Cryptanalysis and Improvement of a Biometric-Based Multi-Server Authentication and Key Agreement Scheme

Chengqi Wang, Xiao Zhang*, Zhiming Zheng*

Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education, and School of Mathematics and Systems Science, Beihang University, Beijing 100191, China

* 09621@buaa.edu.cn (XZ); zzheng@pku.edu.cn (ZMZ)

Abstract

With the security requirements of networks, biometrics authenticated schemes which are applied in the multi-server environment come to be more crucial and widely deployed. In this paper, we propose a novel biometric-based multi-server authentication and key agreement scheme which is based on the cryptanalysis of Mishra et al.’s scheme. The informal and formal security analysis of our scheme are given, which demonstrate that our scheme satisfies the desirable security requirements. The presented scheme provides a variety of significant functionalities, in which some features are not considered in the most of existing authentication schemes, such as, user revocation or re-registration and biometric information protection. Compared with several related schemes, our scheme has more secure properties and lower computation cost. It is obviously more appropriate for practical applications in the remote distributed networks.

Introduction

With the rapid development of Internet, advances in the information and communication technology enhance the quality of online services for distributed networks, which provide the highly useful services to users in a variety of aspects, such as online medicine, online education, online shopping and internet banking [1, 2]. Also there is always interaction between users and servers over a public channel so that design and analysis of secure and efficient authentication scheme have received a considerable attention nowadays [3]. Since the first one was proposed by Lamport, a great number of authentication schemes have been presented, which provide authorized communication between remote entities [4–9]. According to the evidences adopted in the authentication, the existing schemes are divided into two categories: certificate-based and identity-based [10–16]. The former category requires the high computation cost and large storage space for the management of certificate store. Although elliptic curve cryptosystem is introduced, they do not simplify the certificate management so that certificate-based schemes are unacceptable in a real-time application such as multi-media and video conference. To solve the aforementioned problems, Shamir proposed an identity-based public key cryptosystem
But integer factorization problem applied in the Shamir’s scheme is difficult to be implemented efficiently [18]. And then some other identity-based schemes are presented, which are also based on the pairing operation or elliptic curve [19–24]. However, most of them are inefficient because of complicated structures [25–28]. Therefore, secure identity-based authentication schemes that only apply the random numbers and hash function are considered as optimum designs for mobile users and real-time applications.

Furthermore, there are some security vulnerabilities to identity-based authentication schemes in the compromise of passwords and tokens [29–35]. In particular, it is difficult to remember long and random passwords for users, and short passwords are easily broken by simple dictionary attacks because of low entropy. Also it is feasible to extract the information stored in the smart cards by side channel attacks, such as SPA or DPA [36]. To solve these problems, many researchers have combined the biometrics, passwords and tokens to enhance the security of authentication schemes [37–39]. The uniqueness of biometrics in the authentication scheme makes it difficult for adversary to forge the biometric information [40, 41]. And authentication does not request users to remember the biometrics. In fact, biometric characteristics imprinted by users are not exactly the same every time so that direct using them always results in low acceptance of valid users in the biometric-based authentication schemes [42]. Since the failure to authorized users significantly impacts on the availability of schemes, we introduce the fuzzy extractor to reduce the probability of rejection effectively, which is a convenient mechanism to be implemented in the smart card [43, 44].

Meanwhile, conventional authentication schemes are not suitable for the multi-server environment [45, 46]. When single server authenticated schemes are adopted in the multi-server environment, users not only login and access to different remote servers with repetitive registration, but also remember different information about identities and passwords for each server. It decreases the adoption of large network based on the applications. With the assistance of registration center, single registration helps the remote distributed system allow users to access the resources efficiently and conveniently, which is an important consideration in the multi-server architecture. Besides, authentication mechanism is required to achieve a higher level of security in the multi-server environment [47]. There are defects in many multi-server authentication schemes, since users apply the same identities and passwords to login the different servers [48–50]. It gives adversaries opportunities to trace legal users, which usually makes schemes vulnerable to insider attack, masquerade attack and server spoofing attack. For example, Chuang and Chen [51] proposed an anonymous multi-server authenticated key agreement scheme in 2014, and claimed that their scheme not only supported multiple servers but also achieved various security requirements. However, Choi et al. [52] pointed out that Chuang and Chen’s scheme was vulnerable to the smart card attack, user impersonation attack, masquerade attack, DoS attack, and did not achieve the perfect forward secrecy. To achieve the security and efficiency, an authentication scheme for the multi-server environment should meet the following requirements: 1) registration center should be avoided in the authentication phase to avoid the bottlenecks, 2) multiple secret keys in the smart card should not be required to reduce the storage requirement, 3) servers can be easily added on the later stage, and 4) all involved servers may not be trusted [3]. Thus, more work about authenticated key agreement schemes based on the multi-server needs to be studied.

Recently, a user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards is proposed by Mishra et al. [53], which is applicable for expert systems to achieve the anonymous authentication in multi-server environment. Expert systems have several applications such as security auditing and network management, which emulate or act in all respects with decision-making capabilities of human experts. And Mishra et al. claimed that their scheme satisfied the all security attributes. Unfortunately, according to
the cryptanalysis given in this paper, we identify that their scheme does not resist the masquerade attack, replay attack and Denial-of-Service (DoS) attack. We also find that their scheme fails to achieve the perfect forward secrecy. In addition, there is no consideration of the revocation or re-registration phase in the most of existing authentication schemes. To solve these problems, we propose a robust biometric-based multi-server authentication and key agreement scheme. Our scheme improves the Mishra et al.’s scheme and satisfies the desirable security requirements. Also presented scheme provides a variety of significant functionalities, such as anonymity, mutual authentication, session key agreement, perfect forward secrecy, user revocation or re-registration, and biometric information protection. In addition, comparison results show that our scheme has more secure properties, more functionalities and lower computation cost, which make our scheme more appropriate for practical applications in the remote distributed networks.

The remaining of the paper is organized as follows. Next section briefly introduces the threat assumptions, fuzzy extractor and one-way collision-resistant hash function which are adopted in our scheme. Section 3 reviews the Mishra et al.’s scheme. Section 4 mainly discusses the weaknesses of Mishra et al.’s scheme. Section 5 describes the proposed scheme in detail. And then section 6 provides the security, functionality and performance analysis of our algorithm. The last section gives the conclusion.

Preliminaries
In this section, we describe some concepts about threat assumptions, fuzzy extractor and one-way collision-resistant hash function, which are useful in our scheme.

Threat assumptions
In this paper, we introduce the Dolev-Yao threat model [54] and consider the risk of side-channel attacks [55] to construct the threat assumptions which are described as follows:

1. Adversary $E$ eavesdrops all the communications between user and server over a public channel.
2. Adversary $E$ modifies, deletes, resends and reroutes the eavesdropped messages.
3. Adversary $E$ may be a malicious user or an outsider in this system.
4. Adversary $E$ extracts the sensitive stored information from lost or stolen smart card by examining the power consumption.

Fuzzy extractor
The mechanism of fuzzy extractor consists of two procedures ($Gen$, $Rep$), which is illustrated in Fig 1.

The function $Gen$ is a probabilistic generation procedure, which extracts biometric input $BIO$, and outputs a nearly random binary string $R \in \{0, 1\}^l$ and an auxiliary binary string $P \in \{0, 1\}^l$. Also the function $Rep$ is a deterministic reproduction procedure allowing to recover $R$ with the assistance of corresponding auxiliary string $P$ and biometric $BIO^*$. If $dis(BIO, BIO^*) \leq t$ and $Gen(BIO) \rightarrow (R, P)$, then we have $Rep(BIO^*, P) = R$. Otherwise, there is no guarantee provided by function $Rep$. The error-tolerant makes it dependable to recover nearly uniform randomness $R$ with auxiliary string $P$ from biometric input $BIO^*$, as long as it remains reasonably close to original input $BIO$. More details about fuzzy extractor are described in the literature [43, 44].
One-way collision-resistant hash function

The one-way collision-resistant hash function \( h = h(x) : \{0, 1\}^* \rightarrow \{0, 1\}^n \) is a deterministic algorithm, which outputs a fixed-length binary string \( \{0, 1\}^n \) based on the arbitrary length binary string \( \{0, 1\}^* \) \[56\]. It is computationally infeasible to retrieve the input \( x \) from given hash value and hash function, which is called the one-way property. Also hash function possesses weak/strong collision resistant property. For a given input \( x \), finding any input \( y \neq x \) so that \( h(x) = h(y) \) is computationally infeasible. For a given pair of inputs \((x, y)\) with \( x \neq y \), then \( h(x) = h(y) \) is computationally infeasible. The well-known example of hash function is SHA-1. However, Manuel showed that SHA-1 is insecure against the collision attacks in 2011 \[57\]. So we apply the SHA-2 as secure hash function in our scheme.

