Cryptanalysis and improvement of a biometrics-based authentication and key agreement scheme for multi-server environments

Li Yang¹,², Zhiming Zheng¹,²

¹ Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education, Beihang University, Beijing, China, ² School of Mathematics and Systems Science, Beihang University, Beijing 100191, China

* yangli73@buaa.edu.cn (LY); zzheng@pku.edu.cn (ZMZ)

Abstract

According to advancements in the wireless technologies, study of biometrics-based multi-server authenticated key agreement schemes has acquired a lot of momentum. Recently, Wang et al. presented a three-factor authentication protocol with key agreement and claimed that their scheme was resistant to several prominent attacks. Unfortunately, this paper indicates that their protocol is still vulnerable to the user impersonation attack, privileged insider attack and server spoofing attack. Furthermore, their protocol cannot provide the perfect forward secrecy. As a remedy of these aforementioned problems, we propose a biometrics-based authentication and key agreement scheme for multi-server environments. Compared with various related schemes, our protocol achieves the stronger security and provides more functionality properties. Besides, the proposed protocol shows the satisfactory performances in respect of storage requirement, communication overhead and computational cost. Thus, our protocol is suitable for expert systems and other multi-server architectures. Consequently, the proposed protocol is more appropriate in the distributed networks.

Introduction

Tremendous advancements in the wireless technologies enhance the quality of on-line services in the distributed networks. It makes plenty of web users enjoy a variety of helpful on-line services in many aspects, for example, on-line work, on-line medicine, on-line shopping and so on [1, 2]. However, there remains a significant problem, namely, how to help web users enjoy so many on-line services while ensuring the confidentiality of their sensitive datas over an insecure channel. Thus, data protection becomes more and more important for every communication participant in the distributed networks. As a remedy, authenticated key establishment protocols are applied for safeguarding the information and defying the threats, which help web users submit their credentials and acquire various on-line services from a number of remote network servers subsequently [3, 4]. Specifically, mutual authentication that makes network
servers check the legality of web users and vice-versa minimizes the risk of internet fraud. As a next step, key agreement helps communication participants establish a common session key to ensure their subsequent communication in the open networks [5].

Over the four decades, there are three kinds of typical factors to design an authenticated key establishment protocol, that is, knowledge factor (password), possession factor (smart card) and inherence factor (biometric information), respectively [6–9]. In last few years, Khan [10] presented two biometric-based authentication schemes which possessed the self-authentication and deniability, respectively. In 2013, Kumari and Khan [11] put forward an improved smart card-based authentication protocol with user anonymity for remote users. In recent years, Farash et al. [12] proposed a lightweight authentication scheme which was applied for consumer roaming. Over the last two years, Kumari et al. [13] presented a smart card-based authentication protocol for session initiation service.

More specifically, Lamport [7] put forward the first authentication scheme which was based on password and was unable to provide the key agreement in 1981. However, his protocol maintained some password-verification tables that made stolen verification tables attack feasible. Afterwards, a sequence of improved password-based authentication and key establishment schemes have been presented [14–16]. There are some common shortcomings in these authenticated key exchange protocols which only adopt the password, such as, weak password, dictionary attack, stolen verification tables attack and so on. Thus, it is necessary to add the possession factor to design a novel kind of authenticated key agreement schemes, which makes them more robust [17–19].

Later on, two-factor authentication and key establishment protocols which apply both password and smart card have been deployed widely in the distributed networks. In order to log in the expected remote network servers, web users need to insert their smart card into a smart card reader and enter their password. In 1991, Chang et al. [20] presented a password-based authentication scheme with smart card. Since then, a series of cryptanalysis and improvements have been put forward [21–25]. However, it is practicable to acquire some datas stored in the smart card through side channel attacks [26]. Therefore, a lost or stolen smart card makes authenticated key agreement protocols vulnerable [27–30].

In order to solve these aforementioned problems, biometric information (e.g. facial expressions, retina and finger prints and so on) as an inherence factor has been added to propose a variety of three-factor authenticated key establishment protocols. Different from knowledge factor and possession factor, biometric information which possesses the uniqueness further enhances the security of sensitive datas [31, 32]. Besides, it is exceedingly difficult for adversary to forge the biometrics of web users. Also it does not request web users to remember their biometric information which is hard to be forgotten or lost. Thus, biometric information is combined with both password and smart card mentioned above to make a battery of three-factor authenticated key agreement schemes appear [33–38]. In practice, biometric datas imprinted by web users are not the same each time so that directly adopting them usually results in a low success rate for valid web users [39]. To meet this problem, biometric-based fuzzy extractor which is convenient to be implemented by a smart card is introduced to reduce the failure rate [40]. Besides, Bio-Hash code, namely, user specific code is another way to accommodate this problem [41].

Furthermore, earlier authentication and key establishment protocols are only applied for single-server environments, which don’t consider the applicability of multi-server environments. Specifically, it is inefficient for single-server authentication schemes to be directly adopted in the multi-server environments. With a rapid augmentation of different network servers, web users not only register and login each individual server repeatedly, but also maintain massive credentials about identities and passwords. In 2001, Li et al. [42] put forward the
first multi-server authenticated protocol which coped up with this problem mentioned above. In particular, Li et al. [42] efficiently applied a registration center to achieve the single registration in the multi-server architectures. During the past two decades, a large amount of multi-server authentication schemes have been presented, in which some protocols adopt the two-factor [43–46] and others are based on three-factor [47–56].

The multi-server authentication mechanism requires the higher security. Since legal users adopt the same credentials to log into a variety of individual network servers, it is practical for adversaries to make many protocols vulnerable to the user impersonation attack, privileged insider attack and server spoofing attack by tracing web users [47, 57, 58]. As typical multi-server architectures, expert systems which benefit from decision-making capability of human experts have a great deal of applications, for example, security auditing and network management. Particularly, Tsudik and Summers [59] introduced an security auditing expert system called AudES which automated a great deal of manual security auditing procedures in order to alleviate the burden of human auditors. For network management expert systems, Hariri and Jabbour [60] designed a generalized architecture to manage plenty of resources in a distributed computer network. Recently, Mishra et al. [50] put forward an anonymous three-factor multi-server authenticated scheme with key agreement for expert systems which was adopted to ensure the communications between web user and network server. They declared that their protocol provided a high security. However, Wang et al. [61] indicated that Mishra et al.’s scheme was vulnerable to several common attacks and presented an improved protocol to enhance the security. Unfortunately, due to cryptanalysis described below, we claim that Wang et al.’s scheme is still vulnerable to the user impersonation attack, privileged insider attack and server spoofing attack. Besides, their scheme fails to provide the perfect forward secrecy.

As a remedy of these aforementioned problems, we propose a biometric-based authentication and key agreement protocol for multi-server architectures in order to ensure the confidentiality of sensitive data while web user enjoys some decision-making services, such as security auditing and network management in the expert systems. When web user wants to login the network server to acquire these services, our protocol is performed between web user and network server. Concretely, web user submits his login request message to network server. Next, network server tries to authenticate web user with the message received from web user and the beforehand information saved during the registration phase. Also network server issues his authentication request message to web user. Then, web user tries to authenticate network server in a similar way and delivers his authentication reply to network server. Finally, web user and network server apply our protocol to achieve the mutual authentication and key agreement. Compared with other related schemes, our protocol achieves the stronger security and provides more functionality properties. Besides, the presented protocol requires the lower computational cost and shows a satisfactory performance on the communication overhead with the same level of storage requirement. Thus, the proposed protocol is suitable for expert systems and other multi-server architectures, such as, on-line medicine systems, on-line shopping systems and so on. Above all, our protocol is more appropriate in the distributed networks.

The remaining of this paper is organized in seven sections as below. Next section introduces the collision-resistant hash function, threat assumptions and biometrics-based fuzzy extractor, respectively. Section 3 reviews Wang et al.’s scheme. Section 4 discusses some weaknesses of Wang et al.’s scheme. Section 5 describes the proposed biometrics-based authenticated key agreement protocol in details. And then section 6 provides the security analysis, functionality analysis and efficiency analysis of our protocol, and compares our protocol with others in these aforementioned respects. Last section gives the conclusion.
Preliminaries

During this section, we briefly describe some concepts relating to collision-resistant hash function, threat assumptions and biometrics-based fuzzy extractor as follows.

Collision-resistant hash function

According to an arbitrary length binary string, collision-resistant hash function outputs a fixed-length binary string, that is, $h = h(x)$ : $0, 1^* \rightarrow 0, 1^n$ [62]. Furthermore, retrieving this arbitrary length input from a given output is computationally infeasible. Thus, collision resistant property is explained as below. For a given input $x$, it is computationally infeasible to find any input $y \neq x$ makes $h(x) = h(y)$.

Threat assumptions

During this subsection, we introduce some common threat assumptions which includes the Dolev-Yao threat model [63] and the risk of side-channel attacks [27]. More details about these threat assumptions are described as below.

1. Adversary $E$ might be a malicious user or an outside hacker.
2. Adversary $E$ has an ability to eavesdrop all communication messages between participants via an open channel.
3. Adversary $E$ can modify, delete, resend and reroute all eavesdropped messages.
4. Adversary $E$ is able to extract all stored datas from a lost or stolen smart card by examining the power consumption.

Biometrics-based fuzzy extractor

We briefly introduce the mechanism of biometrics-based fuzzy extractor in this subsection. A biometrics-based fuzzy extractor which converts the biometric information into two available and unpredictable values consist of two procedures, namely, $Gen$ and $Rep$ [40]. More specifically, details about this mechanism are illustrated in Fig 1. Based on the biometric information $BIO$, procedure $Gen$ which is a probabilistic generation function outputs an unpredictable binary string $R \in \{0, 1\}^l$ and an auxiliary binary string $P \in \{0, 1\}^*$. With the help of this

![Fig 1. The mechanism of fuzzy extractor.](https://doi.org/10.1371/journal.pone.0194093.g001)
auxiliary string $P$ and another biometric information $BIO^*$, procedure $Rep$ which is a deterministic reproduction function recovers a corresponding unpredictable binary string $R$. When $\text{Gen}(BIO) \rightarrow \langle R, P \rangle$ and $\text{dis}(BIO, BIO^*) \leq t$ hold, then we have $\text{Rep}(BIO^*, P) \rightarrow R$. Otherwise, there is no output provided by procedure $Rep$. Furthermore, error-tolerant makes it more robust to recover a corresponding unpredictable binary string $R$, as long as this biometric information $BIO^*$ keeps reasonable close to an initial biometrics $BIO$.

