Article

Robust Multiple Servers Architecture Based Authentication Scheme Preserving Anonymity

Huawei Wang ¹ , Dianli Guo ^{2,*}, Hua Zhang ¹ and Qiaoyan Wen ¹

¹ State Key Laboratory of Networking and Switching Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China

² The 6th Research Institute of China Electronics Corporation, Beijing 100083, China

* Correspondence: guodianli@163.com

Received: 23 May 2019; Accepted: 15 July 2019; Published: 17 July 2019

Abstract: Recently, many dynamic ID based remote user authentication schemes using smart card have been proposed to improve the security in multiple servers architecture authentication systems. In 2017, Kumari and Om proposed an anonymous multi-server authenticated key agreement scheme, which is believed to be secure against a range of network attacks. Nevertheless, in this paper we reanalyze the security of their scheme, and show that the scheme is vulnerable to impersonation attack and server spoofing attack launched by any adversary without knowing any secret information of the victim users. In addition, their protocol fails to achieve the claimed user privacy protection. For handling these aforementioned shortcomings, we introduce a new biometric-based authentication scheme for multi-server architecture preserving user anonymity. Besides, Burrows—Abadi—Needham (BAN)-logic validated proof and discussion on possible attacks demonstrate the completeness and security of our scheme, respectively. Further, the comparisons in terms of security analysis and performance evaluation of several related protocols show that our proposal can provide stronger security without sacrificing efficiency.

Keywords: authentication; anonymity; BAN-logic; biometrics; multiple server

1. Introduction

In the multiple servers architecture based authentication system, registration center, service providing servers and users are major participants. Registration center is the trusted party to administrate all the involved users and servers in the system. Servers provide network services and legitimate users can access these services. Compared with the conventional two-party authentication system, a multi-server architecture based authentication system offers registration procedure one time and allows users to access services from multiple servers. The latter obliterates the inappropriateness that users should perform stuffy reduplicative registration in each server.

In 2004, Das et al. [1] proposed a dynamic ID based remote user authentication scheme using smart card. Since then, many dynamic ID authentication schemes are published to enhance the security properties and reduce the communication and computation costs [2–12]. However, these schemes are designed for single-server architecture which are not suitable for a multi-server environment.

For fulfilling the particularity of the multi-server architecture, ample authentication schemes designed for the multi-server environment have been investigated by researchers. In 2009, Liao and Wang [13] proposed a remote user authentication scheme for multi-server architecture preserving user anonymity to eliminate the risk of ID-theft. Nevertheless, their scheme is proved to be susceptible to insider attack, masquerade attack and fail to provide a mutual authentication. Later on, Hsiang and Shih [14] introduced a remedied protocol to solve the above security flaws. Unfortunately, Sood et al. [15] reanalyzed their scheme and pointed out that it is vulnerable to replay

attack, impersonation attack and stolen smart card attack. Meanwhile, they presented a novel multiple servers based authentication scheme. After that, many multi-server authentication schemes have been proposed to strengthen the security and improve the efficiency [16–26].

In 2014, Chuang and Chen [16] proposed an anonymous multi-server authentication key agreement scheme using smart cards, password and biometrics. However, Kumari and Om [23] identified the vulnerabilities of their scheme, such as being insecure against DoS attack, user/server impersonation attack, stolen smart card attack, and failing to achieve perfect forward secrecy. For obliterating the aforementioned shortcomings of Chuang et al.'s scheme, they proposed an enhanced protocol for a multiple servers authentication system, which offered non-repudiation utilizing RSA digital signature. Moreover, the authors stated that their proposal possessed all required security properties and resisted all the network attacks. Unluckily, in this paper, we reexamine the security of Kumari et al.'s RSA cryptosystem based authentication scheme and indicate that their scheme falls short to withstand impersonation attack and server spoofing attack. Specifically, any adversary could break through their scheme easily, even without the knowledge of the victim's user information. Moreover, adversaries could create test scenarios to execute the brute force attack and reveal users' low entropy identities. For the purpose of surmounting the identified vulnerabilities, we further devise an improved biometric-based multi-server authentication scheme with a distinctive policy compared with the original. Note that, Burrows—Abadi—Needham (BAN)-logic, one of the important formal methods focusing on evaluating the beliefs of participants in authentication system, is put forward to certify the validity of our proposal. Finally, the security and performance analysis are discussed to observe that the proposed protocol is superior to other related schemes.

This paper is organized as follows. We introduce the basic concept of fuzzy extraction in Section 2. Then, in Sections 3 and 4 we briefly review Kumari et al.'s scheme and identify its security flaws, respectively. Next, we propose a new robust authentication scheme in Section 5 and analyze its security in Section 6. Subsequently, in Section 7 we compare the performance of our new protocol with the previous schemes. Finally, the paper is concluded in Section 8.

2. Preliminaries

In this section, we briefly introduce the basic concept of fuzzy extractor, for more details please refer to [27]. In 1999, Juels and Wattenberg fetched out the definition of fuzzy extractor which focused on verifying the legality of users by biometric template. Noticeably, it could deal with non-uniformity and error tolerance. Concretely, it could output a uniform key R with an auxiliary P and non-uniform noisy biometric input B^* by employing reproducible extraction, which was an error tolerant approach. The auxiliary string P to recover authentication key R is a public parameter and does certainly not compromise secrecy of R . Probabilistic generation algorithm Gen and deterministic reproduction algorithm Rep are efficient procedures of fuzzy extractor with parameters (m, l, t, ϵ) , which are detailed as follows.

- Gen : Inputs biometric template B , outputs an authenticated value $R \in \{0, 1\}^l$ and an auxiliary value $P \in \{0, 1\}^*$.
- Rep : For all B, B' , if $dis(B, B') \leq t$ and $\langle R, P \rangle \leftarrow Gen(B)$, then $Rep(B', P) = R$.

We list the notations used throughout this paper in Table 1.

Table 1. Notations.

