4.3 Server spoofing attack

In Moon *et al.*'s protocol, *RS* shares the same secret information (x, *PSK*) with all the application severs. The compromised sever can impersonate as another legitimate server to deceive any legal user. Now we show the reason why Moon *et al.*'s scheme cannot withstand this kind of server spoofing attack.

1. When U_i submits his login request message $< Y_i, Z_i, M_1, M_2, M_3, T_1 >$ to S_j, the legal but malicious server S_k can intercept this message and compute $y_i = Y_i \oplus h(PSK)$, $K = h(h(y_i||PSK)||SID_j)$, $n_1 = M_2 \oplus K$, $ID_i = M_1 \oplus K$, $PWD_i = M_3 \oplus K$, $X_i = h(ID_i||x)$ and to check $Z? = h(X_i||n_1||PWD_i||T_1)$.
2. S_k generates a random number n_2 and computes $M_4 = n_2 \oplus h(n_1||PWD_i||X_i)$, $M_5 = h(ID_i||n_1||n_2||K||T_2)$, $SK_{ij} = h(n_1||n_2||K||T_2)$, then sends $< M_4, M_5, T_2 >$ to U_i.
3. U_i computes $n_2 = M_4 \oplus h(n_1||PWD_i||X_i)$, $M_5 = h(ID_i||n_1||n_2||K||T_2)$ and compares it with M_5. It is obvious that the values are the same, thus U_i responds with the message $M_6 = h(SK_{ij}||ID_i||n_2||T_3)$.
4. U_i computes the session key $SK_{ij} = h(n_1||n_2||K||T_2)$ and believes that he is communicating with S_j.

Therefore, a legal but malicious server S_k can masquerade as another server S_j to fool any legal user and Moon *et al.*'s scheme is vulnerable to server spoofing attack.

4.4 Guessing attack

Moon *et al.*'s scheme is vulnerable to identity guessing attack, which is a critical concern in their scheme. If the adversary can extract the secret value W_i from the legal user's smart card by some means and get the value of M_1 from public channel, the adversary can easily find out ID_i^* by performing the guessing attack, in which each guess ID_i can be verified as the following steps.

1. The adversary chooses ID_i^* and computes $K = h((W_i \oplus ID_i^*)||SID_j)$.
2. The adversary verifies the correctness of ID_i^* by checking $M_1? = ID_i^* \oplus K$.

3. The adversary repeats the above steps until a correct ID_i^* is found.

4.5 User impersonation attack

In a remote user communication scheme, anyone should be considered as a legal user if a user has valid authentication credentials or could be capable of constructing an effective authentication request message. In Moon *et al.*'s protocol, an adversary can impersonate a valid user as described below.

1. As enlightened in insider attack and guessing attack mentioned above, an adversary obtains U_i's personal identifiable information ID_i. He also extracts the secret values W_i and X_i from the legal user's smart card by some means.
2. The adversary intercepts a valid login request message $< Y_i, Z_i, M_1, M_2, M_3, T_1 >$ which is sent from ID_i via the public channel, then the adversary computes $K = ID_i \oplus M_1$, $PWD_i = K \oplus M_3$, chooses random number n_1, and calculates $M_{1m} = ID_i \oplus K$, $M_{2m} = n_1 \oplus K$, $M_{3m} = PWD_i \oplus K$, $Z_{im} = h(X_i||n_1||PWD_i||\ T_1')$. Now, the malicious adversary sends the forged login request message $< Y_i, Z_{im}, M_{1m}, M_{2m}, M_{3m}, T_1' >$ to S_j by masquerading as legal user U_i.
3. After the authentication of the login request message, the server S_j generates a random number n_2, computes $M_{4m} = n_2 \oplus h(n_1||PWD_i||X_i)$, $M_{5m} = h(ID_i||n_1\ ||n_2||K||T_2)$ and responds with the message $< M_{4m}, M_{6m}, T_2 >$ to the adversary who is masquerading as U_i.
4. The masquerading adversary verifies the correctness of M_{4m} with the values of n_1 and K. Then the masquerading user U_i computes $n_2 = M_{4m} \oplus h(n_1||PWD_i||X_i)$, $SK_{ij} = h(n_1||n_2||K||T_2)$, $M_{6m} = h(SK_{ij}||ID_i||n_2||T_3)$, and sends the message $< M_{6m}, T_3 >$ back to the server S_j.
5. The server S_j computes $M_{6m} = h(SK_{ij}||ID_i||n_2||T_3)$ and verifies it with the received value of M_{6m}. It is obvious that they are equal, so the sever authenticates successfully the legitimacy of the user U_i and the login request message information is accepted.
6. After mutual authentication, the server S_j and the malicious adversary who masquerades as the user U_i agree on the common session key as $SK_{ij} = h(n_1||\ n_2||K||X_i)$.

5 Our proposed scheme

In this section, we propose an improved remote user authentication scheme to fix the drawbacks in Moon *et al.*'s scheme. Our proposed protocol consists of four phases: registration, login, mutual authentication with key-agreement and password change. Fig 2 describes our proposed scheme.

5.1 Registration phase

When the remote user authentication scheme starts, the user U_i and the server S_j need to perform the following steps to register with the registration server(RS).

5.1.1 Server registration. To register with the system, a server S_j submits his identity SID_j and his public key Pub_j which can be obtained by all the users. Then S_j sends his identity SID_j and his public key Pub_j to RS. Upon reception, RS shares the secret key PSK with S_j and publishes S_j's public key Pub_j.