Since biometric features vary slightly at every imprint, another way to extract the biometric features is applying the Bio-Hash codes. In recent times, many Bio-Hashing authentication schemes with key agreement are presented [41, 64, 65]. Similarly, Bio-Hashing is also a convenient technique, which is usable in many small devices.

### Review of Wang et al.’s scheme

During this section, we review Wang et al.’s biometrics-based authentication and key agreement scheme for multi-server environments which is described in Ref. [61]. Their scheme includes six phases, namely, server registration phase, user registration phase, login phase, authentication phase, password change phase and user revocation/re-registration phase. There are the following three participants in their scheme, that is, registration center $RC$, server $S_j$ and user $U_i$. Suppose that registration center $RC$ is a trusted third party. In Wang et al.’s scheme, registration center $RC$ is responsible for user registration and server registration. For convenience, symbols and corresponding notions which are applied in their scheme are respectively shown in Table 1.

#### Server registration phase
1. Server $S_j$ submits a join request message to registration center $RC$, which helps server $S_j$ become an authorized server in the expert system.
2. Upon receiving this join request message, registration center $RC$ sends server $S_j$ a pre shared key $PSK$ to server $S_j$ over a secure channel.
User registration phase

1. Firstly, user $U_i$ imprints his personal biometric information $BIO_i$ at a sensor. Then sensor sketches $BIO_i$ to extract an unpredictable binary string $R_i$ and an auxiliary binary string $P_i$ from $Gen(BIO_i) \rightarrow (R_i, P_i)$. After that, sensor stores this corresponding auxiliary string $P_i$ in the memory. Next, user $U_i$ enters his identity $ID_i$ and password $PW_i$, and calculates $RPW_i = h(PW_i||R_i)$. Finally, user $U_i$ issues his registration request message $\{ID_i, RPW_i\}$ to registration center $RC$ through a secure channel.

2. Upon obtaining this registration request message, registration center $RC$ adds a novel entry $(ID_i, N_i = 1)$ to an internal database for user $U_i$, in which $N_i$ stands for the times of user registration. And then registration center $RC$ successively calculates $A_i = h(ID_i||x||T_i)$, $B_i = RPW_i \oplus h(A_i)$, $C_i = B_i \oplus h(PSK)$, $D_i = PSK \oplus A_i \oplus h(PSK)$ and $V_i = h(ID_i||RPW_i)$, where $T_i$ is registration time.

3. Registration center $RC$ sends user $U_i$ a smart card $SC_i$, which contains $\{B_i, C_i, D_i, V_i\}$ via a secure channel.

4. After receiving his smart card $SC_i$, user $U_i$ stores his auxiliary string $P_i$ mentioned above into his smart card $SC_i$.

Login phase

1. User $U_i$ inserts his smart card $SC_i$ into the smart card reader. Then he inputs his identity $ID_i$ and password $PW_i$. Next, user $U_i$ imprints his biometric information $BIO_i$ at a sensor. After that, sensor sketches user $U_i$’s biometric information $BIO_i$ and recovers the unpredictable binary string $R_i$ from $Rep(BIO_i, P_i) \rightarrow R_i$.

2. Smart card $SC_i$ computes $RPW_i = h(PW_i||R_i)$ and checks whether $h(ID_i||RPW_i) = V_i$ is valid. If it is valid, smart card $SC_i$ further computes $h(PSK) = B_i \oplus C_i$.

3. Smart card $SC_i$ generates a random number $N_i$ to calculate $AID_i = ID_i \oplus h(N_i)$, $M_1 = RPW_i \oplus N_i \oplus h(PSK)$ and $M_2 = h(AID_i||N_i||RPW_i||SID_i||T_i)$, in which $T_i$ is an additional timestamp.

4. Smart card $SC_i$ delivers user $U_i$’s login request message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$ to server $S_j$ over an open channel.

Authentication phase

1. Upon receiving user $U_i$’s login request message, server $S_j$ verifies whether $T_i - T_j \leq \Delta T$ holds, in which $\Delta T$ is a suitable time interval and $T_j$ is the time when server $S_j$ obtains user $U_i$’s login request message. If this verification holds, server $S_j$ continues to execute his next step. Otherwise, user $U_i$’s login request is rejected by server $S_j$.

2. Server $S_j$ retrieves $A_i = D_i \oplus PSK \oplus h(PSK)$, $RPW_i = B_i \oplus h(A_i)$ and $N_i = RPW_i \oplus M_1 \oplus h(PSK)$ in order to check whether $h(AID_i||N_i||RPW_i||SID_i||T_i)$ is consistent with $M_2$.

3. If it holds, server $S_j$ generates a random number $N_2$ to calculate their session secret key $SK_j = h(AID_i||SID_j||N_i||N_2)$.

4. Server $S_j$ computes $M_3 = N_2 \oplus h(AID_i||N_i) \oplus h(PSK)$ and $M_4 = h(SID_j||N_2||AID_i)$ in order to send his authentication request message $\{SID_j, M_3, M_4\}$ to user $U_i$ through an open channel.

5. After receiving server $S_j$’s authentication request message, smart card $SC_i$ retrieves $N_2 = M_3 \oplus h(AID_i||N_i) \oplus h(PSK)$ and $SK_j = h(AID_i||SID_j||N_i||N_2)$ to verify whether $h(SID_j||N_2||AID_i) = M_4$ holds. If it holds, smart card $SC_i$ calculates $M_5 = h(SK_j||N_i||N_2)$ in order to submit user $U_i$’s authentication reply $\{M_5\}$ to server $S_j$ over an open channel.
6. Server \( S_i \) checks whether \( h(SK_i || N_i || N_c) = M_i \) is valid. If this verification is valid, server \( S_i \) further applies this session key \( SK_{ij} \) to communicate with user \( U_i \) in the following communication. Otherwise, authentication phase is rejected by server \( S_i \).

**Password change phase**

1. User \( U_i \) enters his identity \( ID_i \) and password \( PW_{i} \) and imprints his biometric information \( BIO_i \) at a sensor. After that, sensor sketches user \( U_i \) ’s biometric information \( BIO_i^s \) and recovers the unpredictable binary string \( R_i \) from \( Rep(BIO_i^s, P_s) \rightarrow R_i \).

2. Smart card \( SC_i \) computes \( RPW_i = h(PW_i || R_i) \) and verifies whether \( h(ID_i || RPW_i) = V_i \) is valid. If this verification is valid, smart card \( SC_i \) asks user \( U_i \) for a new password. Otherwise, password change phase is terminated immediately by smart card \( SC_i \).

3. User \( U_i \) enters his new password \( PW_{i}^{\text{new}} \) and smart card \( SC_i \) further calculates \( RPW_{i}^{\text{new}} = h(PW_{i}^{\text{new}} || R_i) \), \( B_i^{\text{new}} = B_i \oplus RPW_i \oplus RPW_{i}^{\text{new}} \), \( C_i^{\text{new}} = C_i \oplus RPW_i \oplus RPW_{i}^{\text{new}} \) and \( V_{i}^{\text{new}} = h(ID_i || RPW_{i}^{\text{new}}) \).

4. In the memory, smart card \( SC_i \) respectively replaces \( B_i \) with \( B_i^{\text{new}} \), \( C_i \) with \( C_i^{\text{new}} \) and \( V_i \) with \( V_{i}^{\text{new}} \).

**User revocation/re-registration phase**

1. When user \( U_i \) wants to revoke his privilege, he submits a revocation request message, his smart card \( SC_i \) and verification message \( RPW_i \) to registration center \( RC \) via a secure channel. Registration center \( RC \) checks whether user \( U_i \) is valid. If user \( U_i \) is valid, registration center \( RC \) further modifies a corresponding entry by setting \( ID_i, N_i = 0 \).

2. Similarly, after receiving a re-registration request message through a secure channel, registration center \( RC \) performs these steps mentioned in the subsection 3.2 and replaces \( ID_i, N_i = N_i + 1 \) with \( ID_i, N_i \) to help user \( U_i \) re-register.

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

In this section, we propose a cryptanalysis of Wang et al.’s scheme. In particular, results demonstrate that their protocol is still vulnerable to the user impersonation attack, privileged insider attack and server spoofing attack. Furthermore, their scheme fails to achieve the perfect forward secrecy. More details of these problems are shown in the following subsections.

**User impersonation attack**

Suppose that adversary \( E \) is an outside hacker who steals user \( U_i \) ’s smart card \( SC_i \) and eavesdrops all communications between user \( U_i \) and server \( S_j \). Specifically, adversary \( E \) has an ability to extract the stored data \( \{B_i, C_i, D_i, V_i, P_i\} \) from user \( U_i \) ’s smart card \( SC_i \) by side-channel attacks. Also he is able to collect user \( U_i \) ’s login request message \( \{AID_i, M_1, M_2, B_i, D_i, T_j\} \). Thus Wang et al.’s scheme is vulnerable to user impersonation attack. More narrowly, adversary \( E \) can impersonate as a legal user so that he is authenticated by server \( S_j \). More details are explained as below.

1. Firstly, adversary \( E \) computes \( h(PSK) = B_i \oplus C_i \). Then he generates a random number \( N_i^* \) and further calculates \( B_i^* = B_i \oplus h(PSK) \), \( D_i^* = h(PSK) \), \( M_i^* = B_i \oplus N_i^* \oplus h(PSK) \) and \( M_i^* = h(AID_i || N_i^* || B_i || SID_i || T_i^* ) \), in which \( T_i^* \) is a current timestamp. Finally, adversary \( E \) delivers his login request message \( \{AID_i, M_i^*, M_i^*, B_i^*, D_i^*, T_i^*\} \) to server \( S_j \) over an open channel.

2. When obtaining this login request message from adversary \( E \), server \( S_j \) verifies whether \( T_i^* - T_j^* \leq \Delta T \) holds, where \( T_j^* \) is the time when server \( S_j \) receives adversary \( E \)’s login request
message. Thus adversary $E$ passes server $S_j$’s verification successfully and server $S_j$ continues to execute the subsequent steps normally.