U_i	User
S_j	Service providing server
RC	Registration center
ID_i	Identification of user
PW_i	Password of user
SID_j	Public identification of server
B_i	Biometrics information of user
X_c	Secret key of registration center
p, q	Two distinct large primes
$n, \phi(n)$	$n = p \times q, \phi(n) = (p-1) \times (q-1)$
SK_{ij}	Session key shared between user and server
$H(\cdot)$	Hash function
$\oplus$	Exclusive-OR operation
$\parallel$	String concatenation operation

3. Review of Kumari and Om's Scheme

In this section, we briefly describe Kumari and Om's [23] multi-server architecture based authentication scheme. It consists of initialization, registration, login, authentication and password changing phases. In Figure 1, we describe in detail the login and authentication phases in the form of infographics.

Figure 1. Login phase and authentication phase.

3.1. Initialization Phase

Registration center RC chooses a secret value X_c and two distinct large prime numbers p, q . Subsequently, it calculates $n = p \times q$ and $\phi(n) = (p - 1) \times (q - 1)$. X_c is the master secret key and only kept by RC . p, q could be destroyed to avoid leaking.

3.2. Registration Phase

3.2.1. Server Registration

Application server S_j transmits its identification SID_j to RC and applies for the jurisdiction to offer network services. RC selects a random number $e_j \in (1, \phi(n))$ with $\gcd(e_j, \phi(n)) = 1$. Then it computes and seeks out d_j such that $e_j \times d_j \equiv 1 \pmod{\phi(n)}$. Finally, RC sends the credentials $\{M1_j, e_j, d_j, n\}$ to S_j via a secured communication channel, where $M1_j = H(SID_j \| X_c)$, d_j are kept secret and e_j, n are announced as public values.

3.2.2. User Registration

Step 1: U_i firstly imprints his/her biometrics and uses fuzzy extractor to obtain authenticated value R_i and auxiliary value P_i such that $Gen(B_i) = (R_i, P_i)$. Then, he/she selects identity ID_i and password PW_i to calculate $PB_i = H(PW_i \oplus R_i)$ and sends $\{ID_i, PB_i, P_i\}$ to registration center for registration.

Step 2: Upon receiving registration request from U_i , RC computes $HPW_i = H(ID_i \| PB_i)$, $A_i = H(ID_i \| X_c) \oplus PB_i$, $B_i = H(ID_i \oplus PB_i) \oplus A_i$, $C_{ij} = B_i \oplus M1_j^{e_j}$ and $D_{ij} = M1_j \oplus H(ID_i \| X_c)$. Then, RC transmits the smart card contained $\{B_i, C_{ij}, D_{ij}, HPW_i, P_i, H(\cdot)\}$ to U_i securely.

3.3. Login Phase

U_i inserts his/her smart card into the terminal and inputs identity ID_i , password and the biometric template B_i imprinted at the sensor. The smart card will execute the following procedure.

Step 1: Performs reproduction algorithm $R_i^* = Rep(B_i, P_i)$ and computes $PB_i^* = H(PW_i \oplus R_i^*)$, $HPW_i^* = H(ID_i \| PB_i^*)$.

Step 2: Verifies the equivalence of HPW_i^* and the stored value HPW_i . If they are equal, proceeds to next steps; otherwise, terminates this session immediately.

Step 3: Generates a random number r_u and acquires the current timestamp $T1$ to calculate $N1_{ij} = (C_{ij} \oplus B_i)^{r_u}$, $N2_{ij} = (D_{ij} \oplus B_i \oplus H(ID_i \oplus PB_i) \oplus PB_i)^{r_u}$, $CID_{ij} = ID_i \oplus N2_{ij}$, $N3_{ij} = H(ID_i \| (C_{ij} \oplus B_i) \| N2_{ij} \| T1)$.

Step 4: Submits the login request message $\{CID_{ij}, N1_{ij}, N3_{ij}, T1\}$ to S_j .

3.4. Authentication Phase

On receiving the login request $\{CID_{ij}, N1_{ij}, N3_{ij}, T1\}$ from U_i at $T2$, S_j verifies the validity of $T1$ by checking $T2 - T1$ whether less or equal than the permissible time interval ΔT for a transmission delay. If so, continues to perform the following steps; else, S_j aborts the login session.

Step 1: S_j computes $N2_{ij} = (N1_{ij})^{d_j}$, $ID_i = CID_{ij} \oplus N2_{ij}$, $N3_{ij} = H(ID_i \| M1_j^{e_j} \| N2_{ij} \| T1)$ with the known credential d_j .

Step 2: Then S_j verifies the computed $N3_{ij}$ with the received one. If the equation does not hold, terminates the session; on the contrary, continues to execute the further steps.

Step 3: S_j acquires the current timestamp $T3$ and generates a random number r_s to compute $M2_{ij} = (M1_j)^{r_s \cdot d_j}$, $M3_{ij} = (N1_{ij})^{r_s}$, $SK_{ij} = H(ID_i \| SID_j \| M3_{ij} \| N2_{ij})$, $M4_{ij} = H(SK_{ij} \| T3)$, $CSID_{ij} = SID_j \oplus M3_{ij}$.

Step 4: Subsequently, S_j responses to U_i the replied mutual authentication message $\{CSID_{ij}, M2_{ij}, M4_{ij}, T3\}$.

Step 5: Upon receiving the response message from S_j at T_4 , U_i checks whether $T_4 - T_3 \leq \Delta T$. If it does not hold, U_i gives up this login procedure; otherwise, the smart card computes $M3_{ij} = (M2_{ij})^{e_j^2 \cdot r_u}$, $SID_j = CSID_{ij} \oplus M3_{ij}$, $SK_{ij} = H(ID_i \| SID_j \| M3_{ij} \| N2_{ij})$, $M4_{ij} = H(SK_{ij} \| T3)$.

Step 6: Afterwards, the smart card checks the equivalence of the computed $M4_{ij}$ and the received one. If they are not equal, the authentication fails; else, U_i confirms S_j is authentic and the mutual authentication is completed. Finally, U_i and S_j share a current session key SK_{ij} .

3.5. Password Changing Phase

In the procedure of password changing phase, U_i could update her/his password offline. Firstly, he/she should insert smart card into the device and input ID_i , PW_i , the biometric template B_i . Then, the smart card verifies the legitimation of U_i to launch the following steps.