Review of Mishra et al.’s scheme

Recently, Mishra et al. proposed a biometric-based multi-server key agreement scheme using smart cards to achieve the light-weight authentication and user anonymity. There are five phases relating to Mishra et al.’s scheme, which are the server registration phase, user registration phase, login phase, authentication phase and password change phase, respectively. Suppose that RC is the trusted third party, which is responsible for the registration of users and servers. Table 1 lists the notations used in their scheme.

Server registration phase

1. The server \( S_j \) sends a join message to the \( RC \).
2. After receiving the join message, \( RC \) replies with the pre shared key (PSK) to the server \( S_j \) through a secure channel.
Upon receiving the $PSK$, the authorized server $S_j$ uses this key to authorize the legitimate users.

### User registration phase

1. The new user $U_i$ selects the identity $ID_i$ and password $PW_i$. Then $U_i$ generates a random number $N_i$, computes $W_1 = h(PW_i||N_i)$ and $W_2 = h(ID_i \oplus N_i)$, and sends the registration request message $\{ID_i, W_1, W_2\}$ to the $RC$ via a secure channel.

2. After receiving the registration request message, $RC$ computes $A_i = h(ID_i||x|T_i|)$, $B_i = h(A_i)$, $X_i = B_i \oplus W_1$, $Y_i = h(PSK) \oplus W_2$ and $Z_i = PSK \oplus A_i$, where $T_i$ is the registration time. And $RC$ issues the smart card $SC_i$ to the user $U_i$, which contains $\{X_i, Y_i, Z_i, h(\cdot)\}$ via a secure channel.

3. Upon receiving the $SC_i$, $U_i$ imprints the personal biometrics $BIO_i$ at the sensor, and computes $N = N_i \oplus H(BIO_i)$, $V = h(ID_i||N_i||PW_i)$. Finally, the user $U_i$ stores $\{X_i, Y_i, Z_i, N, V, h(\cdot)\}$ into the $SC_i$.

### Login phase

1. $U_i$ inserts the $SC_i$ into the smart card reader and inputs the identity $ID_i$ and password $PW_i$ and imprints the biometrics $BIO_i$ at the sensor.

2. $SC_i$ computes $N_i = N \oplus H(BIO_i)$ and checks whether $h(ID_i||N_i||PW_i) = V$ holds. If it holds, $SC_i$ further compute $W_1 = h(PW_i||N_i)$, $W_2 = h(ID_i \oplus N_i)$, $B_i = X_i \oplus W_1$ and $h(PSK) = Y_i \oplus W_2$.

3. $SC_i$ generates a random number $n_1$, and computes $M_1 = h(PSK) \oplus n_1$, $M_2 = ID_i \oplus h(n_i||B_i)$ and $M_3 = h(ID_i||n_i||B_i)$.

4. $U_i$ sends the login request message $\{Z_i, M_1, M_2, M_3\}$ to $S_j$ over a public channel.

### Authentication phase

1. When receiving the login request message from $SC_i$, $S_j$ immediately computes $A_i = Z_i \oplus PSK, n_1 = M_1 \oplus h(PSK), ID_j = M_2 \oplus h(n_i||h(A_i))$, and verifies whether $h(ID_i||n_i||B_i)$ is consistent with $M_3$. If this verification holds, $S_j$ generates a random number $n_2$, and computes the session secret key $SK_{ji} = h(ID_i||SID_i||B_i||n_1||n_2)$, $M_4 = n_2 \oplus h(ID_i||n_1), M_5 = h(SK_{ji}||n_1)$.
n_2). Then S_j sends the authentication request message \{SID_j, M_4, M_5\} to SC_i via a public channel.

2. Upon receiving the authentication request message, SC_i retrieves n_2 = M_4 \oplus h(ID_i||N_1), SK_{ij} = h(ID_i||SID_j||B_i||n_1||n_2) and then checks whether h(SK_{ij}||n_1||n_2) = M_5 holds. If it holds, SC_i computes M_6 = h(SK_{ij}||n_2||n_1) and delivers the authentication reply \{M_6\} to S_j via a public channel.

3. S_j verifies whether h(SK_{ij}||n_2||n_1) = M_6 holds. If this verification holds, S_j can now use the session key SK_{ij} to communicate with U_i.

**Password change phase**

1. U_i inputs the ID_i, PW_i, and imprints his biometrics BIO_i at the sensor. SC_i computes N_1 = N \oplus h(BIO_i) and checks whether h(ID_i||N_1||PW_i) = V holds.

2. If the verification holds, U_i chooses the new password PW_{new}. SC_i computes W_1 = h(PW_{new}||N_1), W_{1new} = h(PW_{new}||N_1), X_{inew} = X_i \oplus W_1 \oplus W_{1new} and V_{new} = h(ID_i||N_1||PW_{new}).

3. SC_i replaces X_i with X_{inew} and V_i with V_{new} in the memory.

**Cryptanalysis of Mishra et al.’s scheme**

This section presents a cryptanalysis of Mishra et al.’s scheme and demonstrates that their scheme is still vulnerable to the masquerade attack, replay attack and Denial-of-Service attack. Also their scheme fails to achieve the perfect forward secrecy. Furthermore, Mishra et al.’s scheme does not provide the functionality of revocation/re-registration for user’s requirements.

**Masquerade attack**

Mishra et al.’s scheme is vulnerable to the masquerade attack. More narrowly, adversary E can be authenticated by another server S_k using the messages that user U_i sends to the server S_j for the authentication. Fig 2 shows the masquerade attack on Mishra et al.’s scheme.

First, U_i inserts the smart card and sends a login request message (1) to the S_j when he wants to be authenticated by S_j. After intercepting the login request message, E sends it to another server S_k. The message (1) does not include the information about the S_j as follows.

Message(1) = \{Z_i, M_1, M_2, M_3\},

where Z_i = PSK \oplus h(ID_i||T_i), M_1 = h(PSK) \oplus n_1, M_2 = ID_i \oplus h(n_1||B_i) and M_3 = h(ID_i||n_1||B_i). Therefore S_k executes the operation (2) and sends the authentication request message (3) to the E without any suspicion of the attack.

Then E transmits the message (3) to the U_i. And U_i does not check the identity of the server. He only checks the sameness with the SID_k in the M_3 and the SID_k in the message (3) as follows.

Message(3) = \{SID_k, M_4, M_5\},

where M_4 = n_2 \oplus h(ID_k||n_1), M_5 = h(SK_{k3}||n_1||n_2) and SK_{k3} = h(ID_k||SID_k||B_i||n_1||n_2). So U_i also executes the operation (4) and sends the authentication reply message (5) to the S_j without any suspicion of the attack.
Finally, $E$ intercepts the message (5) and transmits it to the $S_k$. Therefore $E$ can be authenticated by $S_k$. In conclusion, adversary $E$ can masquerade as a legitimate user to log in to the server $S_k$ so that Mishra et al.’s scheme becomes vulnerable to the masquerade attack.

In their scheme, $S_k$ cannot check whether $U_i$ wants to be authenticated by $S_k$. Thus $S_k$ authenticates all legitimate messages though these message are not sent to $S_k$. Similarly, $U_i$ does not check whether $S_k$ wants to be authenticated with $U_i$. He only checks whether $SID$ in the message (3) and $SID$ in the $M_5$ are the same.

To meet these challenges, the destination of message needs to be added to the login request message (1) and the authentication request message (3). So we add the information about $SID_{j}$ of server $S_j$ to the message (1), which means that $U_i$ want to be authenticated by $S_j$, not $S_k$. Meanwhile, the information about $AID_i$ of user $U_i$ needs to be added to the message (3), which means that $S_j$ wants to be authenticated by anonymous $U_i$.

**Replay attack**

In the same way, Mishra et al.’s scheme is vulnerable to the replay attack. In particular, adversary $E$ logs into the server $S_j$ with previous login request message (1). Upon receiving previous message (1), $S_j$ calculates $A_i = Z_i \oplus PSK$, $n_i = M_{P1} \oplus h(PSK)$, $ID_j = M_{P2} \oplus h(n_i || h(A_i))$, and verifies whether $h(ID_i || n_i || B_i) = M_{P3}$ holds without any suspicion of the attack. Since the verification holds, $S_j$ authenticates $E$ and $E$ logs into the server $S_j$. Thus Mishra et al.’s scheme becomes vulnerable to the replay attack.

In their scheme, $S_j$ does not check the freshness of login request message. So $S_j$ authenticates all legitimate login request messages though these messages are not fresh.

As a practical solution to prevent the replay attack, adding the timestamp to the message (1) helps server $S_j$ verify the freshness of login request message.

**Denial-of-Service attack**

Although the means and targets may vary, DoS attack is generally an attempt to make network resource or machines unavailable for intended users, which temporarily or indefinitely interrupts or suspends the services of a host connected to the networks. In the Mishra et al.’s
scheme, an adversary $E$ can carry out the DoS attack without difficulty. Fig 3 describes the procedure and effect of the DoS attack on Mishra et al.’s scheme.