3. Server $S_j$ retrieves $A_i = D_i \oplus PSK \oplus h(PSK)$, $RPW_i = B_i \oplus h(A_i) = B_i$ and $N_i = RPW_i \oplus M_i \oplus h(PSK)$ to check whether $h(AID_i||N_i||RPW_i||SID_i||T_{iE}) = M_i$ holds. Next server $S_j$ generates a random number $N_z$ and further calculate $SK_{j_1} = h(AID_i||SID_i||N_i||N_z)$, $M_i = N_2 \oplus h(AID_i||N_i) \oplus h(PSK)$ and $M_i = h(SID_i||N_2||AID_i)$. Lastly, server $S_j$ sends his authentication request message $\{SID_j, M_j, M_i\}$ to adversary $E$ through an open channel as usual.

4. Upon receiving server $S_j$’s authentication request message, adversary $E$ retrieves $N_i = M_i \oplus h(AID_i||N_i) \oplus h(PSK)$ and $SK_{j_1} = h(AID_i||SID_i||N_i||N_2)$ in order to calculate $M_i = h(SK_{j_1}||N_2||N_z)$ and submit his authentication reply $\{M_j\}$ to server $S_j$.

5. Server $S_j$ checks whether $h(SK_{j_1}||N_i||N_z) = M_j$ is valid.

Thus server $S_j$ authenticates adversary $E$ and they both apply the session key $SK_j$ in the following communication. Unfortunately, server $S_j$ mistakenly believes that he communicates with user $U_i$. Therefore Wang et al.’s scheme becomes vulnerable to the user impersonation attack.

### Privileged insider attack

As shown in this subsection, adversary $E$ who is a privileged insider can impersonate as user $U_i$ if he steals user $U_i$’s smart card $SC_i$ and eavesdrops all communications between user $U_i$ and registration center $RC$. Similarly, adversary $E$ is able to acquire these data sets $\{B_i, C_i, D_i, V_i, P_i\}$ from smart card $SC_i$. And he has an ability to collect user $U_i$’s registration request message $\{ID_i, RPW\}$. So Wang et al.’s scheme is also vulnerable to the privileged insider attack. More details are described as follows.

1. Firstly, adversary $E$ computes $h(PSK) = B_i \oplus C_i$ and generates a random number $N_{1E}$. Then he calculates $AID_{iE} = ID_i \oplus h(N_{1E})$, $M_{1E} = RPW_i \oplus N_{1E} \oplus h(PSK)$ and $M_{2E} = h(AID_{iE}||N_{1E}||RPW_i||SID_i||T_{iE})$, where $T_{iE}$ is a current timestamp. Lastly, adversary $E$ issues his login request message $\{ID_{iE}, M_{1E}, M_{2E}, B_i, D_i, T_{iE}\}$ to server $S_j$ over an open channel.

2. After acquiring this login request message, server $S_j$ verifies whether $T_{iE} - T_{iE} \leq \Delta T$ holds, where $T_{iE}$ is the time when server $S_j$ acquire adversary $E$’s login request message. Unfortunately, adversary $E$’s verification is valid.

3. Server $S_j$ retrieves $A_i = D_i \oplus PSK \oplus h(PSK)$, $RPW_i = B_i \oplus h(A_i)$ and $N_{1E} = RPW_i \oplus M_{1E} \oplus h(PSK)$ in order to verify whether $h(AID_{iE}||N_{1E}||RPW_i||SID_i||T_{iE})$ is consistent with $M_{2E}$. Then server $S_j$ generates a random number $N_{2E}$ and further calculates $SK_{jE} = h(AID_{iE}||SID_i||N_{1E}||N_{2E})$, $M_{3E} = N_{2E} \oplus h(AID_{iE}||N_{1E}) \oplus h(PSK)$ and $M_{4E} = h(SID_i||N_{2E}||AID_{iE})$. Finally, server $S_j$ submits his authentication request message $\{SID_i, M_{3E}, M_{4E}\}$ to adversary $E$ via an open channel without any suspicion.

4. When receiving server $S_j$’s authentication request message, adversary $E$ retrieves $N_{2E} = M_{3E} \oplus h(AID_{iE}||N_{1E}) \oplus h(PSK)$ and $SK_{jE} = h(AID_{iE}||SID_i||N_{1E}||N_{2E})$. Then he calculates $M_{5E} = h(SK_{jE}||N_{1E}||N_{2E})$ and sends his authentication reply $\{M_{5E}\}$ to server $S_j$.

5. Server $S_j$ checks whether $h(SK_{jE}||N_{1E}||N_{2E}) = M_{5E}$ holds as usual.

So server $S_j$ further applies the session key $SK_{jE}$ to communicate with adversary $E$ and authenticates adversary $E$ who is a privileged insider and impersonates as user $U_i$. Unfortunately, Wang et al.’s scheme is unable to resist the privileged insider attack.
Server spoofing attack

In this subsection, we suppose that adversary E who is an insider but isn’t another server $S_i$ has an ability to eavesdrop user $U_i$‘s registration request message $\{ID, RPW\}$ and steal user $U_i$‘s smart card $SC_i$. Furthermore, adversary E is able to collect some datas, for example, $\{B_n, C_n, D_n, V_i, P_i\}$. Thus adversary E can masquerade as server $S_j$ to cheat user $U_i$. Therefore Wang et al.’s scheme becomes vulnerable to the server spoofing attack. More details are shown as below.

1. Firstly, adversary E calculates $h(PSK) = B_i \oplus C_i$ and eavesdrops user $U_i$‘s login request message $\{AID, M_1, M_2, B_n, D_n, T_i\}$.
2. Secondly, adversary E computes $N_1 = RPW_i \oplus M_1 \oplus h(PSK)$ and generates a fresh random number $N_2^e$.
3. Next adversary E further computes $M_1^e = N_2^e \oplus h(AID) || N_1 \oplus h(PSK)$ and $M_2^e = h(SID) || N_2^e || AID$.
4. Finally adversary E issues his authentication request message $\{SID, M_1^e, M_2^e\}$ to user $U_i$ over a public channel.

Furthermore, this fake authentication request message is successfully checked. Particularly, adversary E is treated as server $S_j$ by user $U_i$ without any doubt. In conclusion, Wang et al.’s scheme can’t resist the server spoofing attack.

No perfect forward secrecy

During this subsection, we point out that Wang et al.’s scheme does not possess the perfect forward secrecy. Suppose that adversary E is a privileged insider who eavesdrops user $U_i$‘s registration request message $\{ID, RPW\}$ and steals user $U_i$‘s smart card $SC_i$. Particularly, adversary E can extract these datas which include $B_n, C_n, D_n, V_i$ and $P_i$ from smart card $SC_i$. More details are described as follows.

1. Firstly, adversary E computes $h(PSK) = B_i \oplus C_i$ and collects user $U_i$‘s login request message $\{AID, M_1, M_2, B_n, D_n, T_i\}$.
2. Secondly, adversary E calculates $N_1 = RPW_i \oplus M_1 \oplus h(PSK)$ and further collects server $S_j$‘s authentication request message $\{SID, M_1^e, M_2^e\}$.
3. Finally adversary E computes $N_2 = M_1 \oplus h(AID) || N_1 \oplus h(PSK)$ in order to retrieve $SK_{ij} = h(AID) || SID || N_1 || N_2$.

Therefore it is demonstrated that Wang et al.’s scheme is unable to achieve the perfect forward secrecy.

The proposed scheme

During this section, we propose a novel biometrics-based authentication and key agreement scheme for multi-server environments which is based on cryptanalysis of Wang et al.’s scheme. Our protocol is built by applying the collision-resistant hash function, EOR operation and concatenation operation. The presented scheme consists of six phases, namely, server registration phase, user registration phase, login phase, authentication phase, password change phase and user revocation/re-registration phase. And there are three participants in our algorithm, that is, registration center $RC$, server $S_j$ and user $U_i$. In our protocol, server $S_j$ and user $U_i$ are able to join the network by registering with registration center $RC$. Besides, mutual authentication only carries out between server $S_j$ and user $U_i$ without intervening registration center $RC$. For convenience, symbols and corresponding notions which are applied in our scheme are respectively shown in Table 2.

In particular, our proposed scheme enhances Wang et al.’s scheme in these aspects: 1) it resists the user impersonation attack, 2) it prevents the privileged insider attack, 3) it is secure
against the server spoofing attack and 4) it provides the perfect forward secrecy. More details are described in these following subsections.

Server registration phase
New server \( S_j \) needs to execute the server registration phase with registration center \( RC \) through a secure channel. More specifically, server registration phase of the proposed scheme is shown in the Fig 2 and details are described as below.

1. If it wants to be an authorized server in the multi-server environment, server \( S_j \) issues a join request message to registration center \( RC \).

2. When obtaining this join request message, registration center \( RC \) authorizes server \( S_j \) and replies with a pre shared key \( PSK \) and a master secret key \( s \) to server \( S_j \) by applying the Key Exchange Protocol (IKEv2) via a secure channel.

3. After receiving a pre shared key \( PSK \) and a master secret key \( s \), authorized server \( S_j \) adopts these shared datas, such as \( PSK \) and \( h(PSK) \), to verify user \( U_i \)’s legitimacy in the authentication phase.

Table 2. Symbols and corresponding notions in our scheme.

| Symbol | Notion                      |
|--------|-----------------------------|
| \( RC \) | Registration center        |
| \( S_j \) | \( j \)th server           |
| \( U_i \) | \( i \)th user             |
| \( SC_i \) | User \( U_i \)’s smart card |
| \( ID_i \) | User \( U_i \)’s identity   |
| \( PW_i \) | User \( U_i \)’s password   |
| \( BIO_i \) | User \( U_i \)’s biometric information |
| \( R_i \) | User \( U_i \)’s unpredictable binary string |
| \( P_i \) | User \( U_i \)’s auxiliary binary string |
| \( SID_j \) | Server \( S_j \)’s identity |
| \( PSK \) | Pre shared key             |
| \( s \) | Master secret key          |
| \( h(\cdot) \) | Collision-resistant hash function |
| \( \oplus \) | XOR operation              |
| \( \| \) | Concatenation operation    |

https://doi.org/10.1371/journal.pone.0194093.t002

Fig 2. The server registration phase.

https://doi.org/10.1371/journal.pone.0194093.g002
New user $U_i$ enters $BIO_i$.