Step 1: Computes $R_i = Rep(B_i, P_i)$, $PB_i = H(PW_i \oplus R_i)$ and compares the stored value HPW_i equals to the computed $H(ID_i \| PB_i)$. If they are not equal, the smart card terminates this session; otherwise, the smart card continues to compute $A_i = B_i \oplus H(ID_i \oplus PB_i)$, $H(ID_i \| X_c) = A_i \oplus PB_i$, $M1_j^{e_j} = C_{ij} \oplus B_i$.

Step 2: Subsequently, U_i is allowed to input a new password PW_i^{new} . The smart card calculates $PB_i^{new} = H(PW_i^{new} \oplus R_i)$, $HPW_i^{new} = H(ID_i \| PB_i^{new})$, $A_i^{new} = H(ID_i \| X_c) \oplus PB_i^{new}$, $B_i^{new} = H(ID_i \oplus PB_i^{new}) \oplus A_i^{new}$, $C_{ij}^{new} = B_i^{new} \oplus M1_j^{e_j}$.

Step 3: Finally, the smart card replaces $\{B_i, C_{ij}, HPW_i\}$ with the new parameters to finish the password change phase.

4. Cryptanalysis of Kumari and Om's Scheme

In this phase, we show that Kumari and Om's protocol is vulnerable to impersonation attack, server spoofing attack and fails to protect user anonymity. Their scheme is thoroughly broken down by any malicious user in the multi-server authentication system, even when one knows nothing about the victim user. The detailed demonstration is described as follows.

4.1. Impersonation Attacks

Consider a legitimate but malicious user U_A in the multiple servers authentication system, he/she can obtain $M1_j^{e_j} = C_{Aj} \oplus B_A$ and $M1_j = D_{Aj} \oplus B_A \oplus H(ID_A \oplus PB_A) \oplus PB_A$ with secret parameters stored in his/her own smart card. Then, the adversary can further impersonate any legal user (even a non-existent user) to unauthorized access S_j .

In the login phase, U_A randomly selects a string ID_k with the format of identity and computes $N1_{kj} = (M1_j^{e_j})^{r_A} = M1_j^{e_j \cdot r_A}$, $N2_{kj} = (M1_j)^{r_A}$, $CID_{kj} = ID_k \oplus N2_{kj}$, $N3_{kj} = H(ID_k \| M1_j^{e_j} \| N2_{kj} \| T1_k)$, where r_A and $T1_k$ are generated random number and acquired current timestamp respectively. Subsequently, he/she transmits the forged login request $\{CID_{kj}, N1_{kj}, N3_{kj}, T1_k\}$ to S_j .

After receiving the forged login request, S_j checks the validity of $T1_k$ and calculates $N2_{kj} = (N1_{kj})^{d_j}$, $ID_k = CID_{kj} \oplus N2_{kj}$, $N3_{kj} = H(ID_k \| M1_j^{e_j} \| N2_{kj} \| T1_k)$ with the known credential d_j . Obviously, the computed $N3_{kj}$ is consistent to the forged one in the login request, that is, the verification of U_A is successful and S_j accepts the login request of the adversary. Subsequently, S_j computes $M2_{kj} = (M1_j)^{r_s \cdot d_j}$, $M3_{kj} = (N1_{kj})^{r_s}$, $SK_{kj} = H(ID_k \| SID_j \| M3_{kj} \| N2_{kj})$, $M4_{kj} = H(SK_{kj} \| T3)$, $CSID_{kj} = SID_j \oplus M3_{kj}$ with the random number r_s and timestamp $T3$, and replies $\{CSID_{kj}, M2_{kj}, M4_{kj}, T3\}$ to the adversary.

After that, the adversary computes $M3_{kj} = (M2_{kj})^{e_j^2 \cdot r_A}$, $SID_j = CSID_{kj} \oplus M3_{kj}$, $SK_{kj} = H(ID_k \| SID_j \| M3_{kj} \| N2_{kj})$. Finally, U_A obtains the session key SK_{kj} and uses it to communicate with S_j . Hence, the adversary successfully accesses the service providing server unauthorized.

4.2. Failure of Preserving Anonymity

As described above, a legitimate but malicious user U_A could obtain $M1_j^{e_j}$ with his/her own secret values. Suppose that U_A intercepts U_i 's login request $\{CID_{ij}, N1_{ij}, N3_{ij}, T1\}$ in a prior transaction, he/she could easily get U_i 's identity by the brute force attack. In the following we present the concrete procedures.

Step 1. Firstly, let ID_i^* be an identity candidate in the identity space. Subsequently, the adversary computes $N2_{ij}^* = CID_{ij} \oplus ID_i^*$.

Step 2. Secondly, the adversary checks $N3_{ij} = H(ID_i^* \| M1_j^{e_j} \| N2_{ij}^* \| T1)$ to verify the correctness of chosen candidate ID_i^* .

Step 3. The adversary performs Steps 1 and 2 repeated with another candidate in the identity space until the correct ID_i such that $N3_{ij} = H(ID_i \| M1_j^{e_j} \| (CID_{ij} \oplus ID_i) \| T1)$ is found.

Actually, the above attack could be executed effectively since the amount of identity space is limited. The primary causes of this problem are the inherently restricted human cognition and the limitation of identity format.

4.3. Server Spoofing Attack

In Kumari and Om's protocol, a legitimate but malicious user U_A also can masquerade as an authorized server. Based on the description in the above analysis, U_A can obtain U_i 's identity ID_i by employing feasible brute force attack and record the identity SID_j of S_j in a prior session. Furthermore, he/she also needs to intercept the mutual authentication message $\{CSID_{ij}, M2_{ij}, M4_{ij}, T3\}$ replied to U_i from S_j in a previous session, and records a pair values $(M2_{ij}, M3_{ij}) = (M2_{ij}, CSID_{ij} \oplus SID_j)$. Noticeably, the adversary merely does preparatory work one time, rather than repetitively recording these values before performing each server spoofing attack. The concrete description of server spoofing attack is shown as follows.