In particular, $E$ collects the previous login request message \{Z, M_{P1}, M_{P2}, M_{P3}\} from the user $U_i$ and then forwards it to the server $S_j$. Upon receiving the login request, $S_j$, as always, executes the operation (2) which includes producing the random number once, sending message once, calculating the XOR operation 4 times, and performing the hash function 7 times. By applying the intercepted login request messages repeatedly, adversary $E$ can make the services of network resource or servers unavailable. Therefore Mishra et al.’s scheme becomes vulnerable to the DoS attack.

The reason for this result is that server $S_j$ cannot check the freshness of login request message from the user $U_i$. $S_j$ does not know whether the received messages are outdated so that it executes the operation (2) once receiving the login request message.

To resist the DoS attack, the timestamp needs to be added to the login request message. So we add the timestamp to the message (1), which helps the servers check the freshness of messages.

**No perfect forward secrecy**

The perfect forward secrecy means that if one of long-term keys is compromised, a session key which is derived from these long-term keys will not be compromised in the future [58]. Unfortunately, Mishra et al.’s scheme does not achieve the perfect forward secrecy. So adversary $E$ can calculate all session keys between the user $U_i$ and server $S_j$ if he knows one of long-term keys, such as $A_i$.

First, $E$ intercepts the $Z, SID_j, M_{P1}, M_{P2}$ and $M_{P4}$ from message (1) and message (3) in the previous communication between $U_i$ and $S_j$. Next, adversary knows one of long-term keys $A_i$, so that he can compute $PSK = A_i \oplus Z$, and $B_i$ from $B_i = h(A_i)$. Then, $E$ further calculate $n_{P1}$ from $n_{P1} = M_{P1} \oplus h(PSK)$, $ID_j$ from $ID_j = M_{P2} \oplus h(n_{P1})$, and $n_{P2}$ from $n_{P2} = M_{P4} \oplus h(ID_j)$. Finally, adversary $E$ acquires the all previous session keys from $SK_{Pj} = h(ID_j) \| SID_j \| B_i \| n_{P1} \| n_{P2})$. Therefore Mishra et al.’s scheme does not achieve the perfect forward secrecy.
In their scheme, $A_i$ is a shared key between $RC$ and $U_i$, which is calculated from $A_i = h(ID_i || x || Tr)$. $RC$ stores the information about $A_i$ and $h(A_i)$ in the smart card $SC_i$. The value of $A_i$ is invariable even if $U_i$ updates the password. So $A_i$ is treated as one of long-term keys. From the above, it is demonstrated that there are some defects during the generation of session keys.

To solve this problem, we need to add another secret information, such as $PSK$, to the generation of session keys. Also it is necessary to prevent adversary $E$ from calculating all session keys by using long-term key $A_i$ and information in the public channel.

No user revocation/re-registration phase

There is no user revocation/re-registration phase in the Mishra et al.’s scheme so that user $U_i$ cannot revoke his privilege or re-register when his smart card $SC_i$ is stolen or lost. To promote the functionality of scheme, we design the corresponding revocation/re-registration phase for the user’s requirements. And more details are showed in the Section 5.6.

The proposed scheme

Based on the cryptanalysis of Mishra et al.’s scheme, we present a novel robust biometric-based multi-server authentication and key agreement scheme which consists of six phases: server registration phase, user registration phase, login phase, authentication phase, password change phase and revocation/re-registration phase. There are also three participants, user $U_i$, server $S_j$ and registration center $RC$. Table 2 lists the notations applied in our scheme.

| Symbol | Notion |
|--------|--------|
| $U_i$, $S_j$ | $i$th user and $j$th server |
| $RC$, $E$ | The registration center and adversary |
| $ID_i$, $AID_i$, $SID_j$ | $U_i$’s identity, dynamic identity and $S_j$’s identity |
| $SC_i$, $PW_i$, $BIO_i$ | $U_i$’s smart card, password and biometrics |
| $R_i$, $P_i$ | $U_i$’s nearly random binary string and auxiliary binary string |
| $PSK$, $x$ | Pre shared key and master secret key |
| $h()$, $\oplus$, $||$ | Hash function, XOR operation and concatenation operation |

doi:10.1371/journal.pone.0149173.t002

In their scheme, $A_i$ is a shared key between $RC$ and $U_i$, which is calculated from $A_i = h(ID_i || x || Tr)$. $RC$ stores the information about $A_i$ and $h(A_i)$ in the smart card $SC_i$. The value of $A_i$ is invariable even if $U_i$ updates the password. So $A_i$ is treated as one of long-term keys. From the above, it is demonstrated that there are some defects during the generation of session keys.

To solve this problem, we need to add another secret information, such as $PSK$, to the generation of session keys. Also it is necessary to prevent adversary $E$ from calculating all session keys by using long-term key $A_i$ and information in the public channel.

No user revocation/re-registration phase

There is no user revocation/re-registration phase in the Mishra et al.’s scheme so that user $U_i$ cannot revoke his privilege or re-register when his smart card $SC_i$ is stolen or lost. To promote the functionality of scheme, we design the corresponding revocation/re-registration phase for the user’s requirements. And more details are showed in the Section 5.6.

The proposed scheme

Based on the cryptanalysis of Mishra et al.’s scheme, we present a novel robust biometric-based multi-server authentication and key agreement scheme which consists of six phases: server registration phase, user registration phase, login phase, authentication phase, password change phase and revocation/re-registration phase. There are also three participants, user $U_i$, server $S_j$ and registration center $RC$. Table 2 lists the notations applied in our scheme.

The proposed scheme improves the Mishra et al.’s scheme in the several aspects: 1) it resists the masquerade attack by adding the destination of messages, 2) it appends the timestamp to prevent the Denial-of-Service (DoS) attack, 3) it introduces pre shared key ($PSK$) into generation of session keys to achieve the perfect forward secrecy, 4) it provides the revocation/re-registration phase for user’s requirements, and 5) it enhances the performance of scheme, especially login phase. The details are described in the following subsections.

Server registration phase

The server registration phase is illustrated in Fig 4 and explained as follows.

1. The server $S_j$ sends a join request message to the registration center $RC$, if it wants to become an authorized server in the system.

2. After receiving the join request message, $RC$ authorizes the server and replies with the pre shared key ($PSK$) to the server $S_j$ by applying the Key Exchange Protocol (IKEv2) through a secure channel.
3. Upon receiving the secret key $PSK$, authorized server $S_j$ uses the shared information, such as $PSK$ and $h(PSK)$, to check the user’s legitimacy in the authentication phase.

**User registration phase**

The new user $U_i$ needs to execute the user registration phase with the registration center $RC$ via a secure channel. The user registration phase is showed in Fig 5 and described as follows.

1. First, $U_i$ imprints the personal biometric information $BIO_i$ at the sensor. After that, sensor sketches $BIO_i$ extracts $(R_i, P_i)$ from $Gen(BIO_i) \rightarrow (R_i, P_i)$, and stores $P_i$ in the memory. Next, $U_i$ selects the identity $ID_i$ and password $PW_i$, and computes $RPW_i = h(PW_i||R_i)$.

2. Finally, $U_i$ sends the registration request message $\{ID_i, RPW_i\}$ to the $RC$ via a secure channel.

After receiving the registration request message, $RC$ adds a novel entry $\langle ID_i, N_i = 1 \rangle$ to the database, where $N_i$ means the times of user registration. And then $RC$ computes $A_i = h(ID_i||RPW_i)$.

**Fig 5. The user registration phase.**

![Fig 5. The user registration phase.](doi:10.1371/journal.pone.0149173.g005)
3. RC issues the smart card \( SC_i \) to the user \( U_i \), which contains \{ \( B_i \), \( C_i \), \( D_i \), \( V_i \) \} over a secure channel.

4. Upon receiving the \( SC_i \), \( U_i \) stores \( P_i \) into the \( SC_i \) and initializes the authentication environments.

**Login phase**

During the login phase, smart card \( SC_i \) can check an error event immediately by using the identification, password, and biometric information. The login phase is illustrated in Fig 6 and explained as follows.

1. \( U_i \) inserts the \( SC_i \) into the smart card reader, inputs the identity \( ID_i \) and password \( PW_i \), and imprints \( BIO_i \) at the sensor. After that, sensor sketches \( BIO_i \) and recovers \( R_i \) from \( Rep(BIO_i, P_i) \rightarrow R_i \).

2. \( SC_i \) calculates \( RPW_i = h(PW_i||R_i) \) and checks whether \( h(ID_i||RPW_i) = V_i \) holds. If it holds, \( SC_i \) further calculates \( h(PSK) = B_i \oplus C_i \).

3. \( SC_i \) generates a random number \( N_i \), and computes \( AID_i = ID_i \oplus h(N_i) \), \( M_1 = RPW_i \oplus N_i \oplus h(PSK) \) and \( M_2 = h(AID_i||N_i||RPW_i||SID_j||T_i) \), where \( T_i \) is additional timestamp.

4. \( SC_i \) sends the login request message \{ \( AID_i, M_1, M_2, B_i, D_i, T_i \) \} to \( S_j \) via a public channel.