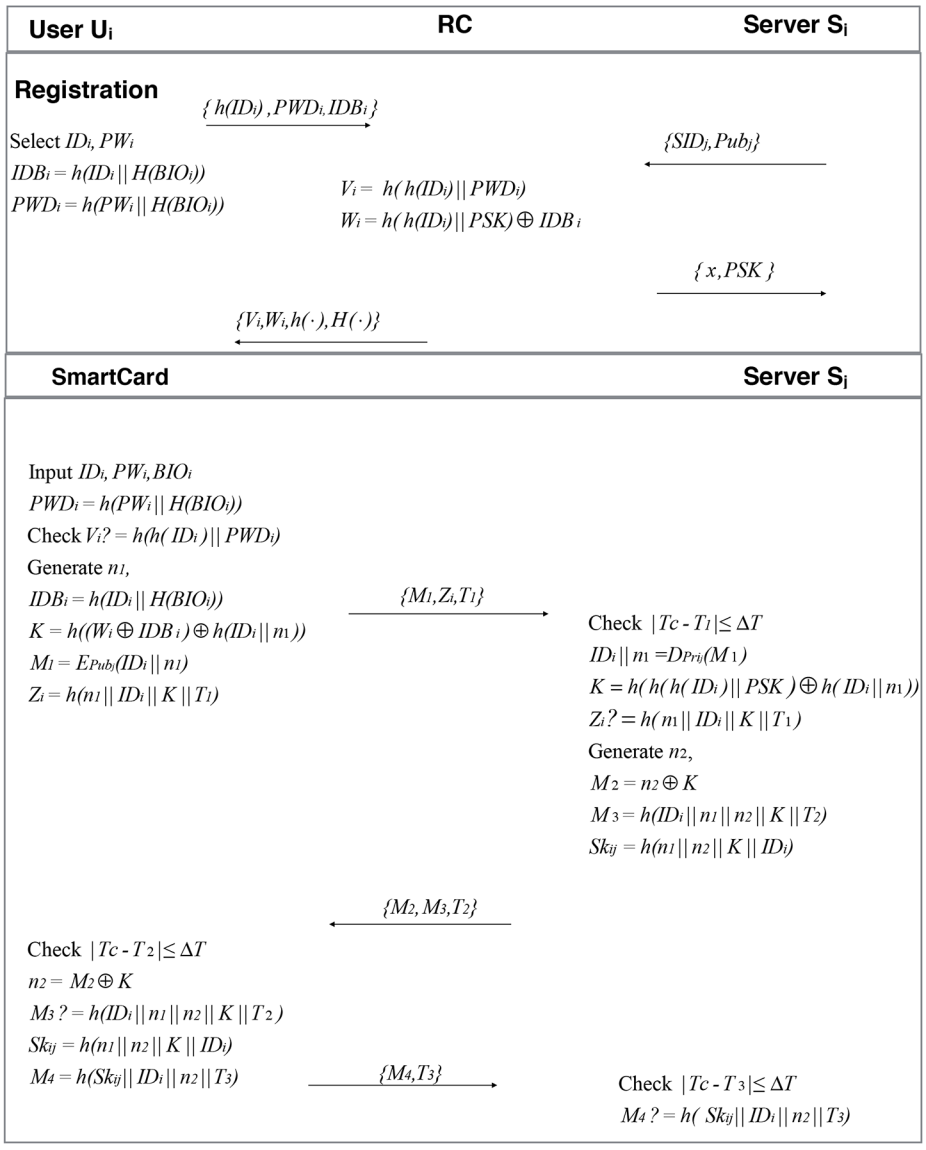

Fig 2. Registration and authentication phases of our scheme. Registration and authentication phases of our scheme.

https://doi.org/10.1371/journal.pone.0187403.g002

5.1.2 User registration.

1. U_i freely selects his identity ID_i which uniquely identifies the user's identity, password PW_i and scans his biometrics BIO_i. Then U_i computes $IDB_i = h(ID_i \,||H(BIO_i))$, $PWD_i = h(PW_i||\, H(BIO_i))$ and sends $< h(ID_i)$, IDB_i, $PWD_i >$ to RS on a secure channel.
2. Upon reception, RS computes $V_i = h(h(ID_i)||PWD_i)$, $W_i = h(h(ID_i)||PSK) \oplus IDB_i$ and stores $< V_i$, W_i, $h(\cdot)$, $H(\cdot) >$ in the smart card SC.
3. RS sends SC to U_i over a secure channel.

5.2 Login phase

1. U_i sends the login request by inserting smart card (SC), and inputting ID_i, PW_i and BIO_i.
2. SC computes $PWD_i = h(PW_i||H(BIO_i))$ and then checks whether the condition $V_i? = h(h(ID_i)||PWD_i)$. If the result is negative, the login session can be aborted. Otherwise, SC generates a random number n_1 and computes $K = h((W_i \oplus IDB_i) \oplus h(ID_i||n_1))$, $M_1 = E_{Pub_j}(ID_i||n_1)$, $Z_i = h(n_1||ID_i||K||T_1)$ and sends $< M_1, Z_i, T_1 >$ to the server S_j as the login request message.