Step 1: Suppose that U_i requests to access S_j with $\{CID_{ij}, N1_{ij}, N3_{ij}, T1\}$. The adversary selects a random number r_A and computes

$$\begin{aligned} M2_{ij}^* &= (M2_{ij})^{r_A} = (M1_j)^{r_s \cdot r_A \cdot d_j}, \\ M3_{ij}^* &= (M3_{ij})^{r_A} = (N1_{ij})^{r_s \cdot r_A}, \\ SK_{ij}^* &= H(ID_i \| SID_j \| M3_{ij}^* \| (CID_{ij} \oplus ID_i)), \\ M4_{ij}^* &= H(SK_{ij}^* \| T3_A), \\ CSID_{ij}^* &= SID_j \oplus M3_{ij}^* \end{aligned}$$

with the values previously recorded. Afterwards, sends the forged response $\{CSID_{ij}^*, M2_{ij}^*, M4_{ij}^*, T3_A\}$ to U_i .

Step 2: After receiving the forged reply, U_i computes

$$\begin{aligned} M3_{ij}^* &= (M2_{ij}^*)^{e_j^2 \cdot r_u} = (M1_j)^{e_j \cdot r_u \cdot r_s \cdot r_A} = (N1_{ij})^{r_s \cdot r_A}, \\ SID_j &= CSID_{ij}^* \oplus M3_{ij}^*, \\ SK_{ij}^* &= H(ID_i \| SID_j \| M3_{ij}^* \| N2_{ij}). \end{aligned}$$

Obviously, the computed $H(SK_{ij}^* \| T3_A)$ is equal to the received $M4_{ij}^*$. Hence, U_i authenticates the adversary successfully and communicates with him/her. The major contributor of the network flaw is the allelomorphism of array $(M2_{ij}, M3_{ij})$ —any attacker could reconstitute these two values by performing an exponentiation with exponent r_A respectively.

5. Our Scheme

Herein, we propose a novel multiple servers architecture based authentication scheme with biometrics, which contains five phases, namely initialization phase, registration phase, login phase, authentication phase and password changing phase. Furthermore, we depict the login and authentication phases in Figure 2.

Figure 2. Login phase and authentication phase.

5.1. Initialization Phase

Registration center RC initializes the authentication system with secret value X_c and two distinct large primes p, q . Then, it keeps $\{X_c\}$ secret and publishes the public parameters $\{n, \phi(n)\}$, where $n = p \times q, \phi(n) = (p - 1) \times (q - 1)$. Finally, RC obliterates the two values p, q .

5.2. Registration Phase

This segment contains two sub-phases: server registration and user registration. Service providing servers and users apply for authorization of registration center through the following procedures, respectively.

5.2.1. Server Registration

Similar to the original protocol, service providing server S_j sends its identity SID_j to RC for registration. After receiving the registration request, it seeks out two large numbers $e_j \in (1, \phi(n))$ and d_j such that $\gcd(e_j, \phi(n)) = 1$ and $e_j \times d_j \equiv 1 \pmod{\phi(n)}$, computes $s_j = H(SID_j \parallel X_c)$. Afterwards, RC transmits the calculated credentials $\{s_j, e_j, d_j, n\}$ to S_j . S_j publishes $\{e_j, n\}$ and keeps $\{s_j, d_j\}$ as secret keys.

5.2.2. User Registration

Step 1: U_i imprints the biometrics B_i and invokes the fuzzy extractor to generate $(R_i, P_i) \leftarrow \text{Gen}(B_i)$. Subsequently, he/she calculates $IB_i = H(ID_i \| R_i)$ and $PB_i = H(PW_i \| R_i)$ with the selected identity and password. After that, U_i registers in RC with $\{IB_i, PB_i\}$.

Step 2: Then, RC computes $K_{ij} = H(IB_i \| s_j)$ with each service providing server secret key $\{s_1, s_2, \dots, s_k\}$. RC continues to calculate $A_{ij} = (K_{ij})^{e_j} \oplus H(IB_i \oplus PB_i)$, $C_{ij} = K_{ij} \oplus PB_i$ and $D_i = H(IB_i \| PB_i)$. Afterwards, RC personalizes the smart card with the $\{(A_{i1}, A_{i2}, \dots, A_{ik}), (C_{i1}, C_{i2}, \dots, C_{ik}), D_i\}$ and sends it to U_i via a secure channel.

5.3. Login Phase

Step 1: U_i inserts his/her smart card into the card reader and inputs ID_i, PW_i , the imprinted biometric template B'_i . Then the smart card recovers the value R_i through $R_i \leftarrow \text{Rep}(B'_i, P_i)$, computes $IB_i = H(ID_i \| R_i)$, $PB_i = H(PW_i \| R_i)$, and verifies whether the computed $H(IB_i \| PB_i)$ equals to the stored D_i or not. If they are consistent, continues to execute Step 2; otherwise, the login phase is aborted directly.

Step 2: The smart card generates a random number r_i and calculates $M1_{ij} = (A_{ij} \oplus H(IB_i \oplus PB_i))^{r_i} = (K_{ij})^{e_j \cdot r_i}$, $M2_{ij} = (C_{ij} \oplus PB_i)^{r_i} = (K_{ij})^{r_i}$, $CID_{ij} = IB_i \oplus M2_{ij}$, $K_{ij} = C_{ij} \oplus PB_i$, $M3_{ij} = H(IB_i \| K_{ij} \| M2_{ij} \| T_i)$, where T_i is the acquired current timestamp.

Step 3: U_i accesses S_j with the login request $\{CID_{ij}, M1_{ij}, M3_{ij}, T_i\}$.

5.4. Authentication Phase

Step 1: Upon receiving U_i 's login request at $T1_i$, S_j checks the validity of T_i . If $T1_i - T_i \geq T$, S_j rejects U_i 's login request; otherwise, it computes $M2_{ij} = (M1_{ij})^{d_j} = (K_{ij})^{r_i}$. Further, S_j uses $M2_{ij}$ to recover IB_i with $CID_{ij} \oplus M2_{ij}$. Subsequently, it verifies the uniformity of the computed value $H(IB_i \| H(IB_i \| s_j) \| M2_{ij} \| T_i)$ and the received $M3_{ij}$. If they are equal, the legitimacy of U_i is ensured; on the contrary, S_j discards the session immediately.

