An Extended Chaotic Map-Based Authentication and Key Agreement Scheme for Multi-Server Environment

Yicheng Yu 1, Oliver Taylor 2, Rui Li 3 and Baiho Sunagawa 4, *

1 Cyberspace Security Research Center, Pengcheng Laboratory, Shenzhen 518055, China; yuych@pcl.ac.cn
2 School of Engineering and Computer Science, University of Hull, Hull HU6 7RX, UK; oliver.taylor@hull.ac.uk
3 Department of Electrical and Computer Engineering, University of Texas at Dallas, Richardson, TX 75080, USA; ruili13@utdallas.edu
4 School of Computing and Mathematics, Keele University, Newcastle ST5 5BG, UK
* Correspondence: b.sun@keele.ac.uk

Abstract: With the increasing number of users and the emergence of different types of network services, a multi-server architecture has emerged in recent years. In order to ensure the secure communication of Internet participants in an open network environment, the authentication and key agreement protocol for multi-server architectures were proposed in the past. In 2018, Chatterjee et al. put forward a lightweight three-factor authentication and key agreement protocol for a multi-server environment, and they claimed that all known security features with satisfactory performance could be realized in their protocol. However, it is found that their scheme is vulnerable to user impersonation attacks and cannot achieve user un-traceability and three-factor security through our cryptanalysis. In order to solve these shortcomings, we propose a new lightweight and anonymous three-factor authentication scheme for the multi-server environment in this article. Furthermore, the proposed protocol is proved to be AKE secure theoretically, and we use BAN-logic to prove that our protocol realizes mutual authentication between communication participants. Finally, we show that our proposed scheme is practical and efficient through the comparison of security features and performance.

Keywords: authentication; key agreement; three-factor; cryptanalysis; multi-server environment

1. Introduction

In the past two decades, people’s lives have changed significantly because of the development of the Internet. People benefit from a variety of Internet services anytime and anywhere, such as telemedicine services, online shopping, online meetings, online games, and so on. Online life has become the mainstream mode of life, and the virtual network has changed the world [1,2]. With the continuous growth of the online network service business, security and privacy protection has become one of the most important challenges restricting its further development [3,4].

Authentication key agreement protocol is an effective security protocol to realize communication security in a client-server architecture. It can realize mutual authentication between users and servers, ensure that only legitimate users can access the server. At the same time, it can also effectively resist server spoofing attacks. When the user and the server complete mutual authentication, the two sides will negotiate to get their session key, which is used to ensure the security of their future communication. Moreover, the session key is obtained by negotiation between the two parties, and both parties have the same contribution to the generation of the session key, which enhances the security of the session key.

In the traditional single-server network environment, there is a service provider that provides network services for many users. When users access network services, they need to provide legal user identity and authentication factors (passwords, smart cards,
biometrics). However, with the strong demand for more types of network services, users need to prepare multiple sets of user identity and authentication factors to register multiple single server network systems in order to access different single server network systems. Obviously, this has caused great inconvenience. If users set the same authentication factor for different systems, when the user password of a system is leaked, it will also affect the security of other systems, which has great security risks. On the other hand, each network service system needs an authentication server to complete the user registration operation, which causes a serious waste of resources.

In order to solve the above drawbacks, the authentication key agreement protocol for a multi-server environment arises at the historic moment. Users can use the same set of identity and authentication factors to complete mutual authentication with different servers in the system so as to obtain the corresponding network services. Generally, the registration center RC needs to complete the initialization of the system. At the same time, it is responsible for the registration of users and service providers into the system and distributes the secret information related to the registrants when the registration is completed. When the registered users want to access the network services, they need to authenticate with the server and establish their session key after the authentication to ensure their future network communication security [5]. The network model of the multi-server environment is shown in Figure 1.

![Network model of multi-server environment.](image)

In 2001, Li et al. [6] proposed the first authentication protocol in a multi-server environment. However, Lin et al. [7] pointed out that the performance of the protocol is poor due to the use of the neural network. Meanwhile, to improve the performance of the protocol, Lin et al. designed an authentication protocol based on a discrete logarithm problem [7]. Unfortunately, their scheme was soon found unable to resist the attack of fake users [8]. At the same time, for the sake of improving the performance, many authentication protocols based on symmetric cryptography primitives [9–15] have been proposed.

Although these protocols use lightweight symmetric cryptography primitives and their performance has been improved, it is difficult for these protocols to achieve strong
security attributes, such as perfect forward secrecy. To ensure the security and practicability of the protocol, researchers designed an authentication protocol based on Elliptic Curve Cryptography in a multi-server environment. In 2013, Yoon and Yoo proposed a three-factor authentication protocol based on elliptic curve cryptography [16]. However, the protocol is not secure; malicious users can fake the identity of other users to obtain network services [17]. Subsequently, He and Wang put forward an improved protocol [18] based on Yoon’s protocol [16], but Odelu et al. [19] pointed out that the improved protocol could not achieve user anonymity. In 2015, Tsai proposed a new authentication protocol for multi-server environments [20] and claimed that their protocol could achieve strong security. However, reference [21] claimed that the protocol could not resist server spoofing attacks. Since 2017, Kumari et al. [22] and Wu et al. [23] have proposed relevant authentication protocols for multi-server environments. However, some security problems were found in the proposed scheme by Feng et al. [24] and Wang et al. [25], respectively. Kumari et al.’s scheme [22] has weak user un-traceability and is vulnerable to man-in-the-middle attacks. Wu et al.’s scheme [23] is vulnerable to smart card stolen attacks and temporary information leakage attacks. Based on the previous works, the improved schemes enhance the security and performance step by step. For example, Haq et al. [26] put forward a new, improved protocol based on the work of Ying-Nayak et al. [27] and Kumar-Ohm et al. [28] in 2021. In recent years, as an effective security mechanism to ensure network security, the authentication protocol in multi-server environments has been paid attention to by scholars, and the related protocols [29] have been proposed one after another. In the research process of authentication and key agreement protocol, these schemes not only need to improve the security (such as introducing biological information as the security factor) but also should have better performance to adapt to the more practical environment, such as wireless sensor network, body area network, and so on.

Through the review of the authentication schemes above, we find that researchers are easy to ignore the user un-traceability and N-factor security of their protocol, and many protocols are also vulnerable to user impersonation attacks. For instance, Chatterjee et al. proposed a three-factor authentication and key agreement protocol based on an extended chaotic map for the multi-server environment in 2018 [30] and claimed that the protocol could achieve all known security features with satisfactory performance. However, it is found that their scheme is vulnerable to user impersonation attacks and cannot achieve user un-traceability and three-factor security through our cryptanalysis.

Based on our analysis of the above protocols, we propose three basic design principles of authentication and key agreement protocol for multi-server environments in this study:

1) The authentication and key agreement protocol with high-level anonymity cannot be realized only by using symmetric cryptography (such as hash function and XOR operation). In other words, public key technology is a necessary condition to realize user anonymity.

2) In order to ensure the n-factor security of the authentication protocol, the local verification of the smart card cannot be the deterministic verification method, and the fuzzy authentication technology should be introduced to avoid the offline password guessing attacks initiated by the adversary.

3) In the login and authentication phase, the requester has a complete set of legal ID, password, smart card, and biological information, which is the necessary condition to generate legal login request information. Only in this way can we ensure the correctness of users’ identity and resist the user impersonation attacks.

Contributions

Our crucial contributions are as follows.

1) We review and analyze Chatterjee et al.’s three-factor authentication scheme for multi-server environments. Further, we show that their scheme is vulnerable to user impersonation attacks and cannot achieve user un-traceability and three-factor security.
(2) We present a new lightweight anonymous three-factor authentication scheme with
perfect forward secrecy for multi-server environments. Our scheme uses an extended
chaotic map and achieves strong security.

(3) The proposed protocol is proved to be AKE secure theoretically, and we use
BAN-logic to prove that our protocol realizes mutual authentication between communica-
tion participants

(4) Through the comparison of security features and performance, it can be found that
our proposed scheme is excellent and practical.

2. Preliminaries

2.1. Discrete Logarithm

Given a finite cyclic group $G$ and its generator $g \in G_1$, there is a unique integer $x$
such that $a = g^x$, $a \in G_1$. x is the discrete logarithm of $a$, which is recorded as $x = \log_g a$.

Discrete logarithm problem (DLP): Given a finite cyclic group $G_1$ whose generator is
$g \in G_1$ and an element $a \in G_1$, DLP is to find the integer $x$ such that $a = g^x$.

Computational Diffie–Hellman problem (CDHP): Given a finite cyclic group $G_1$ whose
generator is $g \in G_1$ and two elements $g^a, g^b \in G_1$, CDHP is to calculate the value of $g^{ab}$.

DLP and CDHP are known mathematical problems, which are not computationally
feasible; that is, they are not solvable in polynomial time. They are often used in the
construction and design of public-key cryptography.

2.2. Chebyshev Chaotic Map

Chebyshev chaotic map satisfies the following iterative relation:
$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \text{where } n \geq 2, n \in \mathbb{Z}, x \in [-1,1], T_0(x) = 1, T_1(x) = x.$$  
Chebyshev chaotic map has semi-group property, i.e., $T_i(T_j(x)) = T_{ij}(x) = T_j(T_i(x))$.