5.3 Authentication phase

1. On getting login message, S_j checks freshness of T_1. S_j computes $(ID_i||n_1) = E_{Prij}(M_1)$, $K = h(h(h(ID_i)||PSK) \oplus h(ID_i||n_1))$ and verifies if $Z_i? = h(n_1||\ ID_i||K||T_1)$. If they are same, S_j authenticates U_i. Otherwise the session is terminated.
2. S_j further generates a random number n_2, and computes $M_2 = n_2 \oplus K$, $M_3 = h(ID_i||n_1||n_2||K||T_2)$, $SK_{ij} = h(n_1||n_2||K||ID_i)$. S_j sends $< M_2, M_3, T_2 >$ to SC.
3. On checking the freshness of T_2, SC computes $n_2 = M_2 \oplus K$ and verifies the condition $M_3? = h(ID_i||n_1||n_2||K||T_2)$. If the condition holds, U_i authenticates S_j. Otherwise the process is terminated. Then, SC computes $SK_{ij} = h(n_1||n_2\ ||K||ID_i)$ and $M_4 = h(SK_{ij}||ID_i||n_2||T_3)$, then sends $< M_4, T_3 >$ to S_j.
4. S_j checks the freshness of T_3. S_j verifies $M_4? = h(SK_{ij}||ID_i||n_2||T_3)$ and reconfirms the authenticity of U_i. Now, U_i and S_j share with the computed session key $SK_{ij} = h(n_1||n_2||K||ID_i)$ for further communication.

5.4 Password changing phase

This procedure is invoked whenever a user (U_i) wants to update his password with a new password PWD_i^*, without through a private channel or communicating with RS.

1. U_i inserts smart card SC and inputs ID_i, PW_i and BIO_i.
2. SC computes $PWD_i = h(PW_i||H(BIO_i))$ and then verifies the condition $V_i? = h(ID_i||PWD_i)$. If the condition doesn't hold, the request can be dropped.
3. U_i chooses a new password PW_i^* and then computes $PWD_i^* = h(PW_i^*||\ H(BIO_i^*))$, $V_i^* = h(h(ID_i)||PWD_i^*)$. Thus the smart card finally contains the parameters $\{V_i^*, W_i, h(\cdot), H(\cdot)\}$.

6 Security analysis of the proposed scheme

In this section, we use Burrows-Abadi-Needham logic (BAN-logic) [39] to verify the completeness of our scheme, then we prove the security of the scheme through formal and informal analysis.

6.1 Verifying the proposed scheme with BAN logic

The BAN logic introduced by Burrows *et al.* is a formal method of analyzing the security features of the information exchange protocol. It helps determine whether the exchanged information is credible, whether it can prevent eavesdropping or both. In this paper, we use BAN

Table 2. BAN logic notations.

Notations	Description
$P\mid\equiv X$	P believes the statement X is true
$P \lhd X$	P sees X
$P\mid\sim X$	P once said that X or has sent a message containing X
$P \Rightarrow X$	P has control over X
$\#X$	X is fresh
$P \overset{K}{\longleftrightarrow} Q$	P and Q can communicate using the shared key K, only P, Q or a trusted third party know K
$(X)_k$	The formula X is hashed by K
$\{X\}_k$	The formula X is encrypted by K
$\overset{K}{\longrightarrow} S_j$	K is the public key of P, only P know the corresponding secret key K^{-1}

https://doi.org/10.1371/journal.pone.0187403.t002

logic to prove that a user and a server share a session key after successfully running the protocol. We first introduce the BAN logic notations used in this paper in Table 2.

1. BAN logical postulates
    1. Message-meaning rule: $\frac{P\mid\equiv P\overset{K}{\longleftrightarrow}Q, P\lhd\{X\}_K}{P\mid\equiv Q\mid\sim X}$: If P believes that K is the shared key of P and Q, and P receives the message X encrypted with K, then P believe that Q has sent message X.
    2. Jurisdiction rule: $\frac{P\mid\equiv Q\Rightarrow X, P\mid\equiv Q\mid\equiv X}{P\mid\equiv X}$: If P believes that Q has the right to control X and P believes that Q also trusts X, then P trusts X.
    3. Nonce-verification rule: $\frac{P\mid\equiv\#(X), P\mid\equiv Q\mid\sim X}{P\mid\equiv Q\mid\equiv X}$: If P believes that X is fresh and P believes that Q has sent X, then P believes that Q believes X.
    4. Freshness-conjuncatenation rule: $\frac{P\mid\equiv\#(X)}{P\mid\equiv\#(X,Y)}$: If P believes that X is new, then the information of (X, Y) is also fresh.
    5. Belief rule: $\frac{P\mid\equiv X, P\mid\equiv Y}{P\mid\equiv(X,Y)}$: If P believes X and Y, then P believes (X, Y).
2. Establishment of security goals

    g1: $S_j\mid\equiv U_i\mid\equiv U_i \overset{SK_{ij}}{\longleftrightarrow} S_j$

    g2: $S_j\mid\equiv U_i \overset{SK_{ij}}{\longleftrightarrow} S_j$

    g3: $U_i\mid\equiv S_j\mid\equiv U_i \overset{SK_{ij}}{\longleftrightarrow} S_j$

    g4: $U_i\mid\equiv U_i \overset{SK_{ij}}{\longleftrightarrow} S_j$
3. Initiative premises

    p1. $U_i\mid\equiv\#n_1$. p2. $U_i\mid\equiv S_j\Rightarrow\#n_2$.

    p3. $S_j\mid\equiv\#n_1$. p4. $S_j\mid\equiv\#n_2$.

    p5. $S_j\mid\equiv U_i \overset{K}{\longleftrightarrow} S_j$. p6. $U_i\mid\equiv U_i \overset{K}{\longleftrightarrow} S_j$.

    p7. $U_i\mid\equiv ID_i$. p8. $S_j\mid\equiv U_i\Rightarrow ID_i$.

p9. $S_j| \equiv U_i \Rightarrow U_i \overset{SK_{ij}}{\longleftrightarrow} S_j$. p10. $U_i| \equiv S_j \Rightarrow U_i \overset{SK_{ij}}{\longleftrightarrow} S_j$.