Step 2: After that, S_j acquires the current timestamp T_j and generates a random integer number r_j to compute $SK_{ij} = (M1_{ij})^{r_j \cdot d_j} = (K_{ij})^{r_j \cdot r_i}$, $V1_{ij} = (K_{ij})^{r_j \cdot d_j}$, $V2_{ij} = H(SID_j \| SK_{ij} \| K_{ij} \| T_j)$. Subsequently, it sends the response authentication message $\{V1_{ij}, V2_{ij}, T_j\}$ to U_i .

Step 3: Upon receiving the replied message at $T1_j$, the smart card verifies the validity of T_j . If $T1_j - T_j$ is less than or equal to the permissible time interval ΔT for a transmission delay, the authentication fails. Otherwise, the smart card calculates $SK_{ij} = (V1_{ij})^{e_j \cdot r_i}$ and $V2_{ij}^* = H(SID_j \| SK_{ij} \| K_{ij} \| T_j)$. If $V2_{ij}^* = V2_{ij}$, U_i confirms that S_j is authentic and mutual authentication is completed successfully; on the contrary, the session will be terminated.

After finishing the above mutual authentication procedures, S_j and U_i agree on the session key SK_{ij} for the future secure communication.

5.5. Password Changing Phase

These procedures are invoked whenever U_i changes the overdue password with a new one.

Step 1: Similar to Step 1 of login phase, the smart card verifies the legitimacy of the card holder. If it confirms the validity of U_i , the smart card proceeds to Step 2; otherwise, it rejects the request of changing password.

Step 2: The smart card permits U_i to enter a new password PW_i^{new} to replace the original. Specifically, U_i should enter the new one twice to prevent him/her from typing errors. Suppose that the entered passwords are unequal—the smart card requests U_i to enter a new one two more times.

Step 3: After that, the smart card computes $PB_i^{new} = H(PW_i^{new} \| R_i)$, $A_{ij}^{new} = A_{ij} \oplus H(IB_i \oplus PB_i^{new})$, $C_{ij}^{new} = C_{ij} \oplus PB_i^{new}$, $D_i^{new} = H(IB_i \| PB_i^{new})$. Then it replaces the original values with $\{(A_{i1}^{new}, A_{i2}^{new}, \dots, A_{ik}^{new}), (C_{i1}^{new}, C_{i2}^{new}, \dots, C_{ik}^{new}), D_i^{new}\}$.

6. Security Analysis and Discussion

6.1. Authentication Proof Based on BAN-Logic

Herein, we present the demonstration for the completeness of the proposed scheme through BAN-logic [28]. BAN-logic is one of the widely employed formal proofs for analyzing the trustworthiness of involved participants in authentication protocol.

In the following, we define some notations for the further BAN-logic analysis.

- $\mathcal{P} \equiv X$: The principal $\mathcal{P}$ believes a statement X or $\mathcal{P}$ would be entitled to believe X .
- $\sharp(X)$: The formula X is fresh.
- $\mathcal{P} \Rightarrow X$: The principal $\mathcal{P}$ has jurisdiction over the statement X .
- $\mathcal{P} \triangleleft X$: The principal $\mathcal{P}$ sees the statement X .
- $\mathcal{P} \sim X$: The principal $\mathcal{P}$ once said the statement X .
- (X, Y) : The formula X or Y is one part of the formula (X, Y) .
- $\langle X \rangle_Y$: The formula X is combined with the formula Y .
- $\mathcal{P} \xleftrightarrow{K} \mathcal{Q}$: The principals $\mathcal{P}$ and $\mathcal{Q}$ use the shared key K to communicate. Here, K will never be discovered by any principal except for $\mathcal{P}$ and $\mathcal{Q}$.
- $\mathcal{P} \stackrel{K}{\rightleftharpoons} \mathcal{Q}$: K is shared secret known to $\mathcal{P}$, $\mathcal{Q}$, and possibly to one trusted by them.
- SK_{ij} : The session key used in the current session.

We present several logical postulates of BAN-logic as follows.

- The message-meaning rule: $\frac{\mathcal{P} \equiv \mathcal{Q} \stackrel{K}{\rightleftharpoons} \mathcal{P}, \mathcal{P} \triangleleft \langle X \rangle_K}{\mathcal{P} \equiv \mathcal{Q} \sim X}$.
- The freshness-conjunction rule: $\frac{\mathcal{P} \equiv \sharp(X)}{\mathcal{P} \equiv \sharp(X, Y)}$.
- The nonce-verification rule: $\frac{\mathcal{P} \equiv \sharp(X), \mathcal{P} \equiv \mathcal{Q} \sim X}{\mathcal{P} \equiv \mathcal{Q} \equiv X}$.
- The jurisdiction rule: $\frac{\mathcal{P} \equiv \mathcal{Q} \Rightarrow X, \mathcal{P} \equiv \mathcal{Q} \equiv X}{\mathcal{P} \equiv X}, \frac{\mathcal{P} \equiv (X, Y)}{\mathcal{P} \equiv X}, \frac{\mathcal{P} \triangleleft (X, Y)}{\mathcal{P} \triangleleft X}, \frac{\mathcal{P} \equiv \mathcal{Q} \sim (X, Y)}{\mathcal{P} \equiv \mathcal{Q} \sim X}$.

In the following, we present the verification goals based on the analytic procedures of BAN-logic.

- **Goal 1:** $U_i \equiv (U_i \xleftrightarrow{SK_{ij}} S_j)$
- **Goal 2:** $S_j \equiv (U_i \xleftrightarrow{SK_{ij}} S_j)$

Next, we present the idealized form of the proposed scheme which was arranged from generic type.

- Message 1: $U_i \rightarrow S_j: (CID_{ij}, M1_{ij}, \langle IB_i, M2_{ij}, T_i \rangle_{K_{ij}}, T_i)$
- Message 2: $S_j \rightarrow U_i: (V1_{ij}, \langle SID_j, SK_{ij}, T_j \rangle_{K_{ij}}, T_j)$

In the following, we present some assumptions about the initial state of our proposed scheme to further analyze it.