In 2008, Zhang et al. [31] extended the domain of Chebyshev chaotic map $x \in [-1,1]$ to
$x \in [-\infty, +\infty]$. The extended Chebyshev chaotic map still has the semi-group prop-
erty, namely $T_n(T_s(x)) = T_{ns}(x) = T_s(T_n(x))$, where $T_n(x) = \cos(n \cdot \arccos(x)) \mod p$,
$n \geq 2, n \in \mathbb{Z}, x \in [-\infty, +\infty]$, and $p$ is a large prime number.

Chaotic map discrete logarithm problem (CMDLP): Given a Chebyshev chaotic map
$T_n(x)$ and two random variables: $x$ and $y = T_s(x)$, CMDLP is to calculate the value of $r$.

Computational chaotic maps Diffie–Hellman problem (CMCDHP): Given a Chebyshev
chaotic map $T_n(x)$ and $(x, y = T_s(x), z = T_s(x))$, CMCDHP is to calculate the value of
$T_{sr}(x)$.

2.3. Adversarial Model

Due to the openness of the Internet, the attacker can easily control the information
spread in the public channel, tamper, replay, block the information, and then launch a
possible malicious attack, as shown in Figure 2. In this paper, the adversary $A$’s capabilities
in a multi-server environment are set as shown in Table 1.

Table 1. Attackers’ capabilities.

| Capabilities                                                                 |
|------------------------------------------------------------------------------|
| 1  | $A$ can enumerate every possibility of user identity and password.           |
| 2  | $A$ can extract the secret information from the smart card through side-channel technology. |
| 3  | $A$ can intercept, modify, or block messages propagated in the public channels. For a three-factor scheme, $A$ can capture two of the authentication factors simultaneously. |
| 4  | $A$ can get the long-term private keys of users, RC, or servers (only when evaluating forward secrecy). |
3. Review of Chatterjee et al.’s Scheme

In the highly cited paper published by Santanu Chatterjee et al., an authentication protocol based on an extended Chebyshev chaotic map for multi-server environments was proposed in 2016 [30]. This section will take Chatterjee’s protocol as an example to analyze and point out the security defects of this kind of authentication protocol.

Chatterjee et al.’s scheme mainly consists of the following phases: system setup phase, user registration phase, server registration phase, login and authentication phase, user password, and biometric update phase. Table 2 lists the symbols used in their scheme.

| Symbol          | Description                                      |
|-----------------|--------------------------------------------------|
| $H(\cdot)$      | Hash function                                    |
| $BH(\cdot)$     | Biological hash function                         |
| $T_x(\cdot)$    | Chebyshev polynomial                             |
| $E_k(\cdot)/D_k(\cdot)$ | Symmetric encryption/decryption algorithms    |
| $K_s, K_u$      | Private key of RC                                |
| $x_j$           | Private key of $S_j$                             |
| $ID_{U_i}$      | Identification of $U_i$                         |
| $ID_{S_j}$      | Identification of $S_j$                         |
| $PW_{U_i}$      | Password of $U_i$                                |
| $B_i$           | Biological information of $U_i$                 |
| $SC_i$          | Smart card of $U_i$                              |
| $SK_{ij}$       | Session key of $U_i$ and $S_j$                   |
| $\oplus$        | XOR operation                                     |
| $\parallel$     | Concatenation operation                           |

The detailed description of the scheme is as follows:
3.1. System Setup Phase

Step 1: The Registration Center RC randomly selects $K_S$ and $K_T$ from $[-\infty, +\infty]$.

Step 2: RC selects a secure hash function $H(\cdot)$, a biological hash function $BH(\cdot)$, a Chebyshev polynomial $T_x(\cdot)$, and a pair of symmetric encryption/decryption algorithms $E_k(\cdot)/D_k(\cdot)$. Then, $\{H(\cdot), BH(\cdot), T_x(\cdot), E_k(\cdot)/D_k(\cdot)\}$ will be passed onto the public.

3.2. Server Registration Phase

Step 1: The server $S_j$ sends its identity $ID_{S_j}$ to RC through a secure channel.

Step 2: After receiving the registration information, RC randomly selects $x_j$ and computes $T_{S_j}(K_S)$ and $T_{S_j}(K_T)$ and sends $\{x_j, T_{S_j}(K_S), T_{S_j}(K_T)\}$ back to $S_j$ through the secure channel.

3.3. User Registration Setup Phase

Step 1: The user $U_i$ selects his identity $ID_{U_i}$, password $PW_i$, and enters his biological information $B_i$. Next, $U_i$ and $RC$ calculate $D_I = H(ID_{U_i} \parallel R_i \parallel T_i)$, $b_i = BH(B_i)$, $RPW_i = H(D_I \parallel PW_i \parallel b_i \parallel R_i)$, $K_i = H(b_i \parallel R_i \parallel ID_i)$, $C_i = R_i \oplus H(b_i \parallel ID_{U_i} \parallel PW_i)$, and transmits the registration information $\{ID_i, T_i, K_i, C_i, RPW_i\}$ to RC through the secure channel.

Step 2: After receiving the registration request from $U_i$, RC randomly selects $x_i$ and computes the Chebyshev polynomials $T_{S_i}(K_S)$ and $T_{S_i}(K_T)$. Afterward, RC computes $S_K = K_i \oplus x_i$, $P = K_i + H(x_i \parallel K_i)$, $A_i = H(ID_{U_i} \parallel RPW_i \parallel T_i \parallel T_{S_i}(K_S) \parallel x_i \parallel P)$, writes $\{ID_i, T_i, A_i, T_{S_i}(K_S), C_i, S_K, P, T_{S_i}(K_T), C_i, m, 1 \leq j \leq m\}$ to smart card $SC_i$, and gives it to $U_i$, where $m$ is the number of servers in the system.

Step 3: After $U_i$ completes registration, RC selects a random number $U_{R_i}$ and calculates $U_{H_i} = H(T_{S_i}(K_T) \parallel U_{R_i})$. Finally, RC transmits $\{U_{H_i}, U_{R_i}, ID_i\}$ to all servers in the system through the secure channel.

3.4. Login and Authentication Phase

Step 1: The user $U_i$ inserts his smart card $SC_i$ into the terminal, inputs his identity $ID_i$, password $PW_i$, and obtains the biological information $B_i$. $SC_i$ calculates $b_i = BH(B_i)$, $R'_i = C_i \oplus H(b_i \parallel ID_{U_i} \parallel PW_i)$, $ID'_i = H(ID_{U_i} \parallel R'_i \parallel T'_i)$, $RPW'_i = H(ID'_i \parallel PW_i \parallel b_i \parallel R'_i)$, $K'_i = H(b_i \parallel R'_i \parallel ID'_i)$, $x'_i = SK_i \oplus K'_i$, $A'_i = H(ID_{U_i} \parallel RPW'_i \parallel T'_i \parallel T_{S_i}(K_T) \parallel x'_i \parallel P)$. The smart card verifies whether $A'_i = A_i$ is true or not; if not, $SC_i$ rejects the login request of $U_i$; otherwise, $SC_i$ obtains the current timestamp $T_{S_i}$, generates a random number $R_{N_i}$, and calculates $K_p = P \oplus H(x_i \parallel K'_i)$, $T_{x_i} = T_{x_i}(K_{S_i})$, $K_1 = H(T_{S_i}(K_S) \parallel ID_i \parallel ID_{S_i} \parallel T_{S_i})$. Finally, $SC_i$ sends the login request information $M_1 = \{ID_i, ID_{S_i}, E_{K_1}(ID_i \parallel ID_{S_i} \parallel T_{K_1} \parallel T_{S_i}(K_S) \parallel T_{x_i}(K_T) \parallel T_{S_i}(K_{S_i}) \parallel T_{x_i}(K_{S_i}) \parallel R_{N_i}, C_i, T_{S_i}, H(K_i \parallel T_{S_i} \parallel ID_i \parallel ID_{S_i} \parallel R_{N_i} \parallel T_{S_i}(K_T) \parallel T_{K_1})\}$ to the server $S_j$.