4. Scheme analysis

a_0. $S_j \lhd \{n_1, ID_i\}_{Pub_j}$

Since $\overset{Pri_j}{\longrightarrow} S_j$, only S_j can get the value of ID_i and n_1. One can get the value of K unless he has the true Pri_j and PSK at the same time.

a_1. $S_j \lhd (n_1, ID_i, T_1)_K, T_1$

We employ Message-meaning rule according to p_5 and a_1 to drive:

a_2. $S_j| \equiv U_i| \sim (n_1, ID_i, T_1)$

According to a_2 and p_3, we apply the Freshness-conjuncatenation rule and Nonce-verification rule to get the following information:

a_3. $S_j| \equiv U_i| \equiv (n_1, ID_i, T_1)$

According to a_3 and p_8, we employ Jurisdiction rule and belief rule to obtain:

a_4. $S_j| \equiv ID_i$

According to a_4 and $S_j \lhd (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j, n_2, T_3)_{ID_i}, T_3$, we employ Message-meaning rule to obtain:

a_5. $S_j| \equiv U_i| \sim (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j, n_2, T_3)$

According to a_5 and p_4, we apply Nonce-verification rule and Freshness- conjuncatenation rule to obtain:

a_6. $S_j| \equiv U_i| \equiv (U_i \overset{SK_{ij}}{\longleftrightarrow} S_j, n_2, T_3)$

Finally, we employ The belief rule to obtain:

g_1. $S_j| \equiv U_i| \equiv U_i \overset{SK_{ij}}{\longleftrightarrow} S_j$.

According to g_1 and p_9, we utilize Jurisdiction rule to obtain:

g_2. $S_j| \equiv U_i \overset{SK_{ij}}{\longleftrightarrow} S_j$.

According to p_6 and $U_i \lhd (ID_i, n_1, n_2, T_2)_K$, we employ Message-meaning rule to obtain:

a_7. $U_i| \equiv S_j| \sim (ID_i, n_1, n_2, T_2)$

According to a_7 and p_1 we apply Nonce-verification rule and Freshness- conjuncatenation rule to derive:

a_8. $U_i| \equiv S_j| \equiv (ID_i, n_1, n_2, T_2)$

According to a_8 and p_1, p_3, p_4, p_6 and $SK_{ij} = h(n_1||n_2||K||ID_i)$, we apply Freshness-conjuncatenation rule and Nonce-verification rule to derive:

g_3. $U_i| \equiv S_j| \equiv U_i \overset{SK_{ij}}{\longleftrightarrow} S_j$.

According to g_3 and p_{10} we utilize Jurisdiction rule to obtain:

g_4. $U_i| \equiv U_i \overset{SK_{ij}}{\longleftrightarrow} S_j$.

6.2 Formal analysis

We use provable security to prove the security of our scheme. The security proof is based on the model of RSA-based password authentication.

Theorem 1. Let $\mathcal{A}$ be an adversary that run in polynomial time t against our protocal $\mathcal{P}$ in the random oracle, D be a uniformly distributed password dictionary and l denotes the

number of bits in the biometric key BIO_i, $|Hash|$ and$|D|$ denotes the range space of hash function and the size of D, respectively. If an attacker $\mathcal{A}$ makes q_h *Hash* queries, q_{send} *Send* queries, then, the advantage of $\mathcal{A}$ of breaking the SK-security of $\mathcal{P}$ is $Adv_{\mathcal{P}}^{ake} \leq \frac{q_h^2}{|Hash|} + \frac{q_{send}}{2^{l-1}.|D|} + 2Adv^{RSA}(t)$, where $Adv^{RSA}(t)$ is the advantage that an adversary $\mathcal{A}$ solves the problem about the factor decomposed of great number.

Proof. The proof is finished by executing a sequence of hybrid games G_i. For each game G_i, let E_i denote the event that the adversary succeeds in guessing the bit b in game G_i.

Game G_0: This game corresponds to the real attack in the random oracle model. Thus, we can write

$$Adv_{\mathcal{P}}^{ake} = |2pr[E_0] - 1| \tag{1}$$

Game G_1: By querying *Execute* oracle, this game simulates $\mathcal{A}$'s eavesdropping attack. After that, the adversary queries *Test* oracle, and decides whether the outcome of the *Test* oracle is the real session key SK or a random number, where SK_{ij} is computed from $SK_{ij} = h(n_1||n_2||K||ID_i)$. Note that PSK and IDB_i are secret to S_j and U_i. The adversary has no knowledge about PSK, IDB_i and ID_i, thus eavesdropping of message can not increase the chance of winning for the adversary in G_1. So we have

$$pr[E_0] = pr[E_1] \tag{2}$$

Game G_2: The difference between G_2 and G_1 is that we add the simulations of the *Send* and the *Hash* oracles. G_2 models an active attack where $\mathcal{A}$ tries to decide a participant into accepting a forged message. $\mathcal{A}$ can make several *Hash* queries to find the collisions. Note that the messages $\{M_1, Z_i, T_1\}$ and $\{M_2, M_3, T_2\}$ are associated with timestamp T_1, T_2, random numbers n_1 and n_2, and ID_i of U_i, hence there is no collision when querying the *Send* oracle. According to the birthday paradox, we have

$$|pr[E_2] - pr[E_1]| = \frac{q_h^2}{2.|Hash|} \tag{3}$$

Game G_3: In this game, G_3 simulates the *CorruptSC* oracle which models the smart card lost attack. Since the chosen password has low entropy, $\mathcal{A}$ may try online dictionary attack with the information obtained from the smart card. In addition, $\mathcal{A}$ may try to obtain biometrics key B_i from information collected from the smart card SC_i. Our protocol $\mathcal{P}$ uses BioHash, which extracts at most l nearly random bits, therefore the probability of guessing biometric key $B_i \in \{0, 1\}^l$ by $\mathcal{A}$ is approximated as $\frac{1}{2^l}$. If the number of wrong password inputs is limited by the system, probabilities can be estimated as follows:

$$|pr[E_3] - pr[E_2]| \leq \frac{q_{send}}{2^l.|D|} \tag{4}$$

Game G_4: This game models an attack wherein $\mathcal{A}$ has to compute the real session key $SK_{ij} = h(n_1||n_2||K||ID_i)$ using K, ID_i from the eavesdropping messages $\{M_1, Z_i, T_1\}$ and $\{M_2, M_3, T_2\}$. $\mathcal{A}$ can not compute $K = h((W_i \oplus IDB_i) \oplus h(ID_i||n_1))$ and $(ID_i||n_1) = E_{Pri_j}(M_1)$ as ID_i, Pri_j and IDB_i are unknown. $\mathcal{A}$ also needs to derive n_1 and n_2 from M_1 and M_2, respectively. We then have

$$|pr[E_4] - pr[E_3]| \leq Adv^{RSA}(t) \tag{5}$$

Additionally, since all session keys are random and independent and no information about the value of c is revealed to $\mathcal{A}$, Then,

$$pr[E_4] = \frac{1}{2} \tag{6}$$

From Eqs (1)–(6), the following result is obtained:

$$Adv_p^{ake} \leq \frac{q_h^2}{|Hash|} + \frac{q_{send}}{2^{l-1}.|D|} + 2Adv^{RSA}(t) \tag{7}$$

6.3 Informal security analysis

This subsection describes the security analysis of our scheme. To evaluate the security of the improved scheme, we assume that the adversary might access the smart card of legal user and extract the information stored in the smart card and intercept information transmitted over the public channel.

6.3.1 Mutual authentication. After receiving the login request information from U_i, S_j checks if $Z_i? = h(n_1||ID_i||K||T_1)$ holds or not. The adversary who masquerades as the legal user cannot forge Z_i without knowing ID_i and the biometrics BIO_i of U_i. Likewise, upon receiving the message M_3, U_i checks $M_3? = h(ID_i||n_1||n_2||K||T_2)$, where $K = h(h(h(ID_i)||PSK) \oplus h(ID_i||n_1))$, which requires the computation of U_i's identity ID_i, the random number n_1 and PSK. Only the server who has the private key Pri_j can compute ID_i and n_1 so as to get the value of K. Hence only legal user can share the session key with corresponding server. Therefore, our proposed scheme can provide proper mutual authentication.

6.3.2 Anonymity. In the proposed scheme, the login request message $< M_1, Z_i, T_1 >$ is dynamic for every login and does not disclose any information about U_i, since it is associated with random number n_1. The identity is protected by the encrypted message $M_1 = E_{pub_j}(ID_i||n_1)$ using Pub_j. The adversary cannot obtain ID_i without having the knowledge of Pri_j. In addition, the unauthorized server cannot decrypt the user's authentication message successfully since it does not own the private key Pri_j. As a result, the user's real identity cannot be retrieved. Thus our protocol can achieve the anonymity property of users as well as protect the privacy of users.

6.3.3 Off-line password guessing attack. An adversary may try to guess the password PW_i from the extracted smart card stored parameters $< V_i, W_i, h(\cdot), H(\cdot) >$. The stored parameter contains the password PW_i in the form $V_i = h(h(ID_i)||PWD_i)$ where $PWD_i = h(PW_i||H(BIO_i))$. An adversary attempts to verify the condition $V_i? = h(h(ID_i)||h(PW_i||H(BIO_i))$ while constantly guessing PW_i. Adversary needs the value of ID_i and BIO_i of U_i in order to achieve the password guessing attack. However, the value of BIO_i is nowhere stored and an adversary cannot get the value of ID_i without knowing the private key Pri_j. As a result, the adversary cannot guess the correct password PW_i. Therefore, our proposed improved protocol can withstand this kind of attack.

6.3.4 Insider attack. In our proposed protocol, U_i does not send his ID_i, password PW_i or his biometrics BIO_i in plain text during user registration phase. U_i submits only $h(ID_i)$, IDB_i and PWD_i to RS instead of original credentials, where $PWD_i = h(PW_i||H(BIO_i))$, $IDB_i = h(ID_i||H(BIO_i))$. Hence, an insider cannot obtain the original sensitive information of any user. On the other hand, the authentication of entities is being done by verifying message like $Z_i? = h(n_1||ID_i||K||T_1)$ in which ID_i is necessary. Moreover, RS doesn't participate in the authentication process. Therefore, the proposed protocol attains resistance to insider attack.

6.3.5 Stolen smart card attack. The adversary can extract the information $< V_i, W_i, h(\cdot), H(\cdot) >$ stored in the smart card by means of power analysis. Assume a legal user's smart card is

stolen by an adversary and the stored information $< V_i, W_i, h(\cdot), H(\cdot) >$ on it are extracted. Then, the adversary may try to get ID_i, PW_i, BIO_i from the extracted information. However, adversary cannot obtain any valuable information from these values, where $V_i = h(h(ID_i)||PWD_i)$ and $W_i = h(h(ID_i)||PSK) \oplus IDB_i$, since all the important parameters such as ID_i and PW_i are protected by a one-way hash function. The adversary cannot obtain any login information using the smart card stored parameters V_i and W_i. At the same time guessing the real identity ID_i and password PW_i is impractical. Therefore, the proposed protocol is secure against smart card stolen attack.