$Gen(BIO_i)\rightarrow (R_i, P_i)$

chooses $ID_i$, $PW_i$

calculates $RPW_i = h(R_i || PW_i)$

adds a novel entry $<ID_i, N_i = 1>$

selects a random number $u_i$

computes $A_i = h(ID_i || s)$

$B_i = h(PSK) \oplus u_i$

$C_i = h(PSK || u_i) \oplus ID_i$

$V_i = h(ID_i || RPW_i)$

stores $\{A_i, B_i, C_i, V_i, h(\cdot)\}$ into smart card $SC_i$

computes $E_i = B_i \oplus h(R_i)$

replaces $B_i$ with $E_i$

stores $P_i$

Fig 3. The user registration phase.

User registration phase

New user $U_i$ should perform the user registration phase with registration center $RC$ over a secure channel. As details, user registration phase of ours is illustrated in the Fig 3 and explained as follows.

1. Firstly, user $U_i$ enters his personal biometric information $BIO_i$ at a sensor. And then, sensor sketches user $U_i$’s biometrics $BIO_i$, extracts $(R_i, P_i)$ from $Gen(BIO_i) \rightarrow (R_i, P_i)$, and stores user $U_i$’s auxiliary binary string $P_i$ in the memory. Next, user $U_i$ chooses his identity $ID_i$ and password $PW_i$, and calculates $RPW_i = h(R_i || PW_i)$. Finally, user $U_i$ submits his registration request message $\{ID_i, RPW_i\}$ to registration center $RC$ through a secure channel.

2. Upon obtaining this registration request message, registration center $RC$ adds a novel entry $<ID_i, N_i = 1>$ to his internal database, in which $N_i$ denotes the times of user registration for user $U_i$. Then registration center $RC$ selects a random number $u_i$, and calculates $A_i = h(ID_i || s)$, $B_i = h(PSK) \oplus u_i$, $C_i = h(PSK || u_i) \oplus ID_i$ and $V_i = h(ID_i || RPW_i)$.

3. Registration center $RC$ sends user $U_i$’s smart card $SC_i$ which includes $\{A_i, B_i, C_i, V_i, h(\cdot)\}$ via a secure channel.

4. After receiving this smart card $SC_i$, user $U_i$ computes $E_i = B_i \oplus h(R_i)$ and replaces $B_i$ with $E_i$. Finally, $U_i$ stores his auxiliary binary string $P_i$ into his smart card $SC_i$ and initializes the login and authentication environments.

Login phase

In the login phase, smart card $SC_i$ is able to find the errors immediately by applying user $U_i$’s identity, password, and biometric information. Specifically, login phase is shown in the Fig 4 and details are described as follows.
During the authentication phase, server S_i inserts his smart card SC_i into a smart card reader, enters his identity ID_i and password PW_i, and imprints their session secret key K_i. Then, sensor sketches user U_i’s personal biometric information BIO_i at a sensor, and recovers R_i from Rep(BIO_i, P_i) → R_i, with the assistance of auxiliary binary string P_i.

2. Smart card SC_i computes \( RPW_i = h(R_i||PW_i) \) and verifies whether \( h(ID_i||RPW_i) = V_i \) is valid. If it is valid, smart card SC_i further computes \( K_i = h(SID_i||ID_i \oplus C_i) \).

3. Smart card SC_i generates a random number \( R_i \), and calculates \( M_i = N_i \oplus K_i, M_2 = ID_i \oplus K_i, M_3 = RPW_i \oplus K_i, B_i = E_i \oplus h(R_i) \) and \( D_i = h(N_i||RPW_i||A_i||T_i) \), in which \( T_i \) is an additional timestamp.

4. Smart card SC_i submits his login request message \{M_1, M_2, M_3, B_i, D_i, T_i\} to server S_j over an open channel.

**Authentication phase**

During the authentication phase, server S_j has an ability to confirm the destination and freshness of login request message. More details, authentication phase is illustrated in the Fig 5 and explained as below.

1. After receiving user U_i’s login request message, server S_j checks whether \( T_i - T_j \leq \Delta T \) holds, in which \( \Delta T \) is a suitable time interval and \( T_j \) is the time when server S_j receives user U_i’s login request message. If it holds, server S_j continues to perform the following steps. Otherwise, this login request is rejected by server S_j.

2. Server S_j retrieves \( M_i = B_i \oplus h(PSK), K_i = h(SID_i||h(PSK||u_i)), N_1 = K_i \oplus M_1, ID_i = K_i \oplus M_2, RPW_i = K_i \oplus M_3 \) and \( A_i = h(ID_i||s) \) to verify whether \( h(N_i||RPW_i||A_i||T_i) = D_i \) is valid.

3. If this verification is valid, server S_j generates another random number \( N_2 \), and calculates their session secret key \( SK_i = h(ID_i||SID_i||N_1||N_2) \) between user U_i and server S_j.

4. Server S_j computes \( M_4 = N_2 \oplus h(A_i||RPW_i||N_1) \) and \( M_5 = h(SID_i||N_1||N_2||ID_i) \), and issues his authentication request message \{M_4, M_5\} to user U_i through an open channel.

5. When obtaining server S_j’s authentication request message, smart card SC_i retrieves \( N_2 = h(A_i||RPW_i||N_1) \oplus M_4 \) and checks whether \( h(SID_i||N_1||N_2||ID_i) \) is consistent with \( M_5 \). If they are consistent, smart card SC_i calculates \( SK_i = h(ID_i||SID_i||N_1||N_2) \) and \( M_6 = h(SK_i||N_1||N_2) \).
And then smart card \( SC_i \) delivers his authentication reply \( \{M_6\} \) to server \( S_j \) over a public channel.

6. Server \( S_j \) further verifies whether \( h(N_1 || RPW_i || A_i || T_i) = D_i \) is valid. If it is valid, server \( S_j \) adopts this session key \( SK_{ij} \) to communicate with user \( U_i \) in the following communication. Otherwise, authentication will be rejected by \( S_j \).

## Password change phase

In the password change phase, user \( U_i \) is able to update his password without any help from server \( S_j \) or registration center \( RC \). More specifically, password change phase includes the following steps.

1. User \( U_i \) inputs his identity \( ID_i \) and password \( PW_i \) and imprints his biometrics \( BIO_i \) at a sensor. And then, sensor sketches user \( U_i \)'s personal biometric information \( BIO_i \) and recovers \( R_i \) from \( Rep(BIO_i, P_i) \rightarrow R_i \) with the assistance of auxiliary binary string \( P_i \).

2. Smart card \( SC_i \) computes \( RPW_i = h(R_i || PW_i) \) and verifies whether \( h(ID_i || RPW_i) = V_i \) is valid. If this verification holds, smart card \( SC_i \) asks user \( U_i \) for a new password. Otherwise, smart card \( SC_i \) terminates the password change phase immediately.

3. User \( U_i \) enters his new password \( PW_{i\text{new}} \), and smart card \( SC_i \) further calculates \( RPW_{i\text{new}} = h(R_i || PW_{i\text{new}}) \) and \( V_{i\text{new}} = h(ID_i || RPW_{i\text{new}}) \).

4. Smart card \( SC_i \) replaces \( V_i \) with \( V_{i\text{new}} \) without any help from server \( S_j \) or registration center \( RC \) in the memory.

Fig 5. The authentication phase.
https://doi.org/10.1371/journal.pone.0194093.g005
User revocation/re-registration phase

If his smart card $SC_i$ is stolen or lost, user revocation/re-registration helps user $U_i$ revoke his privilege or re-register which makes our scheme more robust in the functionality.

1. When user $U_i$ wants to revoke his privilege, he issues his revocation request message, smart card $SC_i$ and verification message $\{RPW_i\}$ to registration center $RC$ through a secure channel. Registration center $RC$ checks whether user $U_i$ is valid. If user $U_i$ is valid, registration center $RC$ further sets $\langle ID_i, N_i = 0 \rangle$ to modify the corresponding entry.

2. Similarly, after obtaining a re-registration request message over a secure channel, registration center $RC$ performs these steps mentioned in the subsection 5.2 and helps user $U_i$ re-register by replacing $\langle ID_i, N_i = N_i + 1 \rangle$ with $\langle ID_i, N_i \rangle$.

Analysis of the proposed scheme

In a multi-server architecture, there are three important requirements for an authentication and key agreement protocol, namely, security, functionality and efficiency. In this section, discussions are performed and results show that our scheme satisfies these requirements mentioned above. Furthermore we compare the proposed protocol with others in respect of security, functionality and efficiency, respectively.

Informal security analysis

Before the formal security analysis, we analyze the resistance of our scheme against these following attacks by informal security analysis. Remark that adversary $E$ has an ability assumed in the threat assumptions to execute these attacks described as follows.

Resistance to replay attack. The proposed scheme applies the timestamp and random nonce to endure the replay attack. Though adversary $E$ eavesdrops user $U_i$’s previous login request message $\{M_1, M_2, M_3, B_i, D_i, T_i\}$ and issues it to server $S_j$ as always, server $S_j$ checks the legality of this message by verifying the timeliness of timestamp $T_i$ and correctness of random nonce $N_1$ as below.

$$D_i = h(N_i || |RPW_i|||A_i|||T_i),$$

in which both timestamp $T_i$ and random nonce $N_1$ are different for each session. Thus adversary $E$ is rejected by server $S_j$. Therefore our protocol prevents the replay attack.

Resistance to Denial-of-Service attack. Adversary $E$ tries to diminish or eliminate server $S_j$’s capability by eavesdropping and repeatedly sending user $U_i$’s previous login request message. However, server $S_j$ verifies the freshness of timestamp $T_i$ and checks whether $D_i = h(N_i || RPW_i|||A_i|||T_i)$ holds. So server $S_j$ treats adversary $E$ as a malicious hacker and terminates this session. Furthermore the presented scheme introduces a biometrics-based fuzzy extractor to meet the applicability of biometric information. Consequently, our protocol resists the Denial-of-Service attack.