**Authentication phase**

In the authentication phase, server \( S_j \) confirms the destination and freshness of login request message. The authentication phase is showed in Fig 7 and described as follows.
1. When receiving the login request message from $U_i$, server $S_j$ verifies whether $T_i - T_j \leq \Delta T$ is valid, where $\Delta T$ is the time interval and $T_j$ is the time when $S_j$ receives the login request message. If it holds, $S_j$ continues to perform the next step. Otherwise, the login request will be rejected by $S_j$.

2. $S_j$ retrieves $A_i = PSK \oplus D_i \oplus h(PSK)$, $RPW_i = B_i \oplus h(A_i)$, $N_1 = RPW_i \oplus M_1 \oplus h(PSK)$, and checks $h(AID_i) || N_1 || RPW_i || SID_i || T_i) = M_2$. $S_j$ generates a random number $N_2$ and computes $SK_{ij} = h(AID_i) || SID_i || N_1 || N_2)$ and calculates $M_3 = N_2 \oplus h(AID_i) || N_1 \oplus h(PSK)$ and $M_4 = h(SID_i) || N_2 || AID_i)$. $S_j$ sends the authentication request message $\{SID_j, M_3, M_4\}$ to $U_i$ via a public channel.

3. Upon receiving the authentication request, $S_i$ retrieves $N_2 = M_3 \oplus h(AID_i) || N_1 \oplus h(PSK)$, $SK_{ij} = h(AID_i) || SID_i || N_1 || N_2)$ checks $h(SID_i) || N_2 || AID_i) = M_4$ holds. If it holds, $S_i$ computes $M_5 = h(SK_{ij}) || N_1 || N_2)$ and delivers the authentication reply $\{M_5\}$ to $S_j$ via a public channel.

4. $S_j$ verifies whether $h(SK_{ij}) || N_1 || N_2) = M_5$ holds. If this verification holds, $S_j$ uses the session key $SK_{ij} to communicate with $U_i$. Otherwise, authentication will be rejected by $S_j$. 

---

**Fig 7. The authentication phase.**

doi:10.1371/journal.pone.0149173.g007
Password change phase

During the password change phase, $U_i$ updates the password without any assistance from
server $S_j$ and registration center $RC$. This phase consists of the following steps.

1. $U_i$ inputs $ID_i$ and $PW_i$, and imprints his biometrics $BIO_i^r$ at sensor. After that, the sensor
   sketches $BIO_i^r$ and recovers $R_i$ from $Rep(BIO_i^r, P_i) \rightarrow R_i$.

2. $SC_i$ calculates $RPW_i = h(PW_i \| R_i)$ and checks whether $h(ID_i \| RPW_i) = V_i$ holds. If the verifica-
   tion holds, $SC_i$ asks $U_i$ for a new password. Otherwise, password change phase is termi-
   nated immediately by $SC_i$.

3. $U_i$ inputs new password $PW_{new}^i$ and $SC_i$ further computes $RPW_{new}^i = h(PW_{new}^i \| R_i)$,
   $B_i^{new} = B_i \oplus RPW_i \oplus RPW_{new}^i$, $C_i^{new} = C_i \oplus RPW_i \oplus RPW_{new}^i$ and $V_i^{new} = h(ID_i \| RPW_{new}^i)$.

4. $SC_i$ replaces $B_i$ with $B_i^{new}$, $C_i$ with $C_i^{new}$ and $V_i$ with $V_i^{new}$ in the memory.

User revocation/re-registration phase

The functionality of user revocation/re-registration helps user $U_i$ revoke his privilege or re-reg-
ister when his smart card $SC_i$ is stolen or lost. If $U_i$ wants to revoke his privilege, he needs to
send a revocation request message, his smart card and verification message $\{RPW_i\}$ to the regis-
tration center $RC$ over a secure channel. $RC$ verifies whether $U_i$ is valid. If it holds, $RC$ further
modifies the corresponding entry by setting $(ID_i, N_i = 0)$. Similarly, upon receiving a re-regis-
tration request message via a secure channel, $RC$ executes the steps described in the section 5.2
and replaces $(ID_i, N_i = N_i + 1)$ with $(ID_i, N_i)$ to help $U_i$ re-register. The user revocation or re-
registration phase makes our scheme more robust than other related schemes in the
functionality.

Analysis of our scheme

An authentication and key agreement scheme has three important requirements: security,
functionality and performance. It is necessary to analyze the proposed scheme from three
aspects mentioned above. In this section, we explain how the proposed scheme is satisfied with
these requirements, and compare our scheme with other related multi-server authentication
and key agreement schemes.

Informal security analysis

In this section, we assume that adversary $E$ has the capacity which is assumed in Section 2.1.
Also we analyze the strength of the proposed scheme against the following common attacks
through informal security analysis.

**Resistance to replay attack.** The replay attack means that adversary $E$ intercepts the trans-
mitted messages for making use of these data in some manner, which involves copying and
possibly altering the data in various ways. Although adversary $E$ intercepts the previous login
request message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$ and sends it to server $S_j$ repeatedly, $S_j$ verifies the
legality of message by checking $T_i$ and $N_i$ as follows.

$$M_x = h(AID_i \| N_i \| RPW_i \| SID_i \| T_i),$$

where $T_i$ and $N_i$ are different in every session so that $E$ is not authenticated by $S_j$. So our
scheme is secure against the replay attack by adding the timestamp $T_i$ and random nonce $N_i$.

**Resistance to modification attack.** Though adversary $E$ intercepts the transmitted mes-
ages and attempts to modify them for authentication, proposed scheme verifies whether
received messages are modified with the help of one-way hash function. And $E$ cannot retrieve $N_1$, $N_2$ and $PSK$ from intercepted messages so that he does not have the capabilities to generate a legitimate authentication message. Therefore, our scheme prevents the modification attack.

**Resistance to stolen-verifier attack.** In the proposed scheme, Registration center $RC$ and servers do not possess the user’s password or biometrics so that adversary $E$ cannot steal the password-verifier or biometrics-verifier about legitimate users even if he has the authority to access the database of the $RC$ and servers. Thus, our scheme resists the stolen-verifier attack.

**Resistance to off-line guessing attack.** With the assistance of the side-channel attacks such as SPA or DPA, adversary $E$ obtains $B_i$, $C_i$, $D_i$ and $V_i$. But he cannot verify the user’s password in the off-line environment without $BIO_i$, $PSK$, $x$ and $N_1$. Also user’s password is protected by one-way hash function, such as, $h(PW_i||R_i)$, where $R_i$ possesses high entropy. Moreover, there is no the same biometric templates between any two people. In conclusion, our scheme is secure against the off-line guessing attack.

**Resistance to masquerade attack.** The forgery attack means that legitimate yet malicious user $E$ attempts to forge another legitimate user for login and authentication. In the communication between server $S_j$ and user $Ui$, $U_i’s$ real identity $ID_i$ is protected by anonymous identity $AID_i$, such as $AID_i = ID_i \oplus h(N_i)$. Furthermore, random nonce $N_i$ changes in every session. So malicious user $E$ cannot acquire another legitimate user’s real identity $ID_i$. As a result, our scheme prevents the forgery attack.

**Resistance to insider attack.** Malicious insider $E$ is familiar with system policies or procedures, and has an authorized system access, who tries to obtain user’s private information such as password and biometrics. $RC$ cannot retrieve the password $PW_i$ or biometrics $BIO_i$ from $RPW_i = h(PW_i||R_i)$, Moreover $RC$ does not store $RPW_i$ in the database. Thus, our scheme resists the insider attack.

**Resistance to masquerade attack.** Under this attack, adversary $E$ is authenticated by server $S_j$ with a fake or real identity. In Mishra et al.’s scheme, $E$ applies the transmitted messages between $S_j$ and $U_i$ to acquire the access of server $S_k$. To meet this problem, destination of message is added to the login request message and authentication request message, such as $M_2 = h(AID_i||N_i||RPW_i||SID_j||T_i)$ and $M_4 = h(SID_j||N_2||AID_i)$, so that $U_i$ and $S_j$ verify whether the one wants to be authenticated by the other one. At the same time, $E$ cannot compute $M_2$ or $M_4$ without $N_1$ or $N_2$. Therefore, our scheme is secure against the masquerade attack.

**Resistance to smart card attack.** In the smart card attack, adversary $E$ tries to apply the information obtained from smart card $SC_i$ to be authenticated by server $S_j$ without the password or biometrics. With SPA or DPA, $E$ obtains $B_i$, $C_i$, $D_i$ and $V_i$, which are stored in $SC_i$. In the proposed scheme, a session key between user $U_i$ and server $S_j$ is generated as follow.

\[
A_i = D_i \oplus PSK \oplus h(PSK),
\]
\[
N_i = RPW_i \oplus M_i \oplus h(PSK),
\]
\[
N_2 = M_4 \oplus h(AID_i||N_i) \oplus h(PSK),
\]
\[
SK_{ij} = h(AID_i||SID_j||N_i||N_2).
\]

Although $E$ obtains $M_1$ and $M_3$ via public channels, it is difficult for him to retrieve $N_1$, $N_2$ and $AID_i$ without $PSK$. Above all, our scheme prevents the smart card attack.