Step 2: The server $S_j$ receives $M_1$ and first verifies the validity of the timestamp $T_{S_i}$. If the timestamp is invalid, $S_j$ rejects the login request; otherwise, $S_j$ calculates $K'_j = H(T_{x_i}(K_S) \parallel ID_i \parallel ID_{S_i} \parallel T_{S_i})$ and decrypts $E_{K_1}(ID_i \parallel ID_{S_i} \parallel T_{K_1} \parallel T_{x_i}(K_S) \parallel T_{x_i}(K_T) \parallel T_{x_i}(K_{S_i}) \parallel R_{N_i} \parallel C_i)$ with $K'_j$ to get $ID_i, ID_{S_i}, T_{K_1}, T_{x_i}(K_S), T_{x_i}(K_T), T_{x_i}(K_{S_i})$, $RN_i, K_i$. $S_j$ uses the decrypted $ID_i$ to search for the corresponding $(U_{H_i}, U_{R_i})$ and verify whether $U_{H_i} = H(T_{x_i}(K_{S_i}) \parallel U_{R_i})$ is true; if not, $S_j$ terminates the session; otherwise, $S_j$ calculates $H(K_i \parallel T_{S_i} \parallel ID_i \parallel ID_{S_i} \parallel R_{N_i} \parallel T_{S_i}(K_T) \parallel T_{K_1})$, $T'_{K_1} = T_{S_i}(T_{x_i}(K_S))$, and compares the calculated result with the received corresponding value; if not, $S_j$ terminates the session; otherwise, $S_j$ authenticates $U_i$ successfully. Next, $S_j$ obtains the current
timestamp $T_{S_j}$ and calculates $T_{K_2} = T_{S_j}(T_{x_j}(K_{u_j}))$, $Y = K_i \oplus T_{K_2}$, $K_2 = H(T_{x_j}(K_{u_j})) \parallel ID_{S_j} \parallel ID_i \parallel TS_i \parallel TS_j \parallel RN_i \parallel RN_j \parallel T'_{K_2}$, $T_{K_3} = T_{x_j}(T_{x_j}(K_{u_j}))$, and the session key $SK_{ij} = H(ID_i \parallel ID_{S_j} \parallel TS_j \parallel TS_j \parallel RN_i \parallel RN_j \parallel T'_{K_2} \parallel T_{K_2} \parallel T_{K_3})$. Finally, $S_j$ transmits $M_2 = \{ID_i, ID_{S_j}, E_{S_j}(ID_i \parallel ID_{S_j} \parallel Y \parallel T_{x_j}(K_{u_j}) \parallel RN_j \parallel T_{K_3}), H(TS_i \parallel TS_j \parallel RN_i \parallel RN_j \parallel Y \parallel T_{K_3} \parallel T_{S_j}(K_{u_j}))\}$ back to user $U_i$.

Step 3: $U_i$ receives $M_2$ and verifies the validity of time stamp $T_{S_j}$. If $T_{S_j}$ is invalid, $U_i$ terminates the session; otherwise, $U_i$ computes $K_2 = H(T_{x_j}(K_{u_j})) \parallel ID_{S_j} \parallel ID_i \parallel TS_i \parallel TS_j \parallel RN_i \parallel T_{K_3}$ and decrypts $E_{S_j}(ID_i \parallel ID_{S_j} \parallel Y \parallel T_{x_j}(K_{u_j}) \parallel RN_j \parallel T_{K_3})$ to get $ID_i$, $ID_{S_j}$, $Y$, $T_{S_j}(K_{u_j})$, $RN_j$, and then $U_i$ computes $T'_{K_2} = K_i \oplus Y$. If $T'_{K_2} \neq T_{K_3}$ holds, $U_i$ authenticates $S_j$ successfully and calculates session key $SK_{ij} = H(ID_i \parallel ID_{S_j} \parallel TS_i \parallel TS_j \parallel RN_i \parallel RN_j \parallel T_{K_3} \parallel T'_{K_2} \parallel T'_{K_3})$.

The process of login and authentication phase is shown in Figure 3.

**Figure 3.** Login and authentication phase of Chatterjee et al.'s scheme.

\[ U_i \]

Insert $SC_i$, input $PW_i$, $B_i$

Compute $b_i = BH(B_i)$, $R'_i = C_i \oplus H(b_i \parallel ID_i \parallel PW_i)$

$ID' = H(ID_i \parallel R'_i \parallel TS_i)$, $RPW' = H(ID' \parallel PW_i \parallel b_i \parallel R'_i)$

$K'_i = H(b_i \parallel R'_i \parallel ID'_{i, j})$, $z'_i = SK_i \oplus K'_i$

$A'_i = H(ID_i \parallel RPW'_{i, j}) \parallel TS_i \parallel T_{S_j}(K_{u_j}) \parallel ID_i \parallel PW_i \parallel P$

Check $A'_i = A_i$?

Generate $TS_i, RN_i$

Compute $K_i = P \oplus H(x_i \parallel K'_i)$, $T_{K_1} = T_{S_j}(T_{x_i}(K_{u_i}))$

$T_{S_j}(K_{u_j}), T_{K_1} = H(T_{S_j}(K_{u_j})) \parallel ID_i \parallel ID_{S_j} \parallel TS_j$

$M_1 = \{ID_i, ID_{S_j}, TS_j, H(K_i \parallel TS_i \parallel ID_i \parallel ID_{S_j} \parallel RN_i \parallel T_{S_j}(K_{u_j}) \parallel T_{K_1}), E_{K_i}(ID_i \parallel ID_{S_j} \parallel TS_i \parallel T_{S_j}(K_{u_j}) \parallel T_{K_1} \parallel T_{S_j}(K_{u_j}) \parallel RN_i \parallel K_i)\}$

\[ S_j \]

Check $TS_i$

Use $K'_i$ for decryption, get $ID_i, ID_{S_j}, T_{K_1}, T_{S_j}(K_{u_j}), T_{S_j}(K_{u_j}), T_{K_1}$

Check $U_{T_{K_1}} = H(T_{S_j}(K_{u_j}) \parallel U_{T_{K_1}})$

Compute $H(K_i \parallel TS_i \parallel ID_i \parallel ID_{S_j} \parallel RN_i \parallel T_{S_j}(K_{u_j}) \parallel T_{K_1} \parallel T_{S_j}(K_{u_j}) \parallel T_{K_1}$)

and compares the calculated result with the received corresponding value

Generate $TS_j$

Compute $T_{K_3} = T_{x_j}(T_{S_j}(K_{u_j}))$, $Y = K_i \oplus T_{K_3}$, $T_{K_3} = T_{x_j}(T_{S_j}(K_{u_j}))$

$K_2 = H(T_{x_j}(K_{u_j}) \parallel ID_i \parallel ID_{S_j} \parallel TS_i \parallel TS_j \parallel RN_i \parallel T'_{K_2}$)

$SK_{ij} = H(ID_i \parallel ID_{S_j} \parallel TS_i \parallel TS_j \parallel RN_i \parallel RN_j \parallel T_{K_3} \parallel T_{K_3})$

$M_2 = \{ID_i, ID_{S_j}, E_{K_i}(ID_i \parallel ID_{S_j} \parallel Y \parallel T_{S_j}(K_{u_j}) \parallel RN_j \parallel T_{K_3} \parallel T_{S_j}(K_{u_j}) \parallel H(TS_i \parallel TS_j \parallel RN_i \parallel RN_j \parallel Y \parallel T_{K_3} \parallel T_{S_j}(K_{u_j}))\}$

Check $TS_j$

Compute $K_2 = H(T_{S_j}(K_{u_j}) \parallel ID_i \parallel ID_{S_j} \parallel TS_i \parallel TS_j \parallel RN_i \parallel T_{K_3}$

Use $K_2$ for decryption, get $ID_i, ID_{S_j}, Y, T_{S_j}(K_{u_j}), RN_j, T_{K_3}$

Compute $T'_{K_3} = K_i \oplus Y$, $T_{K_3} = T_{x_j}(T_{S_j}(K_{u_j}))$

Check $T'_{K_3} = T_{K_3}$?

Compute $SK_{ij} = H(ID_i \parallel ID_{S_j} \parallel TS_i \parallel TS_j \parallel RN_i \parallel RN_j \parallel T_{K_3} \parallel T_{K_3}$)
3.5. User Password and Biometric Update Phase

Step 1: The user $U_i$ inserts his smart card $SC_i$ into the terminal, inputs his identity $ID_{U_i}$ and password $PW_{U_i}$, and collects his biometric $B_i$. $SC_i$ calculates $b_i = BH(B_i)$, $R_i = C_i \oplus H(b_i \| ID_{U_i} \| PW_{U_i})$, $ID_i' = H(ID_{U_i} \| R_i' \| T_i)$, $RPW_i' = H(ID_i' \| PW_i) \|$ $b_i \| R_i')$, $K_i = H(b_i \| R_i' \| ID_i')$, $x_i = SK_i \oplus K_i$, $A_i = H(ID_{U_i} \| RPW_i' \| T_i \| T_{x_i}(K_u)) \|$ $x_i' \| P)$. $SC_i$ verifies whether $A_i' = A_i$ is established; if not, the smart card rejects the login request of $U_i$; otherwise, $SC_i$ makes $U_i$ enter a new password and new biometric information.

Step 2: $U_i$ enters the new password $PW_i^{new}$ and new biometric $B_i^{new}$. Then, the smart card computes $b_i^{new} = BH(B_i^{new})$, $C_i^{new} = R_i' \oplus H(b_i^{new} \| ID_{U_i} \| PW_{i}^{new})$, $RPW_i^{new} = H(ID_i' \| PW_i^{new} \| b_i^{new} \| R_i')$, $A_i^{new} = H(ID_{U_i} \| RPW_{i}^{new} \| T_i \| T_{x_i}(K_u)) \|$ $x_i' \| P)$. $SC_i$ replaces $A_i^{new}$, $C_i^{new}$ with $A_i$ and $C_i$.