Resistance to password guessing attack. With the assistance of power consumption, adversary $E$ applies the side-channel attacks, such as SPA or DPA, to extract the sensitive datas $A_j, C_j, E_j, V_j$ and $P_i$ from user $U_i$’s smart card $SC_i$.

But he is unable to verify whether user $U_i$’s password $PW_j$ is correct in the on-line or off-line environment without biometric information $BIO_i$, pre shared key $PSK$, master secret key $s$ and random nonce $N_1$. Specifically unpredictable binary string $R_i$ which possesses a high entropy protects user $U_i$’s password $PW_i$ in the proposed scheme. In conclusion, our protocol is secure against the password guessing attack.

Resistance to smart card attack. Without the password $PW_i$ or biometric information $BIO_i$, adversary $E$ launches the smart card attack in order to collect some sensitive datas stored
in the smart card $SC_i$ and achieve server $S_j$‘s authentication. In the presented scheme, adversary $E$ is able to acquire user $U_i$‘s sensitive datas $A_i$, $C_i$, $E_i$, $V_i$ and $P_i$ which are saved in the smart card $SC_i$ by SPA or DPA. Also a session key $SK_{ij}$ between user $U_i$ and server $S_j$ is calculated as follows.

$$K_i = h(SID_i||(ID_i \oplus C_i)),$$

$$N_1 = K_i \oplus M_1,$$

$$N_2 = h(A_i||RPW_i||N_1) \oplus M_1,$$

$$SK_{ij} = h(ID_i||SID_i||N_1||N_2).$$

It is feasible for adversary $E$ to obtain $M_1$ and $M_4$ through a public channel. However, it is pretty difficult for him to retrieve the random nonces $N_1$ or $N_2$. As a result, our protocol withstands the smart card attack.

**Resistance to user impersonation attack.** Under the user impersonation attack, adversary $E$ who is an outside hacker tries to impersonate user $U_i$ without the password $PW_i$ or biometric information $BIO_i$. In the proposed scheme, adversary $E$ is unable to acquire $h(PSK)$ even if he eavesdrops user $U_i$‘s previous login request message $[M_1, M_2, M_3, B_i, D_i, T_i]$ and extracts user $U_i$‘s sensitive datas from smart card $SC_i$ by SPA or DPA. Thus, adversary $E$ cannot retrieve the random numbers $N_1$, $N_2$ or session key $SK_{ij}$. Therefore, our protocol is secure against the user impersonation attack.

**Resistance to privileged insider attack.** Adversary $E$ who is a malicious insider and has a privilege to access an authorized system attempts to impersonate user $U_i$. In order to achieve this goal, adversary $E$ collects user $U_i$‘s registration request message $[ID_i, RPW_i]$ and steals his smart card $SC_i$. However, it is impossible to obtain $h(PSK)$ and $B_i$ for adversary $E$. Even if sensitive datas $A_i$, $C_i$, $E_i$, $V_i$ and $P_i$ are extracted from user $U_i$‘s smart card $SC_i$, adversary $E$ is unable to deliver a correct login request message $[M_1, M_2, M_3, B_i, D_i, T_i]$. Furthermore, he cannot retrieve the password $PW_i$ or biometric information $BIO_i$. In conclusion, our protocol resists the privileged insider attack.

**Resistance to server spoofing attack.** Under the assumption that adversary $E$ who is a malicious insider but isn’t another server $S_k$ is able to steal user $U_i$‘s smart card $SC_i$ and eavesdrop his registration request message $[ID_i, RPW_i]$. Adversary $E$ tries to masquerade as server $S_j$ to spoof user $U_i$ by collecting the sensitive datas $A_i$, $C_i$, $E_i$, $V_i$ and $P_i$. But it is hard to retrieve $h(PSK)$ so that adversary $E$ is unable to be authenticated by user $U_i$ successfully. He cannot acquire the random number $N_1$ and valid authentication request message $[M_4, M_5]$. Thus adversary $E$‘s attempt fails. Consequently, our protocol prevents the server spoofing attack.

**Resistance to modification attack.** Though adversary $E$ attempts to modify some intercepted messages for further authentication, the proposed protocol is able to check whether the received messages are valid with the assistance of collision-resistant hash function. And adversary $E$ does not have a capability to retrieve $N_1$, $N_2$ or $h(PSK)$ from any intercepted message. Thus he cannot generate a legitimate authentication message. As a result, our protocol is secure against the modification attack.

**Resistance to stolen-verifier attack.** In the proposed protocol, both server $S_j$ and registration center $RC$ possess no information about user $U_i$‘s password or biometrics. Concretely, there is no password-verifier or biometrics-verifier in the database of server $S_j$ and registration center $RC$. Thus, adversary $E$ cannot launch the stolen-verifier attack even if he has an authority to access the database. Consequently, our protocol withstands the stolen-verifier attack.
Possession of anonymity. During the login phase of the proposed scheme, user \( U_i \) calculates his dynamic identity \( M_2 = ID_j \oplus K_i \) in which \( K_i \) cannot be retrieved by adversary \( E \) from any request or reply message. Thus, adversary \( E \) has no ability to acquire user \( U_i \)'s identity \( ID_j \). However, upon receiving user \( U_i \)'s login request message, authorized server \( S_j \) calculates \( u_i = B_j \oplus h(PSK) \) and further computes \( K_i = h(SID_j \| h(PSK) \| u_i) \) so that user \( U_i \) achieves server \( S_j \)'s authentication anonymously. In other words, user \( U_i \)'s real identity \( ID_j \) is not disclosed by any unauthorized participant. Therefore our protocol provides the anonymity.

Possession of perfect forward secrecy. Perfect forward secrecy protects the session keys even if long-term key is retrieved. Specifically, session key \( SK_{ij} \) in the proposed scheme is generated as follows.

\[
K_i = h(SID_j \| h(PSK) \| u_i)), \\
N_1 = K_i \oplus M_1, \\
ID_i = K_i \oplus M_2, \\
N_2 = h(A_i \| RPW_i \| N_1) \oplus M_1, \\
SK_{ij} = h(ID_i \| SID_j) \| N_i \| N_2).
\]

Though the long-term key \( h(PSK) \) is calculated by adversary \( E \), it is impossible to compute some sensitive datas, such as \( RPW_i, K_i \) and \( PSK \). Thus adversary \( E \) is unable to obtain the random numbers \( N_1 \) or \( N_2 \). Also it is hard for adversary \( E \) to retrieve the session key \( SK_{ij} \) between user \( U_i \) and server \( S_j \). Therefore, our protocol provides the perfect forward secrecy.

Formal security analysis

During this subsection, we provide a formal security analysis and demonstrate that the proposed scheme is secure. In order to achieve this purpose, we define the oracle \( \text{Reveal} \) as below. It unconditionally retrieves the original input \( x \) from the collision-resistant hash function \( y = h(x) \). More details relating to this formal security analysis are shown in the following theorem.

**Theorem.** Suppose that the collision-resistant hash function \( h(\cdot) \) operates closely like the oracle \( \text{Reveal} \), our protocol is provably secure to protect the sensitive datas which include registration center \( RC \)'s master secret key \( s \), pre shared key \( PSK \) between registration center \( RC \) and server \( S_j \), user \( U_i \)'s identity \( ID_j \) and password \( PW_i \).

**Proof.** With the assistance of the oracle \( \text{Reveal} \), we make an assumption that adversary \( E \) has a capacity to retrieve registration center \( RC \)'s master secret key \( s \), pre shared key \( PSK \) between registration center \( RC \) and server \( S_j \), user \( U_i \)'s identity \( ID_j \) and password \( PW_i \). Adversary \( E \) executes the following experimental algorithm \( EXP_{E, \text{EAKAS}}^{\text{HASH}} \), in which \( \text{EAKAS} \) means the presented scheme. More details about the Algorithm \( EXP_{E, \text{EAKAS}}^{\text{HASH}} \) are explained in the Table 3

Furthermore, we define a success probability about \( EXP_{E, \text{EAKAS}}^{\text{HASH}} \) as

\[
\text{Success} = |P(EXP_{E, \text{EAKAS}}^{\text{HASH}} = 1) - 1|.
\]

Thus advantage function of algorithm \( EXP_{E, \text{EAKAS}}^{\text{HASH}} \) is \( \text{Adv}(et, q_{\text{Reveal}}) = \max_{E}[\text{Success}] \), namely, maximum for adversary \( E \) relies on the execution time \( et \) and query counts \( q_{\text{Reveal}} \) which are made to this oracle \( \text{Reveal} \). If \( \text{Adv}(et, q_{\text{Reveal}}) \leq \varepsilon \), our protocol is secure against adversary \( E \) for any sufficiently small \( \varepsilon > 0 \). It enables adversary \( E \) to win this game if it is possible to retrieve the original input \( x \) from the collision-resistant hash function \( y = h(x) \). However, it is a computationally infeasible problem for retrieving the original input \( x \). Therefore, for any sufficiently small \( \varepsilon > 0 \), \( \max_{E}[\text{Success}] = \text{Adv}(et, q_{\text{Reveal}}) \leq \varepsilon \). As a result, our
Table 3. Algorithm EXPRESS