**Resistance to user impersonation attack.** The user impersonation attack means that adversary $E$ impersonates user $U_i$ using only smart card $SC_i$ but without the password or biometrics. The proposed scheme applies $h(PSK)$ to protect $N_i$, $N_2$ and $AID_i$ even if $E$ acquires $B_i$. 
Cryptanalysis and Improvement of an Authenticated Key Agreement Scheme

Formal security analysis

With the help of the formal security analysis, we demonstrate that our scheme is secure against adversary $E$. For this purpose, we define oracle $Reveal$ as follows: it unconditionally outputs $x$ from one-way hash function $y = h(x)$. The following two theorems provide the formal security analysis for our scheme.

**Theorem 1.** Under the assumption that one-way hash function $h(\cdot)$ closely behaves like oracle $Reveal$, our scheme is provably secure against adversary $E$ for retrieving the identity $ID_i$ of user $U_i$, pre shared key $PSK$ of server $S_j$ and session key $SK_{ij}$ between $U_i$ and $S_j$.

**Proof.** We need to construct adversary $E$ who has the capacity to retrieve the identity $ID_i$ of user $U_i$, pre shared key $PSK$ of server $S_j$ and session key $SK_{ij}$ between $U_i$ and $S_j$. Adversary $E$ applies the oracle $Reveal$ to execute the experimental algorithm $EXP_{E, BMAKAS}^{HASH}$, where the BMAKAS means proposed biometric-based multi-server authentication and key agreement scheme. The details of Algorithm 1 are described in the Table 3.

And we define the success probability of $EXP_{E, BMAKAS}^{HASH}$ as

$$\text{Success}_1 = |P(\text{EXP}_{E, BMAKAS}^{HASH} = 1) - 1|,$$

where $P(\cdot)$ means the probability of $\text{EXP}_{E, BMAKAS}^{HASH}$. The advantage function for algorithm $\text{EXP}_{E, BMAKAS}^{HASH}$ becomes $\text{Adv}_1(\epsilon_1, q_{\text{Reveal}}) = \max\{\text{Success}_1\}$, where the maximum for adversary $E$ depends on the execution time $\epsilon_1$ and number of queries $q_{\text{Reveal}}$ made to the oracle $Reveal$. Our scheme is provably secure against adversary $E$, if $\text{Adv}_1(\epsilon_1, q_{\text{Reveal}}) \leq \epsilon_1$, for any sufficiently small $\epsilon_1 > 0$. If adversary $E$ has the ability to retrieve $x$ from one-way hash function $y = h(x)$, then he can easily derive the identity $ID_i$ pre shared key $PSK$ and session key $SK_{ij}$ to win the game. However, it is a computationally infeasible problem to retrieve the inputs of one-way hash function. So $\max_{E}\{\text{Success}_1\} = \text{Adv}_1(\epsilon_1, q_{\text{Reveal}}) \leq \epsilon_1$, for any sufficiently small $\epsilon_1 > 0$. In conclusion, our scheme is provably secure against adversary $E$ for retrieving the identity $ID_i$ of user $U_i$, pre shared key $PSK$ of server $S_j$ and session key $SK_{ij}$ between $U_i$ and $S_j$.

**Theorem 2.** Under the assumption that one-way hash function $h(\cdot)$ closely behaves like oracle $Reveal$, our scheme is provably secure against adversary $E$ for retrieving the password $PW_i$ of user $U_i$, even if smart card $SC_i$ is stolen.

**Proof.** We need to construct the adversary $E$ who has the capacity to retrieve the password $PW_i$. Adversary $E$ extracts all the information $\{B_i, C_i, D_i, V_i\}$ from stolen smart card $SC_i$ and applies the oracle $Reveal$ to execute the experimental algorithm $EXP_{E, BMAKAS}^{HASH}$. The details of Algorithm 2 are described in the Table 4.
Table 3. Algorithm EXP1.IDAI.LBAKAT.

1. Eavesdrop the login request message \{AID_i, M_1, M_2, B_i, D_i, T_i\} during the login phase, where \(AID_i = ID_i \oplus h(N_i)\), \(M_1 = RPW_i \oplus N_i \oplus h(RSK)\) and \(M_2 = h(AID_i||N_i||RPW_i||SID_i||T_i)\).
2. Apply the oracle Reveal to retrieve \(AID'_i, N'_i, RPW'_i, SID'_i\) and \(T'_i\) from \(\text{Reveal}(M_i) \rightarrow \langle AID'_i||N'_i||RPW'_i||SID'_i||T'_i \rangle\).
3. If \((AID'_i = AID_i)\) then
   4. Calculate \(ID'_i = AID_i \oplus h(N'_i)\) and \(H_i = RPW_i \oplus N'_i \oplus M_i\).
   5. Apply the oracle Reveal to retrieve \(RSK_i\) from \(\text{Reveal}(H_i) \rightarrow \langle RSK_i \rangle\).
6. Eavesdrop the authentication request message \{SID'_i, M'_i, A'_i\} during the authentication phase, where \(M'_i = N'_2 \oplus h(AID'_i||N'_i) \oplus h(RSK)\) and \(M'_2 = h(SID_i||N'_i||AID_i)\).
7. Further apply the oracle Reveal to retrieve \(AID''_i, N''_i\) and \(SID''_i\) from \(\text{Reveal}(M_i) \rightarrow \langle AID''_i||N''_i||SID''_i \rangle\).
8. If \((SID_i = SID''_i)\) and \((AID_i = AID''_i)\) then
   9. Calculate \(H'_i = N''_i \oplus h(AID''_i||N''_i) \oplus M'_i\).
10. Apply the oracle Reveal to retrieve \(PSK''_i\) from \(\text{Reveal}(H'_i) \rightarrow \langle PSK''_i \rangle\).
11. If \((PSK''_i = PSK_i)\) then
   12. Calculate \(SK''_i = h(AID||SID||N||N'_i)\).
   13. Accept \(ID'_i, PSK''_i\) and \(SK''_i\) as the identity ID of user \(U_i\), pre shared key \(PSK_i\) of server \(S_j\), and session key \(SK''_i\) between \(U_i\) and \(S_j\), respectively.
14. return 1 (Success)
15. else
16. return 0 (Failure)
17. end if
18. else
19. return 0 (Failure)
20. end if
21. else
22. return 0 (Failure)
23. end if

doi:10.1371/journal.pone.0149173.t003

Table 4. Algorithm EXP2.IDAI.LBAKAT.

1. Extract all the information \{\(B_i, C_i, D_i, V_i\)\} from stolen smart card \(SC_i\); with the help of side channel attacks, where \(V_i = h(ID_i||RPW_i)\) and \(RPW_i = h(PW_i||R_i)\).
2. Apply the oracle Reveal to retrieve \(ID'_i\) and \(RPW'_i\) from \(\text{Reveal}(V_i) \rightarrow \langle ID'_i||RPW'_i \rangle\).
3. Eavesdrop the login request message \{AID_i, M_1, M_2, B_i, D_i, T_i\} during the login phase, where \(AID_i = ID_i \oplus h(N_i)\) and \(M_1 = h(AID_i||N_i||RPW_i||SID_i||T_i)\).
4. Apply the oracle Reveal to retrieve \(AID''_i, N''_i, RPW''_i, SID''_i\) and \(T''_i\) from \(\text{Reveal}(M_i) \rightarrow \langle AID''_i||N''_i||RPW''_i||SID''_i||T''_i \rangle\).
5. Calculate \(ID''_i = AID''_i \oplus h(N''_i)\).
6. If \((ID''_i = ID_i)\) then
   7. Apply the oracle Reveal to retrieve \(PW''_i\) and \(R''_i\) from \(\text{Reveal}(RPW''_i) \rightarrow \langle PW''_i||R''_i \rangle\).
   8. Accept \(PW''_i\) as the password \(PW_i\) of user \(U_i\).
   9. return 1 (Success)
10. else
11. return 0 (Failure)
12. end if

doi:10.1371/journal.pone.0149173.t004
Also we define the success probability of $\text{EXP}^2_{\text{HASH}}^\text{BMAKAS}$ as $\text{Success}_2 = |P(\text{EXP}^2_{\text{HASH}}^\text{BMAKAS} = 1) - 1|$, where $P(\cdot)$ means the probability of $\text{EXP}^2_{\text{HASH}}^\text{BMAKAS}$. The advantage function for algorithm $\text{EXP}^2_{\text{HASH}}^\text{BMAKAS}$ becomes $\text{Adv}_2(t_2, q_{\text{Reveal}}) = \max_{\epsilon} \{\text{Success}_2\}$, where the maximum for adversary $E$ depends on the execution time $t_2$ and number of queries $q_{\text{Reveal}}$ made to the oracle Reveal. Our scheme is provably secure against adversary $E$, if $\text{Adv}_2(t_2, q_{\text{Reveal}}) \leq \epsilon_2$, for any sufficiently small $\epsilon_2 > 0$. If adversary $E$ has the ability to retrieve $x$ from one-way hash function $y = h(x)$, then he can easily derive the password $PW_i$ to win the game. However, it is a computationally infeasible problem to retrieve the inputs of one-way hash function. So $\max_{\epsilon_2} \{\text{Success}_2\} = \text{Adv}_2(t_2, q_{\text{Reveal}}) \leq \epsilon_2$, for any sufficiently small $\epsilon_2 > 0$. In conclusion, our scheme is provably secure against adversary $E$ for retrieving the password $PW_i$ of user $U_i$.