- A.1: $U_i \equiv (U_i \xleftrightarrow{K_{ij}} S_j)$
- A.2: $S_j \equiv (U_i \xleftrightarrow{K_{ij}} U_i)$
- A.3: $U_i \equiv \sharp(T_j)$
- A.4: $S_j \equiv \sharp(T_i)$
- A.5: $S_j \equiv U_i \Rightarrow (IB_i, M2_{ij}, T_i)$
- A.6: $U_i \equiv S_j \Rightarrow (SID_j, SK_{ij}, T_j)$
- A.7: $S_j \equiv r_j$

Next, we analyze the idealized form of the proposed protocol based on the aforementioned assumptions and logical postulates—the primary proof steps of BAN-logic are described in the following:

According to Message 1, we could prove:

$$S_j \triangleleft (CID_{ij}, M1_{ij}, \langle IB_i, M2_{ij}, T_i \rangle_{K_{ij}}, T_i).$$

According to the jurisdiction rule, we could prove:

$$S_j \triangleleft \langle IB_i, M2_{ij}, T_i \rangle_{K_{ij}}.$$

According to assumption A.2 and the message-meaning rule, we could prove:

$$S_j \models U_i \mid \sim (IB_i, M2_{ij}, T_i).$$

According to the assumption A.4 and the freshness-conjunction rule, we could prove:

$$S_j \models \#(IB_i, M2_{ij}, T_i).$$

According to $S_j \models U_i \mid \sim (IB_i, M2_{ij}, T_i)$ and the nonce-verification rule, we could prove:

$$S_j \models U_i \models (IB_i, M2_{ij}, T_i).$$

According to the assumption A.5 and the jurisdiction rule, we could prove:

$$S_j \models (IB_i, M2_{ij}, T_i).$$

According to the jurisdiction rule, we could prove:

$$S_j \models M2_{ij}.$$

According to $SK_{ij} = (M2_{ij})^{r_j}$ and the assumption A.7, we could prove:

$$S_j \models (U_i \xleftrightarrow{SK_{ij}} S_j) \text{ (Goal 2)}.$$

According to Message 2, we could prove:

$$U_i \triangleleft (V1_{ij}, \langle SID_j, SK_{ij}, T_j \rangle_{K_{ij}}, T_j).$$

According to the jurisdiction rule, we could prove:

$$U_i \triangleleft \langle SID_j, SK_{ij}, T_j \rangle_{K_{ij}}.$$

According to assumption A.1 and the message-meaning rule, we could prove:

$$U_i \models S_j \mid \sim (SID_j, SK_{ij}, T_j).$$

According to the assumption A.3 and the freshness-conjunction rule, we could prove:

$$U_i \models \#(SID_j, SK_{ij}, T_j).$$

According to $U_i \models S_j \mid \sim (SID_j, SK_{ij}, T_j)$ and the nonce-verification rule, we could prove:

$$U_i \models S_j \models (SID_j, SK_{ij}, T_j).$$

According to the assumption A.6 and the jurisdiction rule, we could prove:

$$U_i \models (SID_j, SK_{ij}, T_j).$$

According to the jurisdiction rule, we could prove:

$$U_i \models SK_{ij},$$

$$U_i \models SID_j.$$

According to $U_i \models SK_{ij}$ and $U_i \models SID_j$, we could prove:

$$U_i \models (U_i \xleftrightarrow{SK_{ij}} S_j) \text{ (Goal 1)}.$$

6.2. Discussion on Possible Attacks

In this section, we present the security analysis in regard to a series of venomous network attacks and security properties to evaluate the proposed scheme.

6.2.1. Preserve User Privacy

In the proposed scheme, S_j can retrieve the identity information $IB_i = H(ID_i \| R_i)$ from the value $CID_{ij} = IB_i \oplus M2_{ij}$ in the login request, which integrates with exponent r_i of K_{ij} . S_j keeps the secret key d_j and can recover $M2_{ij}$ with another value $M1_{ij}$ in login request by computing $M2_{ij} = M1_{ij}^{d_j}$. In this way, the adversary either compromises S_j 's master secret key d_j or solves the big integer factorization problem. Whereas, it is infeasible for him/her to obtain IB_i in the above introduced method. Additionally, in our scheme, the dynamic identity CID_{ij} is invoked by a hash value IB_i of U_i 's identity and biometric template, rather than a low entropy identity ID_i . Hence, the adversary could

not reveal user's ID_i by the attack introduced for breaching Kumari and Om' protocol. Accordingly, our proposal is secure to against ID-theft attack and achieves user privacy protection.

6.2.2. Off-Line Password Guessing Attack

The adversary could perform brute force attack to compromise the low entropy password with the eavesdropped session messages and revealed parameters stored in the smart card of victim users [29,30]. Because of this vulnerability, we introduce another security factor biometrics in our proposal. Concretely, IB_i , PB_i are both attached with the secret value R_i retrieved by legitimate biometric template B_i . Assume that the attacker has revealed the credential $D_i = H(IB_i || PB_i)$ stored in the smart card, he/she has to guess identity ID_i , password PW_i and secret value R_i simultaneously. Notably, the length of secret value R_i meets the requirements of information security. Thus, off-line password guessing attack is fruitless for our proposed scheme.

6.2.3. Impersonation Attack

Impersonation attack means that an adversary forges a login request to masquerade legitimate users for unauthorized access to network services. It is indispensable for the adversary to generate an authenticated login request message $\{CID_{ij}, M1_{ij}, M3_{ij}, T_i\}$, where $M1_{ij} = (A_{ij} \oplus H(IB_i \oplus PB_i))^{r_i} = (K_{ij})^{e_j r_i}$, $M2_{ij} = (C_{ij} \oplus PB_i)^{r_i} = (K_{ij})^{r_i}$, $CID_{ij} = IB_i \oplus M2_{ij}$, $K_{ij} = C_{ij} \oplus PB_i$, $M3_{ij} = H(IB_i || K_{ij} || M2_{ij} || T_i)$. From computational procedure of these parameters, we can obviously see that K_{ij} and IB_i are the key values to form them. In our proposal, $K_{ij} = H(IB_i || s_j)$ is a unique secret value contributed by secrets of server and user, instead of a static value $M1_j$ of S_j for each user in Kumari and Om' protocol. The measure guarantees that users cannot abuse a unitary element to access servers illegally. On the other hand, the adversary also has no ability to calculate the verified login request without knowing K_{ij} . Consequently, the impersonation attack is trivial in our scheme.