6.3.6 Replay attack. If an adversary has intercepted all the communication message $< M_1, Z_i, T_1 >$ and $< M_2, M_3, T_2 >$, he tries to replay them to U_i or S_j to masquerade as a legal user. However, once the message is replayed, the server can immediately detect the attack and reject the request due to the apply of timestamp. Hence, our scheme is secure against replay attack.

6.4 No verification table

In the proposed scheme, the registration server and application servers do not store the password and the biometrics database of the user. Therefore, even if an adversary steals the information stored in *RS*, he still cannot get ID_i, PW_i, BIO_i or other valid information of users. S_j does not store the password or the biometrics table of users as well. Therefore, even if an adversary steals the database from *RS*, he still cannot obtain user's sensitive information of users.

6.4.1 User masquerade attack. Assume an adversary steals a smart card from a legal user and wants to get service by perpetrating user impersonation attack. If an adversary forges messages so as to impersonate as U_i, he needs to build a login request message $< M_1, Z_i, T_1 >$ firstly, where $M_1 = E_{Pub_j}(ID_i||n_1)$, $Z_i = h(n_1||ID_i||K||T_1)$. Conversely, the adversary cannot compute the messages M_1 and Z_i without user's private information ID_i and $H(BIO_i)$. At the same time, the adversary has to go through login phase before sending login request information. During login phase, *SC* computes $PWD_i = h(PW_i||H(BIO_i))$ and then verifies if $V_i? = h(ID_i||PWD_i)$ is correct. Unless the adversary enters the correct credentials, the process will be terminated. Therefore, the adversary certainly requires ID_i, PW_i and BIO_i for any furthermore computations. However, the probability of obtaining correct ID_i, PW_i and BIO_i is negligible.

6.4.2 Server impersonation attack. Unlike Moon *et al.*'s protocol, the server S_j not only keeps unique long-term key *PSK*, but also contains the key pair $< Pub_j, Pri_j >$. Note that the key pair of each server is distinctive, and Pri_j is known to only server S_j. Consider a scenario where an adversary captures $< M_1, Z_i, T_1 >$ and tries to impersonate valid server by responding with message $< M_2, M_3, T_2 >$. The values of ID_i, K and n_1 are prerequisite. However, adversary cannot yield either of the values without having the knowledge of Pri_j. Though, the adversary cannot get the right values of ID_i, K and n_1, if the adversary forges the massage $< M_2, M_3, T_2 >$. Upon receiving the response message $< M_2, M_3, T_2 >$, U_i can identify it as a malicious attempt due to the non-equivalence of message $M_3'? = M_3$. Thus, our proposed protocol is secure against server impersonation attack.

6.4.3 Forward secrecy. In our improved protocol, the session key is $SK_{ij} = h(n_1||n_2||K||ID_i)$, and the values of the long term private key of the servers vary from server to server and are not shared with any registered U_i. Assume that the adversary has obtained the long term key *PSK*, he still cannot compute a valid session key without the secret parameters ID_i and n_1, which are protected by Pub_j and are decryptable only with Pri_j. Moreover, the parameters n_1 and n_2 are random for each session. Therefore, the session key is considered to be safe even though the long term private key of the server is compromised.

Table 3. Functionality comparison.

Scheme	Chuang [27]	Mishra [28]	Lu [29]	Lu [31]	Moon [30]	our
Provide mutual authentication	No	Yes	Yes	Yes	Yes	Yes
User anonymity	Yes	Yes	No	No	No	Yes
Resist insider attack	Yes	Yes	Yes	Yes	No	Yes
Resist off-line guessing attack	Yes	Yes	Yes	Yes	No	Yes
Resist smart card theft attack	No	Yes	Yes	Yes	Yes	Yes
Resist replay attack	No	No	No	Yes	Yes	Yes
Resist Impersonation attack	No	No	No	No	No	Yes
Session key agreement	Yes	Yes	Yes	Yes	Yes	Yes
Provides Forward secrecy	Yes	No	Yes	Yes	Yes	Yes
Efficient password change phase	No	No	Yes	Yes	Yes	Yes
Resist verifier attack	Yes	Yes	Yes	Yes	Yes	Yes

https://doi.org/10.1371/journal.pone.0187403.t003

7 Functional and performance analysis

In this section, we compare our proposed scheme with the other related schemes in term of the functionality, including Chuang *et al.*'s scheme, Mishra *et al.*'s scheme and Lu *et al.*'s scheme.

7.1 Functional analysis

We perform a comparative analysis of previous schemes, which is illustrated in Table 3. From the table, we can find that the proposed scheme is more secure and provides more functionality requirements than the other related schemes. Moreover, the proposed scheme achieves all resistance requirements.

7.2 Performance analysis

Now we compare the computational costs and execution time between the proposed scheme and the other related schemes. For the evaluation of the computational costs, let T_h, T_{Re}, T_{Rd}, T_{sym} and T_{epm} refer to the execution time of one-way hash, *RSA* encryption, *RSA* decryption, symmetric key encryption/decryption operation and complexity of executing an elliptic curve point multiplication operation. According to Kilinc *et al.*'s [40] estimation, the average running time of T_h is about 0.0023ms, T_{Re} is 3.8500ms, T_{Rd} is 0.1925ms, T_{sym} is 0.1303 ms and T_{epm} is 2.229ms. Table 4 illustrates the comparative performance of our improved scheme and previously proposed schemes.

Table 4. Computation costs comparison.