**Functionality analysis**

Various functionality requirements for a multi-server authentication and key agreement scheme have been suggested in previous studies. In this section, we show that our scheme provides these functionalities.

**Anonymity.** The anonymity means that user’s real identity is not disclosed to an unauthorized party. In the presented scheme, $U_i$ calculates the dynamic identity $AID_i$ from $AID_i = ID_i \oplus h(N_i)$, and $N_i$ does not leak out from the messages over public channels. Thus, adversary $E$ cannot compute the user’s identity $ID_i$ without $N_i$. The authorized server $S_j$ retrieves $A_i = D_i \oplus PSK \oplus h(PSK)$ and $RPW_i = B_i \oplus h(A_i)$, and further calculates $N_i$ from $N_i = RPW_i \oplus M_i \oplus h(PSK)$. So only authorized servers confirm the real identity of $U_i$. As a result, adversary $E$ cannot acquire the user’s real identity, but user $U_i$ is authenticated anonymously by server $S_j$.

**Mutual authentication.** The mutual authentication is achieved when two parties authenticate each other. In our scheme, users and servers authenticate each other by using $N_1$, $N_2$, $h(PSK)$, $D_i$ and $T_i$. During the authentication phase, server $S_j$ verifies whether $M_2$ is consistent with $h(AID_j||N_1||RPW_i||SID_j||T_i)$ to authenticate the user $U_i$. And $U_i$ authenticates $S_j$ by checking whether $h(SID_j||N_2||AID_i) = M_4$ holds. In conclusion, our scheme provides the mutual authentication.

**Session key agreement.** The session key agreement means that users and servers securely establish a session key which is used for protecting the subsequent communication. In the proposed scheme, a session key $SK_{ij} = h(AID_i||SID_j||N_1||N_2)$ is generated by user $U_i$ and server $S_j$, where $N_1$ and $N_2$ are different in every session. Therefore, session keys are different in each session so that it is difficult for adversary $E$ to retrieve the previous session keys from the intercepted messages.

**Perfect forward secrecy.** The perfect forward secrecy means that a session key will not be compromised if the user’s long-term key is compromised in the future [11, 15]. In our scheme, a session key between user $U_i$ and server $S_j$ is calculated as follow:

$$A_i = D_i \oplus PSK \oplus h(PSK),$$

$$RPW_i = B_i \oplus h(A_i),$$

$$N_i = RPW_i \oplus M_i \oplus h(PSK),$$

$$N_2 = M_i \oplus h(AID_j||N_1) \oplus h(PSK),$$

$$SK_{ij} = h(AID_j||SID_j||N_1||N_2).$$
Although user’s long-term key $h(PSK)$ is compromised, adversary $E$ cannot calculate $RPWi$ and $PSK$ so that he cannot retrieve $N_1$ and $N_2$ to generate the session keys between $U_i$ and $S_j$. Above all, our scheme achieves the perfect forward secrecy.

**User revocation/re-registration.** The user $U_i$ needs to send a revocation or re-registration request message to the registration center $RC$ over a secure channel if he wants to revoke his privilege or re-register. $RC$ help $U_i$ revoke his privilege or re-register by modifying $(ID_i, N_i)$ in the database. The functionality of user revocation/re-registration meets the requirements of practical applications. It also makes our scheme more robust than other related schemes.

**Biometric information protection.** In conventional scheme, biometric information of user is directly stored in the smart card $SC_i$ so that adversary $E$ obtains biometrics from lost smart card with the assistance of side channel attacks. We adopt a high security mechanism to solve this problem. The nearly random string $R_i$ is protected by one-way hash function, which is extracted from biometric information $BIO_i$ by fuzzy extractor. And more details are described in Section 2.2. So it makes impossible for $E$ to obtain the biometric information. In conclusion, our scheme provides the biometric information protection.

**Efficiency analysis**

The efficiency is an important consideration in the aspect of evaluating the schemes. The efficiency of a multi-server authentication and key agreement scheme can be measured by the following metrics, single registration, secure and simple password modification, fast error detection, and low computational cost.

**Single registration.** The single registration means that a single point of registration allows users to acquire the access to all servers in the system. In the proposed scheme, user $U_i$ registers with registration center $RC$ only once to be authenticated with every server and apply the server’s services anonymously. So our scheme achieves the single registration.

**Secure and simple password modification.** The secure and simple password modification demands that users change their passwords without the assistance of any third trusted party and the authenticity of the users is verified by their smart card. In our scheme, user $U_i$ changes the password conveniently and does not require any communication with registration center $RC$. Furthermore, smart card $SC_i$ checks whether $h(ID_i||RPWi) = V_i$ holds for every password modification so that adversary $E$ cannot change the password even if he acquires the smart card and password. In conclusion, proposed scheme provides the secure and simple password modification.

**Fast error detection.** It is necessary to provide the fast error detection, which means that smart card $SC_i$ checks the incorrect passwords or any other discrepancies quickly. In the login and password change phases, $SC_i$ detects the errors immediately, such as inaccurate identities, incorrect passwords and false biometrics without the help of registration center $RC$ and server $S_j$. Therefore, our scheme achieves the fast error detection.

**Low computational cost.** The computational cost of the scheme should be minimized in practice. As the major parties of communication, $U_i$ and $S_j$ produce the random number twice, calculate the XOR operation 12 times, and perform the hash function 15 times to complete the login and authentication phases. As a result, computational cost of our scheme is a little lower than other related schemes.

**Comparisons with related schemes**

In this section, we compare the resistance, functionality and performance of our scheme with other related existing biometric-based multi-server authentication and key agreement schemes,
such as Chuang et al.’s scheme [51], Mishra et al.’s scheme [53], Xue et al.’s scheme [59] and Li et al.’s scheme [60].

Table 5 lists the resistance comparison of various biometric-based multi-server authenticated key agreement schemes. We define the following notations: R1: resistance to replay attack, R2: resistance to modification attack, R3: resistance to stolen-verifier attack, R4: resistance to off-line guessing attack, R5: resistance to forgery attack, R6: resistance to insider attack, R7: resistance to masquerade attack, R8: resistance to smart card attack, R9: resistance to user impersonation attack, R10: resistance to DoS attack and R11: resistance to server spoofing attack in the Table 5. The result indicates that our scheme is more secure and achieves the all resistance requirements.

Table 6 shows the functionality comparison of proposed scheme with other related schemes. In the Table 6, we use the following notations: F1: anonymity, F2: mutual authentication, F3: session key agreement, F4: perfect forward secrecy, F5: user revocation/re-registration and F6: biometric information protection. And we further compare our scheme with Lu et al.’s scheme [24] which is another improved scheme. It can be seen that our scheme provides more functionality requirements than other related schemes.

We compare our scheme with other biometric-based multi-server authentication and key agreement schemes for computational overhead, communication overhead and storage requirement involved in the login and authentication phases. In order to measure the computational complexity, we apply the number of hash function operations as time complexity since

Table 5. The resistance comparison.

|                  | Chuang et al.’s [51] | Mishra et al.’s [53] | Xue et al.’s [59] | Li et al.’s [60] | Ours |
|------------------|----------------------|----------------------|------------------|------------------|------|
| R1               | No                   | No                   | No               | No               | Yes  |
| R2               | Yes                  | Yes                  | Yes              | Yes              | Yes  |
| R3               | Yes                  | Yes                  | No               | No               | Yes  |
| R4               | Yes                  | Yes                  | No               | No               | Yes  |
| R5               | Yes                  | Yes                  | Yes              | Yes              | Yes  |
| R6               | Yes                  | Yes                  | No               | Yes              | Yes  |
| R7               | No                   | No                   | No               | No               | Yes  |
| R8               | No                   | Yes                  | Yes              | No               | Yes  |
| R9               | No                   | Yes                  | Yes              | Yes              | Yes  |
| R10              | No                   | No                   | Yes              | Yes              | Yes  |
| R11              | Yes                  | Yes                  | No               | Yes              | Yes  |

doi:10.1371/journal.pone.0149173.t005

Table 6. The functionality comparison.

|                  | Chuang et al.’s [51] | Mishra et al.’s [53] | Xue et al.’s [59] | Li et al.’s [60] | Lu et al.’s [48] | Ours |
|------------------|----------------------|----------------------|------------------|------------------|------------------|------|
| F1               | Yes                  | Yes                  | Yes              | Yes              | Yes              | Yes  |
| F2               | No                   | Yes                  | Yes              | Yes              | Yes              | Yes  |
| F3               | Yes                  | Yes                  | Yes              | Yes              | Yes              | Yes  |
| F4               | No                   | No                   | Yes              | Yes              | Yes              | Yes  |
| F5               | No                   | No                   | No               | No               | No               | Yes  |
| F6               | No                   | Yes                  | No               | No               | Yes              | Yes  |

doi:10.1371/journal.pone.0149173.t006
the XOR operation requires very little computational cost, where \( T_h \) stands for the computation time for hash function. According to the Xue et al.’s work [61], we learn that the average running time of a one-way secure hash function operation is about 0.2 ms. As shown in the Table 7 and Fig 8, we demonstrate the comparison among our scheme and other related schemes in terms of the computation overhead. In the Table 7, we use the following notations: S1: computation overhead in the login phase, S2: execution overhead in the login phase, S3: computation overhead in the authentication phase, S4: execution overhead in the authentication phase and S5: total execution overhead. The proposed scheme requires lower computation overhead than other schemes.