01. Eavesdrop user U’s login request message \( M_1, M_2, M_3, B, D, T \) in the login phase,
in which \( B_1 = E_1 \oplus h(K_1), D_1 = h(N_1) [RPW^T] [A_1] [T] \), \( M_1 = N_1 \oplus \bar{K}_1, M_2 = ID_1 \oplus \bar{K}_1 \) and \( M_3 = RPW_1 \oplus K_1 \\
02. Apply this oracle \textit{Reveal} to extract some values \( N_1', RPW^{T'}, A_1' \) and \( T' \) from \textit{Reveal}(D_1) \rightarrow (N_1' [RPW^{T'}] [A_1'] [T']) \\
03. Eavesdrop server S’s authentication request message \( M_4, M_5 \) during the authentication phase,
in which \( M_4 = N_2 \oplus h(A_2) [RPW^T] [N_1] \) and \( M_5 = h(SID) [N_1] [ID] \).
04. Apply this oracle \textit{Reveal} to extract some values \( SID^u, N_3^u, N_3^o \) and \( ID^o \) from \textit{Reveal}(M_o) \rightarrow (SID^u) [N_3^u] [ID^o] \\
05. if \( (N_1^o = N_3^o) \) then
06. Apply this oracle \textit{Reveal} to extract some values \( K'_u \) and \( PW^u \) from \textit{Reveal}(RPW^o) \rightarrow (K'_u) [PW^u] \\
07. Further apply this oracle \textit{Reveal} to extract some values \( ID^u \) and \( s' \) from \textit{Reveal}(A'_u) \rightarrow (ID^u) [s'] \\
08. Calculate \( K'_u = M_1 \oplus N_3^o \\
09. Further calculate \( K''_u = M_1 \oplus N_3^o \)
10. if \( (K'_u = K''_u) \) then
11. Apply this oracle \textit{Reveal} to extract some values \( SID^u \) and \( h(PSK)[u'_j] \) from \textit{Reveal}(K'_u) \rightarrow (SID^u) [h(PSK)] [u'_j] \\
12. Further apply this oracle \textit{Reveal} to extract some values \( PSK' \) and \( u'_f \) from \textit{Reveal}(h(PSK)[u'_j]) \rightarrow (PSK') [u'_f] \\
13. Calculate \( N'_j = h(A'_j) [RPW^o] [N_3]\) \oplus M_4 \\
14. if \( (N'_j = N_3^o) \) then
15. Accept \( s', PSK', ID^u \) and \( PW^u \) as registration center RC’s master secret key s,
pre shared key PSK between registration center RC and server S, user U’s identity ID, and password PW, respectively.
16. \textbf{return} 1 (Success)
17. \texttt{else}
18. \textbf{return} 0 (Failure)
19. \texttt{end if}
20. \texttt{else}
21. \textbf{return} 0 (Failure)
22. \texttt{end if}
23. \texttt{else}
24. \textbf{return} 0 (Failure)
25. \texttt{end if}

https://doi.org/10.1371/journal.pone.0194093.0003

protocol is provably secure to protect registration center RC’s master secret key s, pre shared key PSK between registration center RC and server S, user U’s identity ID, and password PW.

Security analysis with BAN logic

As an important verification tool, Burrows-Abadi-Needham (BAN) logic has a set of rules [66]. In the security analysis, BAN logic is used for defining and analyzing the information exchange schemes, especially authentication and key agreement protocols. Particularly, BAN logic is able to verify whether exchanged information is trustworthy [67]. During this subsection, we apply BAN logic to prove that session key SK_j between server S and user U is correctly generated during the authentication phase of our protocol. For convenience, symbols and corresponding notions about BAN logic are respectively shown in Table 4.

The BAN logical postulates. 1. The message-meaning rule, namely, \( A = A \equiv [B \Rightarrow [X] \bar{K}] \). Particularly, if principal A believes that principal A and principal B share session key K, and principal A sees that statement X is encrypted by session key K, then principal A believes that principal B once said the statement X.
Table 4. Symbols and corresponding notions in the BAN logic.

| Symbol | Notion |
|--------|--------|
| A! ≡ X | Principal A believes the truth of statement X. |
| A ⊲ X | Principal A and principal B share session key K. |
| A ⇒ X | Principal A has a jurisdiction over the truth of statement X. |
| #X | Statement X is fresh. |
| A ≈ X | Principal A sees the statement X. |
| (X, Y)K | Statement X and statement Y are XORed by session key K. |
| (X, Y)K | Statement X and statement Y are hashed by session key K. |
| <X> | Statement X is XORed by session key K. |

https://doi.org/10.1371/journal.pone.0194093.t004

2. The nonce-verification rule, namely, A! = #X, A! = #Y. Specifically, if principal A believes that statement X is fresh and principal B once said the statement X, then principal A believes that principal B believes the truth of statement X.

3. The belief rule, namely, A! = #X. In particular, if principal A believes the truth of statement X and statement Y, then principal A believes the truth of (X, Y).

4. The freshness-conjunction rule, namely, A! = #X. Concretely, if principal A believes that statement X is fresh, then principal A believes that (X, Y) is fresh.

5. The jurisdiction rule, namely, A! = #X, A! = #Y. Especially, if principal A believes that principal B has a jurisdiction over the truth of statement X and principal B believes the truth of statement X, then principal A believes the truth of statement X.

The idealized scheme. Uj : <N1, IDn, RPWj> ⊲ Kj, (N1, A*, Tj)RPWj, and (Uj ⊲ SKj, Sj, Nj)Sj.

The establishment of security goals. g1. Uj | ≡ Sj | ≡ Uj ⊲ SKj, Sj

\[ g2. \] Uj | ≡ Uj ⊲ SKj, Sj

\[ g3. \] Sj | ≡ Uj | ≡ Uj ⊲ SKj, Sj

\[ g4. \] Sj | ≡ Uj ⊲ SKj, Sj

The initiative premises. p1. Uj | ≡ #N1

\[ p2. \] Uj | ≡ Sj ⇒ #N2

\[ p3. \] Sj | ≡ #N1

\[ p4. \] Sj | ≡ #N2

\[ p5. \] Sj | ≡ Uj ⊲ SKj, Sj

\[ p6. \] Uj | ≡ Uj ⊲ SDj, Sj

\[ p7. \] Uj | ≡ IDj

\[ p8. \] Sj | ≡ Uj ⇒ RPWj

\[ p9. \] Sj | ≡ Uj ⇒ IDj

\[ p10. \] Sj | ≡ Uj ⊲ SKj, Sj

\[ p11. \] Sj | ≡ Uj ⇒ Uj ⊲ SKj, Sj

\[ p12. \] Uj | ≡ Sj ⇒ Uj ⊲ SKj, Sj

The security analysis. a1. Because of p5 and Sj ≜ <N1, IDn, RPWj> ⊲ Kj, we execute the message-meaning rule to obtain Sj | ≡ Uj | ≈ (N1, IDn, RPWj).
a2. Since $p3$ and $a1$, we adopt both freshness-conjuncatenation rule and nonce-verification rule to acquire $S_i \equiv U_i \equiv (N_i, ID, RPW)$.

a3. Because of $p10$ and $S_i \equiv (U_i \xleftarrow{SK_i} S_j, N_j)^{N_i}$, we use the message-meaning rule to derive $S_i \equiv U_i \sim (U_i \xleftarrow{SK_i} S_j, N_j)$.

a4. Since $p4$ and $a3$, we apply both freshness-conjuncatenation rule and nonce-verification rule to get $S_i \equiv U_i \equiv (U_i \xleftarrow{SK_i} S_j, N_j)$.

g3. Because of $a4$, we execute the belief rule to obtain $S_i \equiv U_i \equiv U_i \xleftarrow{SK_i} S_j$.

g4. Since $p11$ and $g3$, we adopt the jurisdiction rule to acquire $S_i \equiv U_i \xleftarrow{SK_i} S_j$.

a5. Because of $p6$ and $U_i \xleftarrow{} (ID, N_i, N_j)_{SID}$, we use the message-meaning rule to derive $U_i \equiv S_i \sim (ID, N_i, N_j)$.

a6. Since $p2$ and $a5$, we apply both freshness-conjuncatenation rule and nonce-verification rule to get $U_i \equiv S_i \equiv (ID, N_i, N_j)$.

a7. Because of $a6$, we execute the belief rule to obtain $U_i \equiv S_i \equiv N_j$.

a8. Since $p2$ and $a7$, we adopt the jurisdiction rule to acquire $U_i \equiv N_j$.

a9. Because of $p8$, $p9$ and $a2$, we execute both belief rule and jurisdiction rule to obtain $S_i \equiv ID_i$.

g1. Since $p1$, $p3$, $p4$, $p6$, $p7$, $a8$, $a9$ and $SK_{ij} = h(ID_i||SID_i||N_i||N_j)$, we adopt both freshness-conjuncatenation rule and nonce-verification rule to acquire $U_i \equiv S_i \equiv U_i \xleftarrow{SK_i} S_j$.

g2. Because of $g1$ and $p12$, we use the jurisdiction rule to derive $U_i \equiv U_i \xleftarrow{SK_i} S_j$.

Above all, results mentioned above demonstrate that our protocol enables to generate the shared session key $SK_{ij}$ correctly between server $S_j$ and user $U_i$.

Functionality analysis

It is necessary to meet the functionality requirements which include mutual authentication, session key agreement, user revocation/re-registration and biometric information protection.

In this section, we demonstrate that our protocol provides all functionality mentioned above. More details relating to functionality analysis are shown as below.

**Mutual authentication.** In the presented scheme, both user $U_i$ and server $S_j$ authenticate each other by taking advantage of some sensitive datas, for example $N_i$, $N_j$, $K_p$, $T_i$ and $SK_{ij}$. In particular, server $S_j$ checks whether $h(N_i||RPW||A_i||T_i) = D_i$ and $h(SK_{ij}||N_i||N_j) = M_5$ are valid. Similarly, user $U_i$ verifies whether $h(SID_i||N_i||N_j||ID_j)$ is consistent with $M_5$. As a result, our protocol achieves the mutual authentication.

**Session key agreement.** During the authentication phase, session key $SK_{ij} = h(ID_i||SID_i||N_j||N_j)$ is established to protect the subsequent communications. Especially, both $N_i$ and $N_j$ change in every authentication phase so that session key $SK_{ij}$ is different during each session. Furthermore it is hard to retrieve their session key $SK_{ij}$ for adversary $E$. In conclusion, our protocol possesses the session key agreement.

**User revocation/re-registration.** It is necessary for user $U_i$ to revoke or re-register his privilege. In the presented scheme, registration center $RC$ helps user $U_i$ achieve the user revocation/re-registration by modifying the entry $(ID, N_i)$ when obtaining user $U_i$’s revocation or re-registration request message via a secure channel. Above all, our protocol achieves the user revocation/re-registration.

**Biometric information protection.** In some conventional schemes, user $U_i$’s biometric information $BIO_i$ is directly stored in his smart card $SC_i$ without appropriate protection. Thus adversary $E$ is able to extract user $U_i$’s biometrics $BIO_i$ from a lost or stolen smart card $SC_i$ through side channel attacks. In order to solve this problem, we apply a high error-tolerant
mechanism to save user $U_i$’s biometric information $BIO_i$. Besides, collision-resistant hash function protects the unpredictable binary string $R_i$. So it is impossible for adversary $E$ to extract user $U_i$’s biometric information $BIO_i$. In conclusion, our protocol possesses the biometric information protection.