6.2.4. Server Spoofing Attack

Server spoofing attack indicates that someone (it could even be a legal but malicious server) pretends to be another server to deceive users. In order to perform the attack, the adversary should reply to U_i a rightful authentication message $\{V1_{ij}, V2_{ij}, T_j\}$ likewise with the victim server, where $V1_{ij} = (K_{ij})^{r_j d_j}$, $V2_{ij} = H(SID_j || SK_{ij} || K_{ij} || T_j)$, $SK_{ij} = (K_{ij})^{r_j r_i}$. The value K_{ij} is also the core to generate the response parameters. As described in Section 6.2.3, K_{ij} is only accessible to U_i and S_j —others could obtain it unless it compromises the secret key X_c of RC. Therefore, server spoofing attack is meaningless in our scheme.

6.2.5. Replay Attacks

The replay attack signifies that someone spitefully resubmits repeated or delayed messages to deceive honest participants for nefarious purposes. Timestamping is one of the most widely employed techniques to prevent replay attack. In our proposal, both login request and replied authentication messages are involved in current timestamp. Both participants can verify its validity by detecting message transmitting delay. As a consequence, the replay attack is resisted effectively.

6.2.6. Forward Secrecy

Forward secrecy of information exchange protocol safeguards the past sessions to be revealed in which the long term key of RC is compromised in the future, even if the adversary actively interfered. Our proposed scheme achieves forward secrecy because the session key $SK_{ij} = (K_{ij})^{r_j r_i}$ is surrounded by r_i and r_j . Even though the adversary calculates K_{ij} with the leaked key X_c , he/she also is unable to further compromise SK_{ij} computed with the contribution of one-time random numbers $\{r_i, r_j\}$.

7. Performance and Functionality Analysis

Herein, we present performance and functionality evaluation analysis of our proposed scheme and other recently related protocols, that is, Chuang et al.'s scheme [16], Kumari and Om's scheme [23] and Jangirala et al.'s scheme [26]. Tables 2 and 3 show the comparative study in terms of security features and computational cost of the proposed scheme along with the aforementioned schemes, separately.

Table 2. Comparisons of functionality.

	[16]	[23]	[26]	Ours
Prevention of impersonation attack	No	No	No	Yes
Prevention of off-line password guessing attack	Yes	No	Yes	Yes
Prevention of server spoofing attack	No	No	No	Yes
Preserving user privacy	Yes	No	No	Yes
Prevention of replay attack	No	Yes	Yes	Yes
Formal security proof	No	Yes	Yes	Yes
Mutual authentication	No	No	No	Yes
Smart card breach attack	Yes	No	No	Yes
Perfect forward secrecy	No	Yes	Yes	Yes

Table 3. Performance comparisons.

	[16]	[23]	[26]	Ours
Login phase	$4T_h$	$4T_h + 2T_e$	$8T_h$	$4T_h + 2T_e$
Authentication phase	$13T_h$	$5T_h + 5T_e$	$17T_h$	$4T_h + 4T_e$
Computation cost	$17T_h$	$9T_h + 7T_e$	$25T_h$	$8T_h + 6T_e$

According to the comparisons of Table 2, we can see that our scheme satisfies all the requirements and criterion for multiple servers based authentication system. In contrast, the other three schemes suffer from more or less susceptibilities, even the superiorities claimed by the authors. In the modified scheme, we eliminate these flaws and enhance the security by targeted renovation.

In Table 3, the notations T_h and T_e denote the consuming time for a one-way hash function and a modular exponential operation, respectively. The evaluation shown in Table 3 focuses on the login phase, authentication phase and neglects the other three phases which do not frequently need to be performed. Chuang et al.'s and Jangirala et al.'s schemes use symmetric encryption and only perform hash function. Our proposed scheme and Kumari & Om's scheme are employed by RSA cryptosystem and require to execute modular exponentiation. Thus, the latter two schemes need to expend more computational cost. From Table 3, the total computation cost of Chuang et al.'s scheme, Kumari & Om's scheme, Jangirala et al.'s scheme and our scheme are $17T_h$, $9T_h + 7T_e$, $25T_h$ and $8T_h + 6T_e$. Noticeably, our scheme can thwart many security threats identified on these schemes. Additionally, our scheme is proved formally with the BAN-logic.

8. Conclusions

This paper firstly identified that Kumari and Om's anonymous multi-server authenticated key agreement scheme was plagued by impersonation attack, server spoofing attack and privacy disclosure. Even worse, any attacker could decipher it by launching a malicious attack without the knowledge of the victim's secret information. Secondly, we introduce a modified multiple servers architecture based authentication scheme with biometric to rectify these security flaws. Subsequently, to evaluate the devised scheme, we present the formal proof validated by BAN-logic and logical analysis for a range of network attacks. The performance and functionality comparisons in terms of computational cost and security features show that the designed protocol is superior for multiple servers authentication system.