Scheme	Login	Authentication	Total	Time(ms)
Chuang *et al.*'s [27]	$4T_h$	$13T_h$	$17T_h$	0.0391
Mishra *et al.*'s [28]	$4T_h$	$11T_h$	$15T_h$	0.0345
Lu *et al.*'s [29]	$6T_h$	$12T_h$	$18T_h$	0.0414
Moon *et al.*'s [30]	$5T_h$	$13T_h$	$18T_h$	0.0414
Lu *et al.*'s [31]	$4T_h + 3T_{Re}$	$14T_h + 3T_{Rd}$	$18T_h + 3T_{Re} + 3T_{Rd}$	12.1689
Mishra's [32]	$6T_h + 2T_{epm}$	$10T_h + 1T_{epm}$	$16T_h + 3T_{epm}$	6.7148
Chaudhry's [33]	$2T_h + 3T_{epm}$	$6T_h + 5T_{epm}$	$8T_h + 8T_{epm}$	17.8504
Jiang's [34]	$3T_h + 1T_{epm} + T_{sym}$	$6T_h + 3T_{epm} + 3T_{sym}$	$9T_h + 6T_{epm} + 4T_{sym}$	13.9159
our scheme	$7T_h + 1T_{Re}$	$11T_h + 1T_{Rd}$	$18T_h + T_{Re} + T_{Rd}$	4.0866

https://doi.org/10.1371/journal.pone.0187403.t004

The time consumption of our proposed scheme and of the other related schemes is listed in Table 4. The results shows that the proposed scheme is the most computationally inexpensive one among those schemes based on public key cryptography [31–34]. Note that although our proposed scheme costs more time than rest of the schemes [27–30], it is more secure than these schemes. To sum up, only the proposed scheme provides both the computation efficiency to accomplish mutual authentication and key agreement, and the basic security properties against the known threats. The rest of schemes either are vulnerable to various attacks [27–31], or need more time than our scheme [31–34].

8 Conclusion

In this paper, we firstly analyzed the security of Moon *et al*'s scheme, and demonstrated that their scheme is vulnerable to the known internal attack, guess attack and impersonation attack. Moreover, their scheme is found not anonymous for the user. To withstand these drawbacks, we proposed an improved biometric-based authentication scheme for multi-server environment and proved that the improved scheme provides secure authentication through the formal security analysis using Burrows-Abadi-Needham logic (BAN-logic) and random oracle model. Moreover, we have shown that our scheme is robust against all known attacks through the informal security analysis. The functional and performance analysis shows that the improved scheme has the best secure functionality and is computational efficient.

Author Contributions

Conceptualization: Hua Guo, Xiyong Zhang.

Formal analysis: Hua Guo.

Funding acquisition: Hua Guo.

Investigation: Pei Wang, Xiyong Zhang, Yuanfei Huang, Fangchao Ma.

Methodology: Pei Wang.

Resources: Yuanfei Huang, Fangchao Ma.

Supervision: Xiyong Zhang.

Writing – original draft: Pei Wang.

Writing – review & editing: Hua Guo.

References

1. Lamport L. Password authentication with insecure communication[J]. Communications of the Acm, 1981, 24(24):770–772. https://doi.org/10.1145/358790.358797
2. Harn L, Huang D, Laih C S. Password authentication using public-key cryptography[J]. Computers & Mathematics with Applications, 1989, 18(12):1001–1017. https://doi.org/10.1016/0898-1221(89)90028-X
3. Shieh S P, Yang W H, Sun H M. An authentication protocol without trusted third party[J]. IEEE Communications Letters, 1997, 1(3):87–89. https://doi.org/10.1109/4234.585805
4. Sun H M. An efficient remote use authentication scheme using smart cards[J]. IEEE Transactions on Consumer Electronics, 2000, 46(4):958–961. https://doi.org/10.1109/30.920446
5. Chien H Y, Jan J K, Tseng Y M. An Efficient and Practical Solution to Remote Authentication: Smart Card[J]. Computers & Security, 2002, 21(4):372–375. https://doi.org/10.1016/S0167-4048(02)00415-7
6. Das M L, Saxena A, Gulati V P. A dynamic ID-based remote user authentication scheme[J]. IEEE Transactions on Consumer Electronics, 2004, 50(2):629–631. https://doi.org/10.1109/TCE.2004.1309441