To estimate the communication efficiency, we assume that the length of security parameters, such as the bit length of random number \( N_i \) is 160, the bit length of user identity is 160, the bit length of timestamp \( T_i \) is 16 and the output length of hash function is 160 if we follow

|               | Chuang et al.’s [51] | Mishra et al.’s [53] | Xue et al.’s [61] | Li et al.’s [60] | Lu et al.’s [48] | Ours |
|---------------|-----------------------|-----------------------|-------------------|-----------------|-----------------|------|
| S1            | \( 4T_h \)            | \( 7T_h \)            | \( 5T_h \)        | \( 7T_h \)      | \( 4T_h \)      | \( 4T_h \)    |
| S2            | 0.8ms                 | 1.4ms                 | 1.0ms             | 1.4ms           | 1.0ms           | 0.8ms |
| S3            | 13T_h                 | 11T_h                 | \( 14T_h \)       | \( 16T_h \)     | \( 13T_h \)     | \( 11T_h \)  |
| S4            | 2.6ms                 | 2.2ms                 | 2.8ms             | 3.2ms           | 2.6ms           | 2.2ms |
| S5            | 3.4ms                 | 3.6ms                 | 3.8ms             | 4.6ms           | 3.6ms           | 3.0ms |

doi:10.1371/journal.pone.0149173.t007

Fig 8. The computation cost comparison.

doi:10.1371/journal.pone.0149173.g008
the SHA-1 which is applied in the most of previous schemes. In our scheme, $U_i$ transmits the request message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$ to $S_j$ during the login phase, and its length is $(160 + 160 + 160 + 160 + 160 + 16)/8 = 102$ bytes. And in the stage of authentication, communication overhead is $(160 + 160 + 160 + 160)/8 = 80$ bytes, which contains the authentication request message $\{SID_j, M_3, M_4\}$ and authentication reply $\{M_5\}$. So total communication overhead of proposed scheme is $102 + 80 = 182$ bytes. Analogously, we measure the communication overhead of related schemes. In order to estimate the storage requirement, we consider the messages stored in the smart card as the storage overhead and calculate the byte length of stored information. In our scheme, the stored message $\{B_i, C_i, D_i, V_i, P_i\}$ requires $(160 + 160 + 160 + 160 + 160)/8 = 100$ bytes. Similarly, we estimate the storage requirement of other schemes. Table 8 and Fig 9 show the comparisons regarding on the communication and storage costs of various multi-server authentication and key agreement schemes. We provide the following notations: C1: communication cost in the login phase, C2: communication cost in

|                | Chuang et al.’s [51] | Mishra et al.’s [53] | Xue et al.’s [59] | Li et al.’s [60] | Lu et al.’s [48] | Ours     |
|----------------|----------------------|----------------------|-------------------|-----------------|-----------------|---------|
| C1             | 80 bytes             | 80 bytes             | 83 bytes          | 80 bytes        | 82 bytes        | 102 bytes |
| C2             | 80 bytes             | 80 bytes             | 259 bytes         | 60 bytes        | 64 bytes        | 80 bytes |
| C3             | 160 bytes            | 160 bytes            | 342 bytes         | 140 bytes       | 146 bytes       | 182 bytes |
| C4             | 80 bytes             | 100 bytes            | 60 bytes          | 100 bytes       | 60 bytes        | 100 bytes |

doi:10.1371/journal.pone.0149173.t008

Fig 9. The communication and storage costs comparison.

doi:10.1371/journal.pone.0149173.g009
the authentication phase, C3: total communication cost and C4: storage cost in the Table 8. With the same level of communication overhead and storage requirement, our scheme obviously has advantages in the computational complexity by considering the computation cost of these related schemes. From the results of comparisons given above, we conclude that our scheme has better efficiency between resistance, functionality and performance than other related schemes.

Conclusion
With the security requirements of networks, biometrics authenticated schemes which are applied in the multi-server environment come to be more crucial and widely deployed. In this paper, we analyze the security of Mishra et al.’s scheme. Based on the cryptanalysis of their scheme, we propose a novel biometric-based multi-server authentication and key agreement scheme. The presented scheme improves the Mishra et al.’s scheme, and satisfies the desirable security requirements which are demonstrated through informal and formal security analysis respectively. Also our scheme provides some significant functionalities which are not considered in the most of existing authentication schemes, such as, user revocation or re-registration and biometric information protection. In addition, comparisons in the security, functionality and performance between proposed scheme and several related ones are given. The results show that our scheme has more secure properties, more functionalities and lower computation cost with the same level of communication overhead and storage requirement. We conclude that our scheme is obviously more appropriate for practical applications in the remote distributed networks.

Author Contributions
Conceived and designed the experiments: CQW XZ ZMZ. Performed the experiments: CQW XZ ZMZ. Analyzed the data: CQW XZ ZMZ. Contributed reagents/materials/analysis tools: CQW XZ ZMZ. Wrote the paper: CQW XZ ZMZ.

References
1. Khan MK, Zhang J. Improving the security of a flexible biometrics remote user authentication scheme’. Computer Standards & Interfaces. 2007; 29(1): 82–85. doi: 10.1016/j.csi.2006.01.002
2. He D, Kumar N, Khan MK, Lee JH. Anonymous two-factor authentication for consumer roaming service in global mobility networks. IEEE Transactions on Consumer Electronics. 2013; 59(4): 811–817. doi: 10.1109/TCE.2013.6689693
3. Mishra D. Design and analysis of a provably secure multi-server authentication scheme. Wireless Personal Communications. 2015. doi: 10.1007/s11277-015-2975-0
4. Lamport L. Password authentication with insecure communication. Communications of the ACM. 1981; 24(11): 770–772. doi: 10.1145/358790.358797
5. Farash MS, Attari M. A secure and efficient identity-based authenticated key exchange protocol for mobile client-server networks. The Journal of Supercomputing. 2014; 69(1): 395–411. doi: 10.1007/s11227-014-1170-5
6. Xiong H, Chen Z, Li FG. New identity-based three-party authenticated key agreement protocol with provable security. Journal of Network and Computer Applications. 2013; 36(2): 927–932. doi: 10.1016/j.jnca.2012.10.001
7. Xie Q, Hu B, Dong N, Wong DS. Anonymous three-party password-authenticated key exchange scheme for telecare medical information systems. PLoS ONE. 2014; 9(7): e102747. doi: 10.1371/journal.pone.0102747
8. Du WB, Wu ZX, Cai KQ. Effective usage of shortest paths promotes transportation efficiency on scale-free networks. Physica A. 2013; 392(17): 3505–3512. doi: 10.1016/j.physa.2013.03.032
9. Li XW, Zhang YQ, Zhang GF. A new certificateless authenticated key agreement protocol for SIP with different KGCs. Security and Communication Networks. 2013; 6(5): 631–643. doi: 10.1002/sec.595
10. Kounga G, Mitchell CJ, Walter T. Generating certification authority authenticated public keys in ad hoc networks. Security and Communication Networks. 2012; 5(1): 87–106. doi: 10.1002/sec.279

11. Mishra D, Kumari S, Khan MK, Mukhopadhyay S. An anonymous biometric-based remote user-authenticated key agreement scheme for multimedia systems. Journal International Journal of Communication Systems. 2015. doi: 10.1002/dac.2946

12. Ustaoglu B. Integrating identity-based and certificate-based authenticated key exchange protocols. International Journal of Information Security. 2011; 10(4): 201–212. doi: 10.1007/s10207-011-0136-3

13. Lu YR, Li LX, Peng HP, Yang YX. A biometrics and smart cards-based authentication scheme for multi-server environments. Security and Communication Networks. 2015. doi: 10.1002/sec.1246

14. Chang YF, Tai WL, Chang HC. Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update. International Journal of Communication Systems. 2014; 27(11): 3430–3440. doi: 10.1002/dac.2552

15. Mishra D, Das AK, Chaturvedic A, Mukhopadhyay S. A secure password-based authentication and key agreement scheme using smart cards. Journal of Information Security and Applications. 2015; 23: 28–43. doi: 10.1016/j.jisa.2015.06.003

16. Huang H, Cao ZF. IIDOAKE: Strongly secure ID-based one-pass authenticated key exchange protocol. Security and Communication Networks. 2011; 4(10): 1153–1161. doi: 10.1002/sec.241