Author Contributions: Methodology, Q.W.; Project administration, D.G.; Writing—original draft, H.W.; Writing—review & editing, H.Z.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Das, M.; Saxena, A.; Gulati, V. A dynamic ID-based remote user authentication scheme. *IEEE Trans. Consum. Electron.* **2004**, *50*, 665–667. [\[CrossRef\]](#)
2. Li, X.; Wen, Q.; Zhang, H.; Jin, Z. An improved authentication with key agreement scheme on elliptic curve cryptosystem for global mobility networks. *Int. J. Netw. Manag.* **2013**, *23*, 311–324. [\[CrossRef\]](#)
3. Guo, D.; Wen, Q.; Li, W.; Zhang, H.; Jin, Z. A Novel Authentication Scheme Using Self-certified Public Keys for Telecare Medical Information Systems. *J. Med. Syst.* **2015**, *39*, 62. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Guo, D.; Wen, Q.; Li, W.; Zhang, H.; Jin, Z. Analysis and Improvement of ‘Chaotic Map Based Mobile Dynamic ID Authenticated Key Agreement Scheme’. *Wirel. Pers. Commun.* **2015**, *83*, 35–48. [\[CrossRef\]](#)
5. Shahzad, M.; Singh, M.P. Continuous Authentication and Authorization for the Internet of Things. *IEEE Internet Comput.* **2017**, *21*, 86–90. [\[CrossRef\]](#)
6. Tan, H.; Gui, Z.; Chung, I. A Secure and Efficient Certificateless Authentication Scheme with Unsupervised Anomaly Detection in VANETs. *IEEE Access* **2018**, *6*, 74260–74276. [\[CrossRef\]](#)
7. Tan, H.; Chung, I. A Secure and Efficient Group Key Management Protocol with Cooperative Sensor Association in WBANs. *Sensors* **2018**, *18*, 3930. [\[CrossRef\]](#) [\[PubMed\]](#)
8. Shahzad, M.; Liu, A. Fast and Accurate Estimation of RFID Tags. *IEEE/ACM Trans. Netw.* **2015**, *23*, 241–254. [\[CrossRef\]](#)
9. Wang, D.; Li, W.; Wang, P. Measuring Two-Factor Authentication Schemes for Real-Time Data Access in Industrial Wireless Sensor Networks. *IEEE Trans. Ind. Inf.* **2018**, *14*, 4081–4092. [\[CrossRef\]](#)
10. Wang, D.; Wang, P. Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound. *IEEE Trans. Dependable Secure Comput.* **2018**, *15*, 708–722. [\[CrossRef\]](#)
11. Jiang, Q.; Qian, Y.; Ma, J.; Ma, X.; Cheng, Q.; Wei, F. User Centric Three-Factor Authentication Protocol For Cloud-Assisted Wearable Devices. *Int. J. Commun. Syst.* **2019**, *32*, e3900. [\[CrossRef\]](#)
12. Jiang, Q.; Ma, J.; Yang, C.; Ma, X.; Shen, J.; Chaudhry, S. Efficient end-to-end authentication protocol for wearable health monitoring systems. *Comput. Electr. Eng.* **2017**, *63*, 182–195. [\[CrossRef\]](#)
13. Liao, Y.; Wang, S. A secure dynamic ID based remote user authentication scheme for multi-server environment. *Comput. Stand. Interface* **2009**, *19*, 13–22. [\[CrossRef\]](#)
14. Hsiang, H.; Shih, W. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. *Comput. Stand. Interface* **2009**, *31*, 1118–1123. [\[CrossRef\]](#)
15. Sood, S.; Sarje, A.; Singh, K. A secure dynamic identity based authentication protocol for multi-server architecture. *J. Netw. Comput. Appl.* **2011**, *34*, 609–618. [\[CrossRef\]](#)
16. Chuang, M.C.; Chen, M.C. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. *Expert Syst. Appl.* **2014**, *41*, 1411–1418. [\[CrossRef\]](#)
17. Li, X.; Xiong, Y.; Ma, J.; Wang, W. An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. *J. Netw. Comput. Appl.* **2012**, *35*, 763–769. [\[CrossRef\]](#)
18. Li, X.; Ma, J.; Wang, W.; Xiong, Y.; Zhang, J. A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments. *Math. Comput. Model.* **2013**, *58*, 85–95. [\[CrossRef\]](#)
19. Guo, D.; Wen, F. Analysis and improvement of a robust smart card based-authentication scheme for multi-server architecture. *Wirel. Pers. Commun.* **2014**, *78*, 475–490. [\[CrossRef\]](#)
20. Wang, B.; Ma, M. A smart card based efficient and secured multi-server authentication scheme. *Wirel. Pers. Commun.* **2013**, *68*, 361–378. [\[CrossRef\]](#)
21. Pippal, R.; Jaidhar, C.; Tapaswi, S. Robust smart card authentication scheme for multi-server architecture. *Wirel. Pers. Commun.* **2013**, *72*, 729–745. [\[CrossRef\]](#)
22. Xue, K.; Hong, P.; Ma, C. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *J. Comput. Syst. Sci.* **2014**, *80*, 195–206. [\[CrossRef\]](#)

23. Kumari, S.; Om, H. Cryptanalysis and Improvement of an Anonymous Multi-server Authenticated Key Agreement Scheme. *Wirel. Pers. Commun.* **2017**, *96*, 2513–2537. [[CrossRef](#)]
24. Lu, Y.; Li, L.; Peng, H.; Yang, Y. Cryptanalysis and improvement of a chaotic maps-based anonymous authenticated key agreement protocol for multiserver architecture. *Secur. Commun. Netw.* **2016**, *9*, 1321–1330. [[CrossRef](#)]
25. Tsai, J.L.; Lo, N.W. A chaotic map-based anonymous multi-server authenticated key agreement protocol using smart card. *Int. J. Commun. Syst.* **2015**, *28*, 1955–1963. [[CrossRef](#)]
26. Jangirala, S.; Mukhopadhyay, S.; Das, A.K. A Multi-server Environment with Secure and Efficient Remote User Authentication Scheme Based on Dynamic ID Using Smart Cards. *Wirel. Pers. Commun.* **2017**, *95*, 2735–2767. [[CrossRef](#)]
27. Juels, A.; Wattenberg, M. A fuzzy commitment scheme. In Proceedings of the 6th ACM Conference on Computer and Communications Security, Kent Ridge Digital Labs, Singapore, 1–4 November 1999; ACM: New York, NY, USA, 1999; pp. 28–36.
28. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [[CrossRef](#)]
29. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the 19th Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999; pp. 388–397.
30. Messerges, T.; Dabbish, E.; Sloan, R. Examining smart-card security under the threat of power analysis attacks. *IEEE Trans. Comput.* **2002**, *5*, 541–552. [[CrossRef](#)]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).