4. Cryptanalysis of Chatterjee et al.’s Scheme

4.1. User Un-Traceability

The adversary can intercept the information transmitted between the user and the server in the public channel. Due to the protection of hash function, the adversary cannot directly extract the user’s identity. However, in the login request information of user $u$, $ID_i = H(ID_{U_i} \| R_i \| T_i)$, $R_i = C_i \oplus H(b_i \| ID_{U_i} \| PW_i)$ where $T_i$ is the time stamp obtained when $U_i$ registers. It can be found that the $ID_i$ generated by the same user in each login request is fixed. Therefore, it is easy for adversaries to determine whether two sessions are initiated by the same user through $ID_i$, so as to track the user’s behavior. Therefore, the protocol proposed by Chatterjee et al. cannot achieve user un-traceability.

4.2. Three-Factor Security

Chatterjee et al.’s protocol involves three security factors: user password, smart card, and user’s biometrics. Suppose that the adversary accidentally obtains the smart card and biometric $B_i$ of user $U_i$, the adversary can obtain the password of $U_i$ through the following operations:

Step 1: The adversary $A$ uses the side-channel attack technology [32] to extract the secret information $\{ID_i, T_i, A_i, T_{x_i}(K_u), C_i, SK_i, P, T_{x_i}(K_u), s<ID_{S_j}, T_{x_i}(K_u), 1 \leq j \leq m>\}$ stored in the smart card of $U_i$, and calculates $b_i = BH(B_i)$.

Step 2: $A$ guesses that the identity and password of $U_i$ are $(ID_{U_i}^{*}, PW_{i}^{*})$, where $ID_{U_i}^{*}$ and $PW_{i}^{*}$ are generated from identity space $D_id$ and password space $D_pw$, respectively.

Step 3: $A$ calculates $R_i^{*} = C_i \oplus H(b_i \| ID_{U_i}^{*} \| PW_{i}^{*})$, $ID_i^{*} = H(ID_{U_i}^{*} \| R_i^{*} \| T_i)$, $RPW_i^{*} = H(ID_i^{*} \| PW_{i}^{*} \| b_i \| R_i^{*})$, $K_i^{*} = H(b_i \| R_i^{*} \| ID_i^{*})$, $x_i^{*} = SK_i \oplus K_i^{*}$, $A_i^{*} = H(ID_{U_i}^{*} \| RPW_{i}^{*} \| T_i \| T_{x_i}(K_{u})) \|$ $x_i^{*} \| P)$. $A_i^{*}$ is correct; otherwise, go to Step 2.

According to the above steps, it takes $(5T_h) \cdot |D_id| \cdot |D_pw|$ to complete the offline password guessing attack, where $T_h$ is the time-consuming of hash function running once, and XOR operation can be ignored due to its small time-consuming. According to reference [33], $|D_id| \leq |D_pw| \leq 10^6$. Using the computing processor intel-i7-5500 3.6 g Hz in reference [34], $T_h \approx 0.564$μs, the adversary can complete the above attack within 33 days. If a high-performance cloud platform launches the attack, the user’s password can be guessed within a few hours.
4.3. User Impersonation Attack

Since \( \{ID_{S_j}, T_x(K_s)\}_{1 \leq j \leq m} \) is stored in each user’s smart card, malicious users can intercept the login request information of user \( U_i \) to initiate login request as user \( U_i \) and pass the authentication of server \( S_j \). The specific operations are as follows:

**Step 1:** The malicious user \( A \) intercepts \( M_1 = \{ID_{j'}, ID_{S_j}, E_{K_s}(ID_{j'} || ID_{S_j} || T_k || T_x(K_s) || RN_i || T_x(K_u) \} \) transmitted in the public channel.

**Step 2:** \( A \) calculates \( K_1 = H(T_x(K_s) || ID_{j'} || ID_{S_j} || TS_{i'}) \), where \( T_x(K_s) \) is extracted from the smart card of \( A \), \( ID_{i'}, ID_{S_j}, TS_{i'} \) is obtained from \( M_1 \). Then, \( A \) uses \( K_1 \) to decrypt \( E_{K_1}(ID_{j'} || ID_{S_j} || T_k || T_x(K_s) || T_x(K_u) || RN_i || K_1) \) to get \( ID_{j'}, ID_{S_j}, T_k, T_x(K_s), T_x(K_u), RN_i, K_1 \).

**Step 3:** Through the information obtained in Step 2, \( A \) can generate a new times-brand \( TS_{i'} \) and construct the legitimate request information \( M_1' = \{ID_{j'}, ID_{S_j}, E_{K_1}(ID_{j'} || ID_{S_j} || T_k || T_x(K_s') || T_x(K_u') || RN_i || K_1') \} \) and pass the authentication of server \( S_j \).

5. The Proposed Scheme

The proposed protocol includes the following phases: system setup phase, server registration phase, login and authentication phase, user registration phase, user password, and biometric update phase. The symbols used in the proposed protocol are shown in Table 3.

| Symbol | Description |
|--------|-------------|
| \( h( ) \) | Hash function |
| \( BH( ) \) | Biological hash function |
| \( T_x( ) \) | Chebyshev polynomial |
| \( x, y \) | Private key of \( RC \) |
| \( K_j \) | Private key of \( S_j \) |
| \( ID_{i'} \) | Identification of \( U_i \) |
| \( SID_{i'} \) | Identification of \( S_j \) |
| \( PW_i \) | Password of \( U_i \) |
| \( Bio_i \) | Biological information of \( U_i \) |
| \( SC_j \) | Smart card of \( U_i \) |
| \( SK_{ij} \) | Session key of \( U_i \) and \( S_j \) |
| \( \oplus \) | XOR operation |
| \( \parallel \) | Concatenation operation |
| \( mod \) | Modulus operation |

The detailed description of the agreement is as follows:

5.1. System Setup Phase

The registration center \( RC \) randomly selects \( x, y \) as the system master keys in \( [-\infty, +\infty] \). Next, \( RC \) selects a secure hash function \( h( ) \).

5.2. Server Registration Phase

**Step 1:** The server \( S_j \) selects its identity \( SID_{i'} \) and passes it to \( RC \) through a secure channel.

**Step 2:** After receiving \( SID_{i'} \), \( RC \) calculates \( K_j = h(SID_{i'} || y) \) and publishes information \( \{SID_{i'}, z\} \). Next, \( RC \) sends \( K_j \) back to \( S_j \) through the secure channel.
Step 3: $S_j$ receives $K_j$ and keeps it in secret.

5.3. User Registration Setup Phase

Step 1: The user $U_i$ selects his identity $ID_i$ and password $PW_i$ and enters his biometric $Bio_i$. Then, $U_i$ uses the biological hash function $BH(.)$ to get $b_i$ and calculates $PID_i = h(ID_i \parallel b_i)$, $PWB_i = h(PW_i \parallel b_i)$. Finally, $U_i$ transmits the registration information $\{ID_i, PID_i, PWB_i\}$ to $RC$ through a secure channel.

Step 2: After receiving $U_i$’s registration information, $RC$ computes $A_i = h(ID_i \parallel PWB_i) \mod n$, $B_i = h(PID_i \parallel x)$, $C_i = B_i \oplus PWB_i$, where $2^4 \leq n \leq 2^8$. Next, $RC$ calculates $D_{ij} = h(B_i \parallel K_j)$, $E_{ij} = B_i \oplus K_j$, $F_{ij} = D_{ij} \oplus h(B_i)$ where $1 \leq j \leq m$ and $m$ is the number of servers in the systems. At last, $\{A_i, C_i, E_{ij}, F_{ij}, n, h(.), h(x \parallel y), z\}$ are written into the smart card $SC_i$, and $SC_i$ is transmitted to $U_i$ via the secure channel.

Step 3: $U_i$ keeps $SC_i$ properly.

The process of the registration phase is shown in Figure 4.

![Figure 4. Registration phase of proposed scheme.](image-url)
5.4. Login and Authentication Phase

Step 1: The user \( U_i \) inserts his smart card \( SC_i \) into the terminal and inputs his identity \( ID_i \), password \( PW_i \), and biometric \( Bio_i \). \( SC_i \) calculates \( b_i = BH(Bio_i) \), \( PID_i = h(ID_i \| b_i) \), \( PWB_i = h(PW_i \| b_i) \), and verifies whether \( A_i \neq h(ID_i \| PWB_i) \mod n \) is established; if not, \( SC_i \) terminates the session; otherwise, \( SC_i \) generates a random number \( n_i \), selects the identity of the server to be accessed \( SID_j \), and calculates \( N_i = T_{n_i}(z), P_j = E_{ij} \oplus h(SID_j \| h(x \| y) \| N_i), N_k = h(B_i \| N_i), D_{ij} = F_{ij} \oplus h(B_i), CID_{ij} = PID_i \oplus h(P_{ij} \| B_i), M_1 = h(B_i \| D_{ij} \| CID_{ij} \| N_i) \). Finally, \( U_i \) sends \( \{P_{ij}, CID_{ij}, N_i, M_1\} \) to server \( S_j \).