7. Islam S H. Design and analysis of an improved smartcard-based remote user password authentication scheme[J]. International Journal of Communication Systems, 2014, 29(11):n/a–n/a.
8. Song R. Advanced smart card based password authentication protocol[J]. 2010, 32(5-6):321–325.
9. Mishra D, Chaturvedi A, Mukhopadhyay S. Design of a lightweight two-factor authentication scheme with smart card revocation[J]. Journal of Information Security & Applications, 2015, 23(C):44–53. https://doi.org/10.1016/j.jisa.2015.06.001
10. Srinivas J, Mukhopadhyay S, Mishra D. A Self-Verifiable Password Based Authentication Scheme for Multi-Server Architecture Using Smart Card[J]. Wireless Personal Communications, 2017:1–25.
11. Maitra T, Islam S H, Amin R, Giri D, Khan M K, Kumar N. An enhanced multiserver authentication protocol using password and smart card: cryptanalysis and design[J]. Security & Communication Networks, 2016, 9(17):4615–4638. https://doi.org/10.1002/sec.1653
12. Amin R, Islam SH, Khan MK, Karati A, Giri D, Kumari S. A Two-factor RSA-based Robust Authentication System for Multi-Server Environments[J]. Security and Communication Networks. 2017, 13(1):74–84.
13. Messerges T S, Dabbish E, Sloan R H. Examining smart-card security under the threat of power analysis attacks[J]. IEEE Transactions on Computers, 2002, 51(5):541–552. https://doi.org/10.1109/TC.2002.1004593
14. Kocher P C, Jaffe J, Jun B. Differential Power Analysis[C]// International Cryptology Conference on Advances in Cryptology. Springer-Verlag, 1999:388-397.
15. Wang D, Wang P. Offline Dictionary Attack on Password Authentication Schemes Using Smart Cards [M]// Information Security. Springer International Publishing, 2015:1212-1217.
16. Lee J K, Ryu S R, Yoo K Y. Fingerprint-based remote user authentication scheme using smart cards[J]. Electronics Letters, 2002, 38(12):554–555.
17. Lin C H, Lai Y Y. A flexible biometrics remote user authentication scheme[J]. Computer Standards & Interfaces, 2004, 27(1):19–23. https://doi.org/10.1016/j.csi.2004.03.003
18. Khan M K, Zhang J. Improving the Security of 'A Flexible Biometrics Remote User Authentication Scheme'[J]. Computer Standards & Interfaces, 2007, 29(1):82–85. https://doi.org/10.1016/j.csi.2006.01.002
19. Rhee H S, Kwon J O, Lee D H. A remote user authentication scheme without using smart cards[J]. Computer Standards & Interfaces, 2009, 31(1):6–13.
20. Li C T, Hwang M S. An efficient biometrics-based remote user authentication scheme using smart cards [J]. Journal of Network & Computer Applications, 2010, 33(1):1–5. https://doi.org/10.1016/j.jnca.2009.08.001
21. Das A K. Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards[J]. Iet Information Security, 2011, 5(3):145–151.
22. Li X, Niu J W, Ma J, Wang WD, Liu CL. Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. Journal of Network and Computer Applications, 34(1), 73-79[J]. Journal of Network & Computer Applications, 2011, 34(1):73–79. https://doi.org/10.1016/j.jnca.2010.09.003
23. Li X, Niu J, Kumari S, Wu F, Wu F, Khan M K, et al. A Novel Chaotic Maps-Based User Authentication and Key Agreement Protocol for Multi-server Environments with Provable Security[J]. Wireless Personal Communications, 2016, 89(2):569–597. https://doi.org/10.1007/s11277-016-3293-x
24. Kumari S, Das A K, Li X, Wu F, Khan M K, Jiang Q, et al. A provably secure biometrics-based authenticated key agreement scheme for multi-server environments[J]. Multimedia Tools & Applications, 2017:1–31.
25. Li X, Niu J, Wang Z, Chen C. Applying biometrics to design three-factor remote user authentication scheme with key agreement[J]. Security & Communication Networks, 2014, 7(10):1488–1497.
26. Chaturvedi A, Mishra D, Jangirala S, Mukhopadhyayet S. A privacy preserving biometric-based three-factor remote user authenticated key agreement scheme[J]. Journal of Information Security & Applications, 2016.
27. Chuang M C, Chen M C. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics[J]. Expert Systems with Applications, 2014, 41(4):1411–1418. https://doi.org/10.1016/j.eswa.2013.08.040
28. Mishra D, Das A K, Mukhopadhyay S. A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards[J]. Expert Systems with Applications, 2014, 41(18):8129–8143. https://doi.org/10.1016/j.eswa.2014.07.004
29. Lu Y, Li L, Yang X, Yang Y. Robust Biometrics Based Authentication and Key Agreement Scheme for Multi-Server Environments Using Smart Cards[J]. Plos One, 2015, 10(5):e0126323 https://doi.org/10.1371/journal.pone.0126323 PMID: 25978373

30. 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.[J]. Plos One, 2015, 10(12): e0145263. https://doi.org/10.1371/journal.pone.0145263 PMID: 26709702

31. Lu Y, Li L, Peng H, Yang W. A biometrics and smart cards-based authentication scheme for multi-server environments[J]. Security & Communication Networks, 2015, 8(17):3219–3228. https://doi.org/10.1002/sec.1246

32. Mishra D. Design and Analysis of a Provably Secure Multi-server Authentication Scheme[J]. Wireless Personal Communications, 2016, 86(3):1–25. https://doi.org/10.1007/s11277-015-2975-0

33. Chaudhry S A. A secure biometric based multi-server authentication scheme for social multimedia networks[M]. Kluwer Academic Publishers, 2016.

34. Jiang Q, Khan M K, Lu X, Ma J, He D. A privacy preserving three-factor authentication protocol for e-Health clouds[J]. Journal of Supercomputing, 2016, 72(10):3826–3849. https://doi.org/10.1007/s11227-015-1610-x

35. Li X, Niu J, Kumari S, Wu F, Choo KK R. A robust biometrics based three-factor authentication scheme for Global Mobility Networks in smart city[J]. Future Generation Computer Systems, 2017.

36. Li X, Ibrahim M H, Kumari S, Sangaiah A K, Gupta V, Choo KK R. Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks[J]. Computer Networks, 2017. https://doi.org/10.1016/j.comnet.2017.03.013

37. Jin A T B, Ling D N C, Goh A. Biohashing: two factor authentication featuring fingerprint data and tokenised random number[J]. Pattern Recognition, 2004, 37(11):2245–2255. https://doi.org/10.1016/j.patcog.2004.04.011

38. Abdalla M, Fouque P A, Pointcheval D. Password-Based authenticated key exchange in the three-party setting[C]// International Workshop on Public Key Cryptography. Springer Berlin Heidelberg, 2005:65-84.

39. Burrows M, Abadi M, Needham R. A logic of authentication[J]. Proceedings of the Royal Society A Mathematical Physical & Engineering Sciences, 1990, 8(5):18–36.

40. Kilinc H H, Yanik T. A Survey of SIP Authentication and Key Agreement Schemes[J]. IEEE Communications Surveys & Tutorials, 2014, 16(2):1005–1023. https://doi.org/10.1109/SURV.2013.091513.00050