17. Shamir A. Identity-based cryptosystems and signature schemes. Advances in Cryptology. 1985; 196: 47–53. doi: 10.1007/3-540-39568-7_5

18. He DB, Chen JH, Hu J. An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security. Information Fusion. 2012; 13(3): 223–230. doi: 10.1016/j.inffus.2011.01.001

19. Yang JH, Chang CC. An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. Computers & Security. 2009; 28(3-4): 138–143. doi: 10.1016/j.cose.2008.11.008

20. Hsu CL, Chuang YH, Kuo CL. A Novel Remote User Authentication Scheme from Bilinear Pairings Via Internet. Wireless Personal Communications. 2015; 83(1): 163–174. doi: 10.1007/s11277-015-2386-2

21. Yoon EJ, Yoo KY. Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. The Journal of Supercomputing. 2013; 63(1): 235–255. doi: 10.1007/s11227-010-0512-1

22. Islam SKH. A provably secure ID-based mutual authentication and key agreement scheme for mobile multi-server environment without ESL attack. Wireless Personal Communications. 2014; 79(3): 1975–1991. doi: 10.1007/s11227-014-0196-8

23. Baruah KC, Banerjee S, Dutta MP, Bhunia CT. An improved biometric-based multi-server authentication scheme using smart card. International Journal of Security and Its Applications. 2015; 9(1): 397–408. doi: 10.14257/ijisia.2015.9.1.38

24. Lu Y, Li L, Yang X, Yang Y. Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. PLoS ONE. 2015; 10(5): e0126323. doi: 10.1371/journal.pone.0126323 PMID: 26040884

25. Xiong H, Qin ZG. Revocable and Scalable Certificateless Remote Authentication Protocol With Anonymity for Wireless Body Area Networks. IEEE Transactions on Information Forensics and Security. 2015; 10(7): 1442–1455. doi: 10.1109/TIFS.2015.2414399

26. Nam J, Choo KKR, Han S, Kim M, Paik J, Won D. Efficient and anonymous two-factor user authentication in wireless sensor networks: achieving user anonymity with lightweight sensor computation. PLoS ONE. 2015; 10(4): e0116709. doi: 10.1371/journal.pone.0116709 PMID: 25849359

27. Cao LL, Ge WC. Analysis and improvement of a multi-factor biometric authentication scheme. Security and Communication Networks. 2015; 8(4): 617–625. doi: 10.1002/sec.1010

28. Mishra D, Mukhopadhyay S. Cryptanalysis of pairing-free identity-based authenticated key agreement. Information Systems Security. 2013; 8303: 247–254. doi: 10.1007/978-3-642-45204-8_19

29. Sun HM, Leu MC. An efficient authentication scheme for access control in mobile pay-TV systems. IEEE Transactions on Multimedia. 2009; 11(5): 947–959. doi: 10.1109/TMM.2009.2021790

30. Ku WC, Chen SM. Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics. 2004; 50(1): 204–207. doi: 10.1109/TCE.2004.1277863

31. Mishra D. Understanding security failures of two authentication and key agreement schemes for telecare medicine information systems. Journal of medical systems. 2015; 39(3): 1–8. doi: 10.1007/s10916-015-0193-7
32. Hsiang HC, Shih WK. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces. 2009; 31(6): 1118–1123. doi: 10.1016/j.csi.2008.11.002

33. Leung KC, Cheng LM, Fong AS, Chan CK. Cryptanalysis of a modified remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics. 2003; 49(4): 1243–1245. doi: 10.1109/TCE.2003.1261224

34. Ma CG, Wang D, Zhao SD. Security flaws in two improved remote user authentication schemes using smart cards. International Journal of Communication Systems. 2014; 27(10): 2215–2227. doi: 10.1002/dac.2468

35. Mishra D. On the security flaws in id-based password authentication schemes for telecare medical information systems. Journal of medical systems. 2015; 39(1): 1–16. doi: 10.1007/s10916-014-0154-6

36. Messerges TS, Dabbish EA, Sloan RH. Examining smart-card security under the threat of power analysis attacks. IEEE Transactions on Computers. 2002; 51(5): 541–552. doi: 10.1109/TC.2002.1004593

37. Das AK. Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards. IET Information Security. 2011; 5(3): 145–151. doi: 10.1049/iet-ifs.2010.0125

38. Islam SKH. Provably secure dynamic identity-based three-factor password authentication scheme using extended chaotic maps. Nonlinear Dynamics. 2014; 78(3): 2261–2276. doi: 10.1007/s11071-014-1584-x

39. Zhang M, Zhang JS, Zhang Y. Remote three-factor authentication scheme based on fuzzy extractors. Security and Communication Networks. 2015; 8(4): 682–693. doi: 10.1002/sec.1016

40. Li CT, Hwang MS. An efficient biometrics-based remote user authentication scheme using smart cards. Journal of Network and Computer Applications. 2010; 33(1): 1–5. doi: 10.1016/j.jnca.2009.08.001

41. Li X, Niu JW, 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. 2011; 34(1): 73–79. doi: 10.1016/j.jnca.2010.08.003

42. Farid B, Kadda BB. Password hardened fuzzy vault for fingerprint authentication system. Image and Vision Computing. 2014; 32(8): 487–496. doi: 10.1016/j.imavis.2014.04.014

43. Dodis Y, Reyzin L, Smith A. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. Advances in Cryptology—EUROCRYPT 2004. 2004; 3027: 523–540. doi: 10.1007/978-3-540-24676-3_31

44. Dodis Y, Kanukurthi B, Katz J, Reyzin L, Smith A. Robust Fuzzy Extractors and Authenticated Key Agreement From Close Secrets. IEEE Transactions on Information Theory. 2012; 58(9): 6207–6222. doi: 10.1109/TIT.2012.2200290

45. He DB, Wang D. Robust Biometrics-Based Authentication Scheme for Multi-server Environment. IEEE Systems Journal. 2015; 9(3): 816–823. doi: 10.1109/JSYST.2014.2301517

46. Zhang JS, Ma J, Li X, Wang WD. A secure and efficient remote user authentication scheme for multi-server environments using ECC. KSII Transactions on Internet and Information Systems. 2014; 8(8): 2930–2947. doi: 10.3837/tiis.2014.08.021

47. Liao YP, Wang SS. A secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces. 2009; 31(1): 24–29. doi: 10.1016/j.csi.2008.11.002

48. Yoon EJ, Yoo KY. Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. The Journal of Supercomputing. 2013; 63(1): 235–255. doi: 10.1007/s11227-010-0512-1

49. Zhu HF. A provable one-way authentication key agreement scheme with user anonymity for multi-server environment. KSII Transactions on Internet and Information Systems. 2015; 9(2): 811–829. doi: 10.3837/tiis.2015.02.019

50. Li X, Niu JW, Kumari S, Liao JG, Liang W. An enhancement of a smart card authentication scheme for multi-server architecture. Wireless Personal Communications. 2015; 80(1): 175–192. doi: 10.1007/s11227-014-2002-x

51. Chuang MC, Chen MC. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. Expert Systems with Applications. 2014; 41(4): 1411–1418. doi: 10.1016/j.eswa.2013.08.040

52. Choi YS, Nam JH, Lee DH, Kim JY, Jung JW, Won DH. Security enhanced anonymous multi-server authenticated key agreement scheme using smart cards and biometrics. The Scientific World Journal. 2014; 281305. doi: 10.1155/2014/281305 PMID: 25276847

53. 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
54. 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

55. Kocher P, Jaffe J, Jun B, Rohatgi P. Introduction to differential power analysis. Journal of Cryptographic Engineering. 2011; 1(1): 5–27. doi: 10.1007/s13389-011-0006-y

56. Dang Q. Changes in Federal Information Processing Standard (FIPS) 180-4, secure hash standard. Cryptologia. 2013; 37(1): 69–73. doi: 10.1080/01611194.2012.687431

57. Manuel S. Classification and generation of disturbance vectors for collision attacks against SHA-1. Designs, Codes and Cryptography. 2011; 59(1-3): 247–263. doi: 10.1007/s10623-010-9458-9

58. Zhu HF, Hao X. A provable authenticated key agreement protocol with privacy protection using smart card based on chaotic maps. Nonlinear Dynamics. 2015; 81(1-2): 311–321. doi: 10.1007/s11071-015-1993-5

59. Xue KP, Hong PL, Ma CS. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. Journal of Computer and System Sciences. 2014; 80(1): 195–206. doi: 10.1016/j.jcss.2013.07.004

60. Li X, Ma J, Wang WD, Xiong YP, Zhang JS. A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments. Mathematical and Computer Modelling. 2013; 58(1-2): 85–95. doi: 10.1016/j.mcm.2012.06.033

61. Xue KP, Hong PL. Security improvement on an anonymous key agreement protocol based on chaotic maps. Communications in Nonlinear Science and Numerical Simulation. 2012; 17(7): 2969–2977. doi: 10.1016/j.cnsns.2011.11.025