Step 2: Upon the receipt of login request from \( U_i, S_j \) computes \( E_{ij} = P_{ij} \oplus h(SID_j \| h(x \| y) \| N_i), B_i = E_{ij} \oplus K_{ij}, D_{ij} = h(B_i \| K_{ij}), N_k = h(B_i \| N_i), M_1^j = h(B_i \| D_{ij} \| CID_{ij} \| N_k) \), and verifies that \( M_1 \) and \( M_1^j \) match. If not, \( S_j \) terminates the session. Otherwise, \( S_j \) generates a random number \( n_j \) and calculates \( N_j = T_{n_j}(z), PID_j = CID_{ij} \oplus h(P_{ij} \| B_i), M_2 = h(PID_j \| P_{ij} \| D_{ij} \| B_i \| SID_j \| N_j), M_3 = N_k \oplus N_j \). Afterward, \( S_j \) sends \( \{M_2, M_3\} \) to \( U_i \).

Step 3: \( U_i \) receives \( \{M_2, M_3\} \) and calculates \( N_j = M_3 \oplus N_k \). \( U_i \) verifies whether \( M_2 \neq h(PID_i \| P_{ij} \| D_{ij} \| B_i \| SID_j \| N_j) \) is established; if not, \( U_i \) terminates the session; otherwise, \( U_i \) identifies \( S_j \) as legal. After that, \( U_i \) computes \( M_4 = h(B_i \| D_{ij} \| N_j \| SID_j), T_{ij} = T_{n_j}(N_j) \), and gets the session key \( SK_{ij} = h(PID_i \| P_{ij} \| T_{ij}) \). Ultimately, \( \{M_4\} \) is delivered to \( S_j \).

Step 4: \( S_j \) receives \( \{M_4\} \) and verifies whether \( M_4 \neq h(B_i \| D_{ij} \| N_j \| SID_j) \) holds; if not, \( S_j \) terminates the session; otherwise, \( S_j \) certifies that the identity of \( U_i \) is legal. Furthermore, \( S_j \) computes \( T_{ij} = T_{n_j}(N_j) \) and reaches the same session key with \( U_i: SK_{ij} = h(PID_i \| P_{ij} \| T_{ij}) = h(PID_i \| P_{ij} \| T_{ij}) = SK_{ij} \).

The process of login and authentication phase is shown in Figure 5.

5.5. User Password and Biometric Update Phase

Step 1: The user \( U_i \) inserts his smart card \( SC_i \) into the terminal, inputs his identity \( ID_{U_i} \), password \( PW_{i} \), and collects his biometric \( B_{i} \). \( SC_i \) calculates \( b_i = BH(Bio_i) \), \( PID_i = h(ID \| b_i) \), \( PWB_i = h(PW_i \| b_i) \) and verifies whether \( A_i \neq h(ID_i \| PWB_i) \mod n \) is established; if not, \( SC_i \) terminates the session; otherwise, \( SC_i \) makes \( U_i \) enter a new password and new biological information.

Step 2: \( U_i \) enters the new password \( PW_{i}^{new} \) and new biometric \( Bio_{i}^{new} \). Then, \( SC_i \) computes \( b_i^{new} = BH(Bio_{i}^{new}) \), \( PID_{i}^{new} = h(ID_{i} \| b_{i}^{new}) \), \( PWB_{i}^{new} = h(PW_{i}^{new} \| b_{i}^{new}) \), \( A_i^{new} = h(ID_{i} \| PWB_{i}^{new}) \mod n, C_i^{new} = C_i \oplus PWB_i \oplus PWB_{i}^{new} \).

Step 3: \( SC_i \) replaces \( A_i^{new}, C_i^{new} \) with \( A_i \) and \( C_i \).
Step 2: Upon the receipt of login request from $U_i$, $S_j$ computes 
$$E_{ij} = P_{ij} \oplus h(SID_j \parallel h(x||y)||N_i)$$
Compute $B_i = E_{ij} \oplus K_j$, $D_{ij} = F_{ij} \oplus h(B_i)$
$n_i$, compute $N_i = T_{n}(z)$
$C_{ID_{ij}} = PID_i \oplus h(P_{ij}||B_i)$
$M_1 = h(B_i||D_{ij}||C_{ID_{ij}}||N_k)$

\[ \{P_{ij}, C_{ID_{ij}}, M_1, N_i \} \]

Compute $N_j = M_3 \oplus N_k$
Check $M_2? = h(PID_i||P_{ij}||D_{ij}||B_i||SID_j||N_j)$
Compute $M_4 = h(B_i||D_{ij}||N_j||SID_j)$
$T_{ij} = T_{n}(N_j)$
$SK_{ij} = h(PID_i||P_{ij}||T_{ij})$

\[ \{M_2, M_3 \} \]

Check $M_4? = h(B_i||D_{ij}||N_j||SID_j)$
$T_{ji} = T_{n}(N_i)$
$SK_{ji} = h(PID_i||P_{ij}||T_{ji})$

Figure 5. Login and authentication phase of the proposed scheme.

6. Security Analysis

6.1. Provable Security

Based on the BPR2000 model [35], the following is the description of the random oracle model and the definition of $AKE$ security:
(1) Participants

As participants, users $U$ and servers $S$ have many different instances, which are called oracle. The $i$-th instance of $U$ and the $j$-th instance of $S$ are denoted as $U^i$ and $S^j$, respectively, and any instance can be denoted as $I$ uniformly.

(2) Queries

$\text{Execute}(U^i, S^j)$: The query captures the passive eavesdropping of the scheme, and its output includes all the communication records of the scheme between $U^i$ and $S^j$.

$\text{Send}(U^i, \text{start})$: This query indicates a login request that triggers the scheme startup and outputs $U^i$.

$\text{Send}(I^i, m)$: This query captures active attacks. More precisely, the adversary $A$ constructs a forged message $m$ by interrupting and intercepting messages. Then, $A$ sends $m$ to $I^i$ and gets a response from $I^i$.

$\text{Reveal}(I^i)$: If $I^i$ accepts the session and generates the session key $SK$, it will respond to $A$ with $SK$.

$\text{Corrupt}(I^i, a)$: The query simulates the capture of any two of the three security factors. If $a = 1$ and $I = U$, the user password and all parameters stored in the smart card are returned to $A$. If $a = 2$ and $I = U$, the user biometrics and all parameters stored in the smart card are returned to $A$. If $a = 3$ and $I = U$, the user password and biometrics are returned to $A$. If $a = 1$ and $I = S$, the long-term private key of the server is returned to $A$.

$\text{Test}(I^i)$: The oracle tosses a coin $b \in (0,1)$; if $b = 1$, it returns the session key; if $b = 0$, it returns a random number with the same length as the session key.

(3) Partnership

$U^i$ and $S^j$ are called partnerships if: (i) $U^i$ and $S^j$ are accepted; (ii) and have the same session identifier ($sid$), that is, $sid^i_j = sid^j_i$; (iii) the partner identifier of $S^j$ is $U^i$ and vice versa.

(4) Freshness

A user instance or server instance is called fresh if (i) $I$ has calculated an acceptable session key; (ii) $A$ has not made any $\text{Reveal}$ queries to $I$ or its partners. (iii) From the beginning of the game, $A$ makes $\text{Corrupt}$ query to $I$ or its partners at most once.

Definition 1. The adversary $A$ outputs the result of guess $b'$ through Test queries. If $b' = b$, $A$ wins the game. The advantage probability of breaking the security of the protocol $P$ is defined as:

$Adv_{A}^{\text{AKE}}(A) = 2\Pr[b' = b] - 1$. If the probability $Adv_{A}^{\text{AKE}}(A)$ is negligible for any probabilistic polynomial time adversary $A$, the protocol $P$ is AKE secure.

Theorem 1. Suppose the adversary $A$ operates $q_{\text{send}}$ Send queries, $q_{\text{exe}}$ Execute queries and $q_h$ Hash queries to break the AKE security of the protocol. $Adv_{A}^{\text{CMCDH}}(t)$ represents the advantage probability of $A$ solving CMCDH problem in the polynomial time $t$, then we have:

$$Adv_{A}^{\text{AKE}}(A) \leq 2C's_{\text{send}} + \frac{q_{\text{send}} + q_h + q_h^2}{2^{t-1}} + \frac{2(q_{\text{send}} + q_{\text{exe}})^2}{p} + 2q_h \cdot Adv_{A}^{\text{CMCDH}}(t')$$

(1)

where $C'$ and $s'$ are the CDF-Zipf regression parameters of password space, $l$ is the bit length of hash function output, $t' \leq t + (q_{\text{send}} + q_{\text{exe}} + 1)T_c$, and $T_c$ represents the running time of extended chaotic map operation.