**Efficiency analysis**

In this subsection, we estimate the storage requirement, communication overhead and computational cost of the presented scheme. More details about efficiency analysis are shown as below.

**Storage requirement.** For the storage requirement, we apply these messages which are stored in user $U_i$’s smart card $SC_i$ as storage overhead. Particularly, byte length of nonce both $N_1$ and $N_2$ is 20, byte length of user $U_i$’s identity $ID_i$ is 20, byte length of timestamp $T_i$ is 2 and byte length of collision-resistant hash function’s output is 20 if we apply the SHA-1. Thus, we are able to calculate the byte length of stored datas in the proposed scheme. As a result, all saved messages $\{A_i, C_i, E_i, V_i, P_i\}$ require $20 + 20 + 20 + 20 + 20 = 100$ bytes in respect of storage need.

**Communication overhead.** In order to estimate the communication overhead, we consider user $U_i$’s login request message $\{M_1, M_2, M_3, B_i, D_i, T_i\}$ which is submitted to server $S_j$ in the stage of login. According to assumption described above, length of this message is $20 + 20 + 20 + 20 + 20 + 2 = 102$ bytes. Similarly, communication overhead that includes server $S_j$’s authentication request message $\{M_4, M_5\}$ and user $U_i$’s authentication reply $M_6$ is $20 + 20 + 20 = 60$ bytes during the authentication phase. Therefore, total communication overhead of our protocol is $102 + 60 = 162$ bytes.

**Computational cost.** Considering the computational complexity, we apply the frequency of collision-resistant hash function as computational cost. Besides, it is practicable to ignore the computational complexity of XOR operation which requires very little time. In the environment where CPU is 2.20 GHz and RAM is 2048 MB, it takes 0.0023 ms to execute the collision-resistant hash function on average [55, 68]. In the presented scheme, we execute the collision-resistant hash function four times and thirteen times in the login phase and authentication phase, respectively. Above all, our protocol requires $0.0115 + 0.0299 = 0.0414$ ms for computational cost.

**Comparisons with related schemes**

During this section, we compare the proposed protocol with other related schemes in terms of security, functionality and efficiency. In particular, our protocol is compared with some multi-server authentication schemes, such as Mishra et al.’s scheme [50], Lin et al.’s scheme [53], Wang et al.’s scheme [61], Chaudhry et al.’s scheme [64], Chaudhry et al.’s scheme [41] and Khan et al.’s scheme [65]. Results ensure that the presented protocol is efficient in these aspects mentioned above.

In particular, Table 5 lists the security comparison between various authentication schemes and ours. For convenience, we define some following notations in the Table 5, where R1 represents the resistance to replay attack, R2 represents the resistance to Denial-of-Service attack, R3 represents the resistance to password guessing attack, R4 represents the resistance to smart card attack, R5 represents the resistance to user impersonation attack, R6 represents the resistance to privileged insider attack, R7 represents the resistance to server spoofing attack, R8 represents the resistance to modification attack, R9 represents the resistance to stolen-verifier attack, R10 represents the possession of anonymity and R11 represents the possession of perfect forward secrecy. Concretely, Mishra et al.’s scheme [50] cannot resist the replay attack,
Denial-of-Service attack, smart card attack, user impersonation attack, privileged insider attack and server spoofing attack. Also their scheme is unable to provide the anonymity and perfect forward secrecy. According to the cryptanalysis in Ref. [69], Lin et al.’s scheme [53] is insecure against the user impersonation attack and server spoofing attack. And their scheme fails to possess the anonymity. Wang et al.’s scheme [61] cannot prevent the user impersonation attack, privileged insider attack and server spoofing attack. Also their scheme is unable to achieve the perfect forward secrecy. Due to the cryptanalysis in Ref. [70], Chaudhry et al.’s scheme [64] is insecure against the Denial-of-Service attack and cannot provide the perfect forward secrecy. Consequently, result demonstrates that our protocol achieves all security properties.

Besides, Table 6 shows the functionality comparison between some related schemes and ours. Also we further compare our protocol with Reddy et al.’s scheme [69] and Irshad et al.’s scheme [71] which are other improved schemes. In the Table 6, we apply some following notations, where F1 represents the mutual authentication, F2 represents the session key agreement, F3 represents the user revocation/re-registration and F4 represents the biometric information protection. Concretely, Mishra et al.’s scheme [50] cannot provide the user revocation/re-registration. Similarly, Lin et al.’s scheme [53] fails to achieve the user revocation/re-registration. As a result, our protocol provides more functionality properties.

Specifically, Table 7 and Fig 6 indicate the computational cost comparison between various related schemes and ours involved in both login phase and authentication phase. As a convenience, we define some following notations in the Table 7, where C1 represents the computational cost during the login phase, C2 represents the execution overhead during the login phase, C3 represents the computational cost during the authentication phase, C4 represents
the execution overhead during the authentication phase and C5 represents the total execution overhead. Besides, \(T_h\) represents the computation time for collision-resistant hash function, \(T_p\) represents the computation time for point multiplication based on elliptic curve, \(T_s\) represents the computation time for symmetric encryption/decryption and \(T_c\) represents the computation time for Chebyshev chaotic map. According to the execution overhead given in [55] and [68], in the environment where CPU is 2.20 GHz and RAM is 2048 MB, it spends about 2.2260 ms, 0.0046 ms and 0.0045 ms to execute the point multiplication based on elliptic curve, symmetric encryption/decryption and Chebyshev chaotic map, respectively. Compared with other schemes, result indicates that our protocol requires the lower computational cost.

Furthermore, Table 8 and Fig 7 show the comparisons regarding on communication overhead and storage requirement. Similarly, we adopt some following notations in the Table 8, where S1 represents the communication overhead during the login phase, S2 represents the communication overhead during the authentication phase, S3 represents the total communication overhead and S4 represents the storage requirement. With the same level of storage requirement, our protocol shows a satisfactory performance on the communication overhead.
Both Reddy et al. [69] and Irshad et al. [71] who proposed other improvements of Wang et al.’s scheme also have done well jobs. In this sense, we are in the same field with these groups. However, there are notable characters to distinguish our work. After the cryptanalysis of Wang et al.’s scheme, we have applied novel methods to remedy their weaknesses, which is not included in other improved schemes. For example, we have adopted new ways to resist the user impersonation attack, privileged insider attack and server spoofing attack, and provide the perfect forward secrecy, respectively. Furthermore, our work is focus on reducing the computational complexity and providing more functionalities in a distinct way. In particular, compared with other improved works, our scheme has obvious advantages in the computational complexity with the same level of communication overhead and storage requirement.

**Conclusion**

This paper cryptanalyzes Wang et al.’s scheme. In particular, we indicate that their protocol is still vulnerable to the user impersonation attack, privileged insider attack and server spoofing attack. Furthermore, their protocol fails to provide the perfect forward secrecy. As a remedy of these aforementioned problems, we propose a biometrics-based authentication and key...
agreement scheme for multi-server environments. Our protocol improves Wang et al.’s scheme. Discussions relating to security, functionality and efficiency are performed. Furthermore, results show that the proposed scheme satisfies these requirements mentioned above. Compared with other related schemes, our protocol achieves the stronger security and provides more functionality properties. Besides, the presented scheme requires the lower computational cost and shows a satisfactory performance on the communication overhead with the same level of storage requirement. Thus, the proposed protocol is suitable for expert systems and other multi-server architectures, such as, on-line medicine systems, on-line shopping systems and so on. Consequently, we conclude that our protocol is more appropriate in the multi-server environments.

Author Contributions

Conceptualization: Li Yang, Zhiming Zheng.
Data curation: Li Yang, Zhiming Zheng.
Formal analysis: Li Yang.
Funding acquisition: Li Yang.
Investigation: Li Yang, Zhiming Zheng.
Methodology: Li Yang.
Project administration: Li Yang.
Resources: Li Yang.
Software: Li Yang.
Supervision: Li Yang.
Validation: Li Yang, Zhiming Zheng.
Visualization: Li Yang, Zhiming Zheng.
Writing – original draft: Li Yang.
Writing – review & editing: Li Yang, Zhiming Zheng.

References

1. Khan MK, Zhang JS. Improving the security of a flexible biometrics remote user authentication scheme. Computer Standards & Interfaces. 2007; 29(1): 82–85. https://doi.org/10.1016/j.csi.2006.01.002
2. He DB, 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. https://doi.org/10.1109/TCE.2013.6689693
3. Diffie W, Van Oorschot PC, Wiener MJ. Authentication and authenticated key exchanges. Designs, Codes and Cryptography. 1992; 2(2): 107–125. https://doi.org/10.1007/BF00124891
4. Mishra D. Design and analysis of a provably secure multi-server authentication scheme. Wireless Personal Communications. 2016; 86(3): 1095–1119. https://doi.org/10.1007/s11277-015-2975-0
5. Mitchell JC. Finite-state analysis of security protocols. International Conference on Computer Aided Verification. Springer, Berlin, Heidelberg. 1998; 71–76.
6. Moon J, Choi Y, Jung J, Won D. An improvement of robust biometrics-based authentication and key agreement scheme for multi-server environments using smart cards. PLoS ONE. 2015; 10(12): e0145263. https://doi.org/10.1371/journal.pone.0145263 PMID: 26709702
7. Lamport L. Password authentication with insecure communication. Communications of the ACM. 1981; 24(11): 770–772. https://doi.org/10.1145/388790.388797
8. Farash MS, Attari MA. A secure and efficient identity-based authenticated key exchange protocol for mobile client-server networks. The Journal of Supercomputing. 2014; 69(1): 395–411. https://doi.org/10.1007/s11227-014-1170-5

9. 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. https://doi.org/10.1371/journal.pone.0102747 PMID: 25047235

10. Khan MK. Fingerprint biometric-based self-authentication and deniable authentication schemes for the electronic world. IETE Technical Review. 2009; 26(3): 191–195. https://doi.org/10.4103/0256-4602.50703

11. Kumari S, Khan MK. More secure smart card-based remote user password authentication scheme with user anonymity. Security and Communication Networks. 2014; 7(11): 2039–2053. https://doi.org/10.1002/sec.916

12. Farash MS, Chaudhry SA, Heydari M, Sadough S, Mohammad S, Kumari S, Khan MK. A lightweight anonymous authentication scheme for consumer roaming in ubiquitous networks with provable security. International Journal of Communication Systems. 2017; 30(4). https://doi.org/10.1002/dac.3019

13. Kumari S, Chaudhry SA, Wu F, Li X, Farash MS, Khan MK. An improved smart card based authentication scheme for session initiation protocol. Peer-to-Peer Networking and Applications. 2017; 10(1): 92–105. https://doi.org/10.1007/s12083-015-0409-0

14. Bellovin SM, Merritt M. Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise. Proceedings of the 1st ACM Conference on Computer and Communications Security. 1993; 244–250.