Proof. Game $G_i$, $0 \leq i \leq 5$ is created to prove that the proposed scheme is provably secure, and $Suc_i$ stands for $A$ correctly guessing $b$ in game $G_i$ using Test queries.
Game $G_0$: This game simulates the real attack in the random oracle model. We can get:

$$Adv_{p}^{Ake}(A) = |2\Pr[Suc_0] - 1|$$  \hspace{1cm} (2)

Game $G_1$: This game manages Hash list $L_h$ while simulating random oracle. Then we get:

$$\Pr[Suc_1] = \Pr[Suc_0]$$  \hspace{1cm} (3)

Game $G_2$: In game $G_2$, if there is a collision of interactive information or a collision of Hash query results, the game ends; otherwise, $G_2$ simulates all queries in $G_1$. According to the birthday paradox [36], the collision probability of the result of Hash query is $\frac{q^2}{2^h}$ and the collision probability of interaction information is \((\frac{q_{send}+q_{exe}}{2^h})^2\); therefore, we derive the following result:

$$\Pr[Suc_2] - \Pr[Suc_1] \leq \frac{q^2}{2^{l+1}} + \frac{(q_{send} + q_{exe})^2}{2p}$$  \hspace{1cm} (4)

Game $G_3$: In game $G_3$, if $A$ guesses the information $M_1$ and $M_2$ used for authentication correctly, the game ends; otherwise, $G_3$ is the same simulation as the previous game; therefore, we derive the following result:

$$\Pr[Suc_3] - \Pr[Suc_2] \leq \frac{q_{send}}{2^l}$$  \hspace{1cm} (5)

Game $G_4$: In this game, $A$ guesses the session key without asking the corresponding random oracle. Therefore, this game is indistinguishable from the previous game, except that $A$ makes queries for $SK_j = h(PID_i \parallel P_j \parallel T_i) = h(PID_i \parallel P_j \parallel T_j) = SK_j$. Thus, we get that:

$$\Pr[Suc_4] - \Pr[Suc_3] \leq q_{a}Adv_{A}^{CMCDH}(t') + \frac{q^2_h}{2^l}$$  \hspace{1cm} (6)

where $t' \leq t + (q_{send} + q_{exe} + 1)T_c$.

Game $G_5$: This game is similar to the previous game, but the only difference is the Test query. If $A$ performs the Test query on $h(PID_i \parallel P_j \parallel T_{n,n}(z))$, the game will be terminated. Therefore, the maximum probability of obtaining session key by random oracle query is $\frac{q^2}{2^l}$. Moreover, if $Corrupt(U^1, 2)$ query is executed, $Corrupt(U^1, 1)$ and $Corrupt(U^1, 3)$ can no longer be queried. According to reference [37], in the case of $q_{send}$ times of send query for online guess, the probability of getting the password is at most $C^t q_{send}$. According to the definition of freshness, $A$ can perform Test($U^1$) query after performing $Corrupt(U^1, a)$ query. As a result, outdated copies are used in old games (perfect forward secrecy). Therefore, the maximum probability of $A$ getting $T_{n,n}(z)$ is $\frac{(q_{send} + q_{exe})^2}{2p}$. Then, we get:

$$\Pr[Suc_5] - \Pr[Suc_4] \leq C^tq_{send} + \frac{q^2_h}{2^l} + \frac{(q_{send} + q_{exe})^2}{2p}$$  \hspace{1cm} (7)

If $A$ does not request any random oracle query with valid input, then the game has no advantage to distinguish the real $SK$ from the random string with the same length, so we get:

$$\Pr[Suc_5] = \frac{1}{2}$$  \hspace{1cm} (8)

According to Formulas (2), (3), and (8), we come to the conclusion

$$Adv_{p}^{Ake}(A) = 2 \times [(\Pr[Suc_5] - \Pr[Suc_4]) + (\Pr[Suc_4] - \Pr[Suc_3]) + (\Pr[Suc_3] - \Pr[Suc_2]) + (\Pr[Suc_2] - \Pr[Suc_1])] 
\leq 2C^t q_{send} + \frac{q^2_h + q^2}{2^l} + \frac{2(q_{send} + q_{exe})^2}{2p} + 2q_h \cdot Adv_{A}^{CMCDH}(t')$$  \hspace{1cm} (9)
6.2. BAN-Logic

Burrow, Abadi, and Needham proposed BAN-logic [38] in 1989. BAN-logic is a belief-based modal logic, which can be used to describe and verify authentication protocols. When using BAN-logic to analyze the security of authentication protocol, we first need to idealize the interaction information in the protocol, then make initialization assumptions according to the specific situation, and finally get the expected goal through reasoning rules. Table 4 introduces the notations for the BAN-logic, and some basic rules are described in Table 5.

Table 4. BAN-logic notations.

| Symbol | Description |
|--------|-------------|
| P ≡ X  | P believes X. |
| P ≪ X  | P sees . |
| P ⊳ X  | P sends X. |
| P ⊳ X  | P has jurisdiction over X. |
| #(X)   | X is fresh. |
| (X,Y)  | X or Y is part of (X,Y). |
| (X)K   | Use the key K to compute X. |
| P SK ↔ Q | P and Q reach shared key SK. |

Table 5. Basic logical postulates of BAN-logic.

| Symbol | Description |
|--------|-------------|
| Message-meaning rule | \( P ≡ (P ⊳ Q) \), \( P ≡ (X)K \) |
| Freshness-conjunctenation rule | \( P ≡ (X,Y) \), \( P ≡ (Q-X) \) |
| Nonce verification rule | \( P ≡ (Q) \), \( P ≡ (X)K \) |
| Jurisdiction rule | \( P ≡ (Q) \), \( P ≡ (X)K \) |
| Believe rule | \( P ≡ (Q) \), \( P ≡ (X)K \) |

(1) The idealized form of the proposed scheme

Message 1: \( U_i \rightarrow S_j : (PID_i, P_{ij})_{U_i \leftrightarrow S_j}, P_{ij}, M_1, N_i \)

Message 2: \( S_j \rightarrow U_i : (PID_i, P_{ij}, D_{ij}, SID_j, N_j)_{U_i \leftrightarrow S_j}, M_3 \)

(2) Verification goals

Goal 1: \( U_i \equiv (U_i SK S_j) \).

Goal 2: \( U_i \equiv (U_i SK S_j) \).

Goal 3: \( S_j \equiv (U_i SK S_j) \).

Goal 4: \( S_j \equiv (U_i SK S_j) \).

(3) Assumptions about the initial state

A1: \( U_i \equiv #(n_i, n_j) \).

A2: \( S_j \equiv #(n_i, n_j) \).

A3: \( U_i \equiv (U_i SK S_j) \).

A4: \( S_j \equiv (U_i SK S_j) \).

A5: \( U_i \equiv (U_i SK S_j) \).

A6: \( S_j \equiv (U_i SK S_j) \).
(4) Proofs

Step 1: According to Message 1, we know that $S_j < (PID_i \cdot P_{ij})_{U_i \leftrightarrow S_j}$.

Step 2: According to Step 1, A4, and the message-meaning rule, we obtain the following:

$S_j \equiv U_i \equiv (PID_i \cdot P_{ij})_{U_i \leftrightarrow S_j}$.

Step 3: According to A2, freshness-conjunction rule, $P_{ij} = E_{ij} \oplus h(SID_j \parallel h(x \parallel y) || N_i)$, and $N_i = T_{n_i}(z)$, the following can be inferred: $S_j \equiv \#(PID_i, P_{ij})$.

Step 4: From Step 2, Step 3, and the nonce verification rule, we get that:

$S_j \equiv U_i \equiv (PID_i \cdot P_{ij})_{U_i \leftrightarrow S_j}$.

Step 5: From Step 4, A4, and $SK = h(PID_i \parallel P_{ij} \parallel T_{ij})$, we prove Goal 4:

$S_j \equiv U_i \equiv (U_i \leftarrow S_j)$.

Step 6: According Step 5, A6, and the jurisdiction rule, we prove Goal 3: $S_j \equiv (U_i \leftarrow S_j)$.

Step 7: According to Message 2, we know that $U_i < (PID_i \cdot P_{ij} \cdot D_{ij} \cdot SID_i \cdot N_j)_{U_i \leftrightarrow S_j}$.

Step 8: According to Step 7, A3, and the message-meaning rule, we obtain the following:

$U_i \equiv S_j \equiv (PID_i \cdot P_{ij} \cdot D_{ij} \cdot SID_i \cdot N_j)_{U_i \leftrightarrow S_j}$.

Step 9: According A1, freshness-conjunction rule, $N_i = T_{n_i}(z)$, the following can be inferred: $U_i \equiv \#(PID_i, P_{ij} \cdot D_{ij} \cdot SID_i \cdot N_j)$.

Step 10: From Step 8, Step 9, and the nonce verification rule, we get that:

$U_i \equiv S_j \equiv (PID_i \cdot P_{ij} \cdot D_{ij} \cdot SID_i \cdot N_j)_{U_i \leftrightarrow S_j}$.

Step 11: From Step 10, A4, and , we prove Goal 2: $U_i \equiv S_j \equiv (U_i \leftarrow S_j)$.

Step 12: According Step 11, A5 and jurisdiction rule, we prove Goal 1: $U_i \equiv (U_i \leftarrow S_j)$.

It can be seen from Goal 1, Goal 2, Goal 3, and Goal 4 that the mutual authentication between user $U_i$ and server $S_j$ is completed, and the session key $SK$ trusted by both parties is reached.

6.3. Informal Security Analysis

The new scheme can effectively improve the shortcomings of Chatterjee et al.’s scheme. First of all, the new protocol ensures that the information related to user identity and security factors are used reasonably in the process of generating login request information, which can effectively resist the user impersonation attack. Secondly, in the verification phase of smart cards, the modular operation is introduced, which can avoid the offline password guessing attack so as to achieve three-factor security. Finally, the construction of user login request information needs the participation of random numbers to ensure the realization of user un-traceability.

On the other hand, according to the description of the login and authentication phase of the new protocol, only with the $ID$, password, biological information, and smart card of the legal user $U_i$, the user can generate the legal login request information while only the server $S_j$ with the legal $K_j$ can generate the legal response information. Therefore, on the basis of ensuring the mutual authentication between the user and the server, the server $S_j$ can get the correct $PID_i$ by calculating $PID_i = CID_{ij} \oplus h(P_{ij} \parallel (E_{ij} \oplus K_j))$. Due to the semi-group property of the extended Chebyshev polynomials, $T_{ij} = T_{n_i}(N_j) = T_{n_i}(N_i) = T_{ij}$, $U_i$ and $S_j$ reach the session key $SK_{ij} = h(PID_i \parallel P_{ij} \parallel T_{ij}) = h(PID_i \parallel P_{ij} \parallel T_{ij}) = SK_{ij}$ for future sessions. They complete the session key agreement, and the contributions to session key generation are equal. Next, we make a specific security analysis of our proposed protocol.
(1) Anonymity and un-traceability

In the login and authentication phase, the adversary can intercept the login request information of the user and the response information of the server. Obviously, under the protection of Hash function, the adversary cannot obtain the user’s identity. Therefore, the proposed scheme can achieve user anonymity. On the other hand, the construction of $P_{ij}, CID_{ij}, N_i, M_1, M_2, M_3,$ and $M_4$ is related to the random number $n_i$ or $n_j$. Therefore, the interactive information generated in each session is different. Even if the adversary intercepts the message, it is still unable to determine whether two sessions originate from the same user. Therefore, the new protocol can achieve user un-traceability.

(2) Perfect forward secrecy

Suppose that the adversary accidentally obtains the private keys of RC: $x$ and $y_i$ and intercepts the information $\{P_{ij}, CID_{ij}, N_i, M_1, M_2, M_3, M_4\}$ propagated in the public channel. The adversary can compute $E_{ij} = P_{ij} \oplus h(SID_j \parallel h(x \parallel y) \parallel N_i), B_i = E_{ij} \oplus h(SID_j \parallel y), PID_i = CID_{ij} \oplus h(P_{ij} \parallel B_i), N_k = h(B_i \parallel N_i), N_j = M_3 \oplus N_k$. However, it is a CMCDH problem to get $T_{ij} = T_{n_i}(N_i) = T_{n_j}(N_i) = T_{ij}$ in polynomial time from the known information. Therefore, the adversary is still unable to calculate the session key between user $U_i$ and $S_j$, and the perfect forward secrecy of the new scheme is realized.

(3) Mutual authentication

According to the description of the new scheme, only with $U_i$’s identity, password, smart card, and biometrics can the legitimate login request information be generated. The server can authenticate the $U_i$’s identity by verifying the legitimacy of the received information. On the other hand, only the server $S_j$ with legal $K_i$ can correctly respond to the user’s login request information. Therefore, the new scheme realizes the mutual authentication between the user and the server.

(4) Session key agreement

Based on the description of the new scheme, the user and the server can reach the session key for future communication after completing the login and authentication phase $SK_{ij} = h(PID_i \parallel P_{ij} \parallel T_{ij}) = h(PID_i \parallel P_{ij} \parallel T_{ij}) = SK_{ij}$.

(5) Three-factor security

For the three-factor authentication protocol, the difficulty of breaking through the user password is obviously lower than the difficulty of breaking through the secret information of smart cards or user biometrics. Suppose the adversary accidentally obtains the smart card and biometrics of $U_i$ and the secret information in the smart card is extracted through the side-channel technology. However, the verification $A_i = h(ID_i \parallel PW_{B_i}) \mod n$ performed by the smart card in the login phase is a fuzzy verification. Even if the adversary’s guess $\{ID_{i,U_i}, PW_{i}^{*}\}$ passes the above verification, the adversary still cannot confirm whether $PW_{i}^{*}$ is the real password of $U_i$. Specifically, through offline password guessing, the adversary can get $|D_{id}||D_{pw}|/2^8 \approx 2^{32}$ possible $\{ID_{i,U_i}, PW_{i}^{*}\}$ pairs. The adversary still needs to log in online (not offline) and traverse these user identity and password pairs to obtain the accurate user password. The server can identify the victim according to the adversary’s login request. By setting the threshold of login times, when the adversary’s online login times exceed the threshold, the server can refuse the adversary’s login request. The adversary cannot log into the system many times, so he cannot get the correct one of the $2^{32}$ possible passwords. Therefore, the new protocol can achieve three-factor security.

(6) Good Repairability

In our proposed scheme, the user $U_i$’s private information stored in the smart card includes $A_i = h(ID_i \parallel PW_{B_i}) \mod n = h(ID_i \parallel h(PW_{B_i} \parallel b_i)) \mod n, C_i = B_i \oplus PW_{B_i} = h(h(ID_i \parallel b_i) \parallel x) \oplus PW_{B_i}, E_{ij} = B_i \oplus K_j = h(h(ID_i \parallel b_i) \parallel x) \oplus K_j$, and the server can authenticate the $U_i$’s login request. By setting the threshold of login times, when the adversary’s online login times exceed the threshold, the server can refuse the adversary’s login request. Therefore, the new protocol can achieve user un-traceability.
\[ F_{ij} = D_{ij} \oplus h(B_i) = D_{ij} \oplus h(h(ID_i \parallel b_i) \parallel x) \]. Therefore, \( U_i \)'s password and biometrics will directly affect the secret information. When the smart card \( SC_i \) is lost, \( U_i \) only needs to modify his password and biometrics to ensure the security of the system. Thus, our scheme has good repairability.

(7) Resistance of other known attacks

Insider attack: Insiders can get the registration information \( \{ID_i, PID_i, PWB_i\} \) of user \( U_i \). However, the information is protected by Hash function, and the attacker cannot extract the user’s password or biometrics. Therefore, the insider attack is invalid for the proposed new scheme.

Stolen verifier table attack: There is no password-related and biometric-related information table stored in the servers and \( RC \). Therefore, the stolen verifier table attack is infeasible in our proposed scheme.

Temporary information leakage attack: In our proposed scheme, the user \( U_i \) and the server \( S_j \) reach a session key \( SK_{ij} = h(PID_i \parallel P_{ij} \parallel T_{ij}) = h(h(ID_i \parallel P_{ij} \parallel T_{ij}) = SK_{ij}. \) Therefore, even if an adversary captured the temporary information \( n_i \) and \( n_j \), he could not launch a temporary information leakage attack without \( PID_i \). As a result, our proposed scheme can resist a temporary information leakage attack.

Replay attack: According to the description of the proposed protocol, the user and the server generate the new random number \( n_i \) and \( n_j \) in the authentication phase. Both sides can easily find replay attacks by checking the validity of the received message. Therefore, the new protocol can effectively resist replay attacks.

DoS attack: After receiving the login request from \( U_i \), the server \( S_j \) verifies whether \( M^*_1 \neq M_1 \) holds. Only \( U_i \) calculates the legitimate login request information according to his identity, password, biometrics, and smart card and can pass the verification. Therefore, \( S_j \) can confirm that the login request is from \( U_i \), which can effectively reject a large number of invalid login requests from attackers.

According to the previous analysis and proof, we also know that the new scheme can resist user impersonation attacks, server spoofing attacks, man-in-the-middle attacks, offline password guessing attacks, and smart card stolen attacks.

7. Performance Analysis

In this section, we will compare the performance of the proposed new protocol with other authentication protocols based on the extended chaotic map in multi-server environments, including the comparison of computation cost and communication cost. Since the registration phase of users and servers only occurs once, and users do not frequently update their passwords and biometrics, this section only discusses the performance comparison between the login and authentication phases.

7.1. Comparison of Computing Costs

The new scheme and other similar protocols [30,39–41] all use fuzzy extractor algorithm or bio-hash function to extract users’ biometrics for protocol design. According to literature [42,43], the time cost of the fuzzy extractor algorithm and bio hash function is considered equal. Therefore, the user biometric extraction operation is ignored in the comparison of computation cost.

The comparison between the new proposed protocol and the protocols proposed by Chatterjee et al. [30], Lee et al. [39], Irshad et al. [40], and Braeken et al. [41] is shown in Table 6. The symbols used in the table have the following meanings:
The proposed scheme is at a better level compared with similar protocols, and it has good
protocol and Chatterjee et al. [30], Lee et al. [39], Irshad et al. [40], and Braeken et al. [41].

Running time of operations (millisecond).