15. Chang TY, Hwang MS, Yang WP. A communication-efficient three-party password authenticated key exchange protocol. Information Sciences. 2011; 181(1): 217–226. https://doi.org/10.1016/j.ins.2010.08.032

16. Lee TF, Hwang T. Simple password-based three-party authenticated key exchange without server public keys. Information Sciences. 2010; 180(9): 1702–1714. https://doi.org/10.1016/j.ins.2010.01.005

17. Wang S, Wang J, Xu M. Weaknesses of a password-authenticated key exchange protocol between clients with different passwords. ACNS. 2004; 4: 414–425.

18. Ku WC, Chen CM, Lee HL. Weaknesses of Lee-Li-Hwang’s hash-based password authentication scheme. ACM SIGOPS Operating Systems Review. 2003; 37(4): 19–25. https://doi.org/10.1145/958965.958967

19. Ding Y, Horster P. Undetectable on-line password guessing attacks. ACM SIGOPS Operating Systems Review. ACM. 1995; 29(4): 77–86.

20. Chang CC, Wu TC. Remote password authentication with smart cards. IEEE Proceedings E (Computers and Digital Techniques). 1991; 138(3): 165–168. https://doi.org/10.1049/ip-e.1991.0022

21. Mishra D, Chaturvedi A, Mukhopadhyay S. Design of a lightweight two-factor authentication scheme with smart card revocation. Journal of Information Security and Applications. 2015; 23: 44–53. https://doi.org/10.1016/j.jisa.2015.06.001

22. Reddy AG, Yoon EJ, Das AK, Yoo KY. Lightweight authentication with key-agreement protocol for mobile network environment using smart cards. IET Information Security. 2016; 10(5): 272–282. https://doi.org/10.1049/iet-ifs.2015.0390

23. Kumari S, Li X, Wu F, Das AK, Arshad H, Khan MK. A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. Future Generation Computer Systems. 2016; 63: 56–75. https://doi.org/10.1016/j.future.2016.04.016

24. Karuppiah M, Kumari S, Das AK, Li X, Wu F, Basu S. A secure lightweight authentication scheme with user anonymity for roaming service in ubiquitous networks. Security and Communication Networks. 2016; 9(17): 4192–4209. https://doi.org/10.1002/sec.1598

25. Chaudhry SA, Farash MS, Naqvi H, Kumari S, Khan MK. An enhanced privacy preserving remote user authentication scheme with provable security. Security and Communication Networks. 2015; 8(18): 3782–3795. https://doi.org/10.1002/sec.1299

26. Wang CQ, Zhang X, Zheng ZM. An improved biometrics based authentication scheme using extended chaotic maps for multimedia medicine information systems. Multimedia Tools and Applications. 2017; 76(22): 24315–24341. https://doi.org/10.1007/s11284-016-4198-0

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

28. 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. https://doi.org/10.1002/dac.2468
29. 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. https://doi.org/10.1109/TC.2002.1004593

30. Wang D, Wang P. Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. Ad Hoc Networks. 2014; 20: 1–15. https://doi.org/10.1016/j.adhoc.2014.03.003

31. 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. https://doi.org/10.1016/j.jnca.2009.08.001

32. 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. https://doi.org/10.1016/j.jnca.2010.09.003

33. Odelu V, Das AK, Kumari S, Huang X, Wazid M. Provably secure authenticated key agreement scheme for distributed mobile cloud computing services. Future Generation Computer Systems. 2017; 68: 74–88. https://doi.org/10.1016/j.future.2016.09.009

34. Wazid M, Das AK, Kumari S, Li X, Wu F. Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS. Security and Communication Networks. 2016; 9(13): 1983–2001.

35. Amin R, Islam SH, Biswas GP, Khan MK, Leng L, Kumar N. Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks. Computer Networks. 2016; 101: 42–62. https://doi.org/10.1016/j.comnet.2016.01.006

36. Fan CI, Lin YH. Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics. IEEE Transactions on Information Forensics and Security. 2009; 4(4): 933–945. https://doi.org/10.1109/TIFS.2009.2031942

37. Lee JK, Ryu SR, Yoo KY. Fingerprint-based remote user authentication scheme using smart cards. Electronics Letters. 2002; 38(12): 554–555. https://doi.org/10.1049/el:20020380

38. Khan MK, Zhang JS. An efficient and practical fingerprint-based remote user authentication scheme with smart cards. Information Security Practice and Experience. 2006; 260–268. https://doi.org/10.1007/11689522_24

39. Benhammadi F, Bey KB. Password hardened fuzzy vault for fingerprint authentication system. Image and Vision Computing. 2014; 32(8): 487–496. https://doi.org/10.1016/j.imavis.2014.04.014

40. 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. https://doi.org/10.1109/TIT.2012.2200290

41. Chaudhry SA, Naqvi H, Farash MS, Shon T, Sher M. An improved and robust biometrics-based three factor authentication scheme for multiserver environments. The Journal of Supercomputing. 2015; 1–17.

42. Li LH, Lin LC, Hwang MS. A remote password authentication scheme for multiserver architecture using neural networks. IEEE Transactions on Neural Networks. 2001; 12(6): 1498–1504. https://doi.org/10.1109/72.963786 PMID: 18249979

43. Chiang 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. https://doi.org/10.1016/j.eswa.2013.08.040
50. 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. https://doi.org/10.1016/j.eswa.2014.07.004

51. Amin R, Biswas GP. Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment. Wireless Personal Communications. 2015; 84(1): 439–462. https://doi.org/10.1007/s11277-015-2616-7

52. He DB, Wang D. Robust biometrics-based authentication scheme for multiserver environment. IEEE Systems Journal. 2015; 9(3): 816–823. https://doi.org/10.1109/JSYST.2014.2301517

53. Lin H, Wen FT, Du CX. An improved anonymous multi-server authenticated key agreement scheme using smart cards and biometrics. Wireless Personal Communications. 2015; 84(4): 2351–2362. https://doi.org/10.1007/s11277-015-2708-4

54. Lu YR, Li LX, Yang X, Yang YX. Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. PLoS ONE. 2015; 10(5): e0126323. https://doi.org/10.1371/journal.pone.0126323 PMID: 25978373

55. Odelu V, Das AK, Goswami A. A secure biometrics-based multi-server authentication protocol using smart cards. IEEE Transactions on Information Forensics and Security, 2015; 10(9): 1953–1966. https://doi.org/10.1109/TIFS.2015.2439964

56. Reddy AG, Das AK, Odelu V, Yoo KY. An enhanced biometric based authentication with key-agreement protocol for multi-server architecture based on elliptic curve cryptography. PLoS ONE. 2016; 11(5): e0154308. https://doi.org/10.1371/journal.pone.0154308 PMID: 27163786

57. 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. https://doi.org/10.3837/tiis.2015.02.019

58. 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. https://doi.org/10.1007/s11277-014-2002-x

59. Tsudik G, Summers RC. AudES-An Expert System for Security Auditing. In Proceedings of the second conference on innovative applications of artificial intelligence. 1990; 221–232.

60. Hariri S, Jabbour K. An expert system for network management. In Proceedings of tenth annual international phoenix conference on computers and communications. 1991; 580–586.

61. Wang CQ, Zhang X, Zheng ZM. Cryptanalysis and improvement of a biometric-based multi-server authentication and key agreement scheme. PLoS ONE. 2016; 11(2): e0149173. https://doi.org/10.1371/journal.pone.0149173 PMID: 26866606

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

63. Dolev D, Yao A. On the security of public key protocols. IEEE Transactions on Information Theory. 1983; 29(2): 198–208. https://doi.org/10.1109/TIT.1983.1056650

64. Chaudhry SA, Naqvi H, Khan MK. An enhanced lightweight anonymous biometric based authentication scheme for TMIS. Multimedia Tools and Applications. 2017; 1–22.

65. Khan I, Chaudhry SA, Sher M, Khan JI, Khan MK. An anonymous and provably secure biometric-based authentication scheme using chaotic maps for accessing medical drop box data. The Journal of Supercomputing. 2016; 1–19.

66. Burrow M, Abadi M, Needham RM. A logic of authentication. ACM Transactions on Computer System. 1990; 8(1): 18–36. https://doi.org/10.1145/77648.77649

67. Moon J, Choi Y, Jung J, Won D. An improvement of robust biometrics-based authentication and key agreement scheme for multi-server environments using smart cards. PLOS ONE. 2015; 10(12): e0145263. https://doi.org/10.1371/journal.pone.0145263 PMID: 26709702

68. Kilinc HH, Yank J. A survey of SIP authentication and key agreement schemes. IEEE Communications Surveys & Tutorials. 2014; 16(2): 1005–1023. https://doi.org/10.1109/SURV.2013.091513.00050

69. Reddy AG, Yoon EJ, Das AK, Odelu V, Yoo KY. Design of Mutually Authenticated Key Agreement Protocol Resistant to Impersonation Attacks for Multi-Server Environment. IEEE Access. 2017; 5: 3622–3639. https://doi.org/10.1109/ACCESS.2017.2666258

70. Qi MP, Chen JH. New robust biometrics-based multi-server authentication scheme with key agreement using elliptic curve cryptography. Multimedia Tools and Applications. 2018; 1–17.

71. Irshad A, Chaudhry SA, Kumari S, Usman M, Mahmood K, Faisal MS. An improved lightweight multi-server authentication scheme. International Journal of Communication Systems. 2017; 30(17). https://doi.org/10.1002/dac.3351