Table 7.

| Protocol      | User          | Server         |
|---------------|---------------|----------------|
| Chatterjee et al. | $10T_h + 3T_{ch} + 2T_{c/d}$ ≈ 85.46 | $6T_h + 3T_{ch} + 2T_{c/d}$ ≈ 8.432 |
| Lee et al.    | $12T_h + 3T_{ch}$ ≈ 69.06 | $7T_h + 3T_{ch}$ ≈ 6.732 |
| Irshad et al. | $7T_h + 4T_{ch}$ ≈ 87.58 | $4T_h + 4T_{ch}$ ≈ 8.656 |
| Braeken et al. | $7T_h + 3T_{ch} + T_{c/d}$ ≈ 75.26 | $6T_h + 3T_{ch} + T_{c/d}$ ≈ 7.552 |
| Proposed     | $10T_h + 2T_{ch}$ ≈ 47.04 | $8T_h + 2T_{ch}$ ≈ 4.688 |

$T_h$: Time to execute a general hash operation.
$T_{c/d}$: Time to execute a symmetric encryption/decryption algorithm.
$T_{ch}$: Time to execute a chaotic map operation.

(The computation overhead of XOR operation is ignored.)

The running time of the user to perform the above operation is obtained from the
experiment of Intel Pentium 4 2600 MHZ processor and 1024 MB memory platform in
reference [30]. The server performance is assumed to be 10 times of 2.4 GHz processor
and 2GB memory platform. The running time of different operations on two platforms is
shown in Table 7.

Table 7. Running time of operations (millisecond).

| Protocol | User | Server |
|----------|------|--------|
| $T_{ch}$ | 21.02 | 2.104 |
| $T_{c/d}$ | 8.7  | 0.88  |
| $T_h$   | 0.5  | 0.06  |

From the results in Table 7, the proposed protocol has a lower computation cost than
the other four protocols for both the user and server sides.

7.2. Comparison of Communication Costs

For the convenience of comparison, it is assumed that the length of identification,
random number, timestamp, and other parameters involved in the new protocol and other
related protocols is 128 bits, the length of large prime $p$ is 128 bits, the output length of Hash
function is 160 bits (such as SHA-1), and the ciphertext length of the symmetric encryption
algorithm is an integral multiple of 128 bits (such as AES).

In the login and authentication phase of the proposed protocol, the interaction
information between the user and the server includes $\{P_{ij}, CID_{ij}, N_i, M_1, M_2, M_3\}$, and
$\{M_4\}$. The total length of interactive information is $160 \times 7 = 1120$ bits.

In the login and authentication phase of Chatterjee et al.’s protocol, the interaction
information between the user and the server includes $\{ID_i, ID_{Si}, E_{K_{j1}}(ID_i) \parallel ID_{Si} \parallel T_{K_l} \parallel T_{x_j}(K_u) \parallel T_{x_j}(K_u) \parallel RN_i \parallel K_i, TS_i, H(K_i) \parallel TS_i \parallel ID_i \parallel ID_{Si} \parallel RN_i \parallel T_{x_j}(K_u) \parallel T_{K_l}\}$ and $\{ID_i, ID_{Si}, E_{K_{j1}}(ID_i) \parallel ID_{Si} \parallel Y \parallel T_{x_j}(K_u) \parallel RN_i \parallel T_{K_l}, H(TS_i) \parallel TS_i \parallel RN_i \parallel Y \parallel T_{K_l} \parallel T_{x_j}(K_u), TS_i\}$. The total length of interactive information is $(128 + 128 + 128 \times 9 + 128 + 160) + (128 + 128 + 128 \times 7 + 128 + 160) = 3136$ bits.

Table 8 shows the comparison of communication cost between the proposed new
protocol and Chatterjee et al. [30], Lee et al. [39], Irshad et al. [40], and Braeken et al. [41].
From the comparison results, it can be seen that the communication cost of the new
proposed scheme is at a better level compared with similar protocols, and it has good
communication efficiency. It should be noted that our scheme is the only one that needs
two times of data transmission. This is to further strengthen the identity authentication of
the server to the user, to further ensure the security of the system. If we give up
the information $M_k$ that the user transmits to the server, the server can complete the
authentication of the user in the second step of the authentication phase and also generate
the session key $SK_{ji}$. We finally choose stronger security, and the communication overhead
carried by this is acceptable.

Table 8. Comparison of communication costs.

| Protocol             | Number of Messages | Length of Interactive Information |
|----------------------|--------------------|-----------------------------------|
| Chatterjee et al.    | 2 messages         | 3136 bits                         |
| Lee et al.           | 2 messages         | 1152 bits                         |
| Irshad et al.        | 2 messages         | 992 bits                          |
| Braeken et al.       | 2 messages         | 1216 bits                         |
| Proposed             | 3 messages         | 1120 bits                         |

8. Conclusions

In recent years, multi-server network architecture is widely used in practical applications.
Moreover, due to the insecurity of the network, abundant researches on authentication
and key agreement protocol for multi-server architecture have been put forward. In
2018, Chatterjee et al. published an authentication protocol based on an extended Cheby-
shev chaotic map for multi-server environments. However, through the analysis of their
protocol, we find that the protocol cannot achieve user un-traceability and three-factor se-
curity and cannot resist the counterfeiting attacks launched by malicious users. In order to
ensure the communication security of participants in multi-server network environments,
this study proposed a secure three-factor authentication protocol based on the extended
chaotic map. The new protocol can effectively avoid the security defects of Chatterjee’s
protocol and achieve all known security goals. Moreover, the proposed scheme is analyzed
and verified by the provable security and BAN logic. The results show that our scheme
realizes the mutual authentication of communication participants and can effectively resist
all kinds of attacks. Compared with other related protocols, the new protocol has good
practicability and can be applied to multi-server environments.

Author Contributions: Conceptualization, Y.Y.; Methodology, Y.Y. and O.T.; Supervision, O.T.;
Visualization, R.L.; Writing—original draft, Y.Y and R.L.; Writing—review & editing, Y.Y. and B.S. All
authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Keele University’s Central Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Segura Beltran, F. Development of Gauging Services in Spain. The Network of Stations of Jucar Hydrographic Confederation. 
   Boletin De La Asociacion De Geografos Espinales 2013, 63, 566–568.
2. Jia, M.; Komeily, A.; Wang, Y.; Srinivasan, R.S. Adopting Internet of Things for the development of smart buildings: A review of
   enabling technologies and applications. Autom. Constr. 2019, 101, 111–126. [CrossRef]
3. Satyanarayanan, M. Fundamental challenges in mobile computing. In Proceedings of the Fifteenth Annual ACM Symposium on
   Principles of Distributed Computing, Philadelphia, PA, USA, 23–26 May 1996.
4. Fu, Z.; Sun, X.; Liu, Q.; Zhou, L.; Shu, J. Achieving Efficient Cloud Search Services: Multi-Keyword Ranked Search over Encrypted
   Cloud Data Supporting Parallel Computing. IEICE Trans. Commun. 2015, 190–200. [CrossRef]
5. Tsai, C.-H.; Su, P.C. The application of multi-server authentication scheme in internet banking transaction environments. Inf.
   Syst. e-Bus. Manag. 2021, 19, 77–105. [CrossRef]
6. Li, L.-H.; Lin, L.-C.; Hwang, M.-S. A remote password authentication scheme for multiserver architecture using neural networks.
   IEEE Trans. Neural Netw. 2001, 12, 1498–1504. [CrossRef]
7. Lin, I.-C.; Hwang, M.-S.; Li, L.-H. A new remote user authentication scheme for multi-server architecture. *Future Gener. Comput. Syst.* 2003, 19, 13–22. [CrossRef]
8. Cao, X.; Zhong, S. Breaking a remote user authentication scheme for multi-server architecture. *IEEE Commun. Lett.* 2006, 10, 380–381. [CrossRef]
9. Lee, C.-C.; Lin, T.-H.; Chang, R.-X. A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. *Expert Syst. Appl.* 2011, 38, 13863–13870. [CrossRef]
10. Kim, H.-W.; Lim, S.-Y.; Lee, H.-J. Symmetric Encryption in RFID Authentication Protocol for Strong Location Privacy and Forward-Security. In Proceedings of the 2006 International Conference on Hybrid Information Technology, Cheju Island, Korea, 9–11 November 2006.
11. Fotouhi, M.; Bayat, M.; Das, A.K.; Far, H.A.N.; Pournaghi, S.M.; Doostari, M. A lightweight and secure two-factor authentication scheme for wireless body area networks in health-care IoT. *Comput. Netw.* 2020, 177, 107333. [CrossRef]
12. Sadri, M.J.; Asaar, M.R. A lightweight anonymous two-factor authentication protocol for wireless sensor networks in Internet of Vehicles. *Int. J. Commun. Syst.* 2020, 33, e4511. [CrossRef]
13. Kwon, D.; Yu, S.; Lee, J.; Son, S.; Park, Y. WSN-SLAP: Secure and Lightweight Mutual Authentication Protocol for Wireless Sensor Networks. *Sensors* 2021, 21, 936. [CrossRef] [PubMed]
14. Hathal, W.; Cruickshank, H.; Sun, Z.; Maple, C. Certificateless and Lightweight Authentication Scheme for Vehicular Communication Networks. *IEEE Trans. Veh. Technol.* 2020, 69, 16110–16125. [CrossRef]
15. Tu, Y.-J.; Gaurav, K.; Selwyn, P. Security of lightweight mutual authentication protocols. *J. Supercomput.* 2020, 77, 4565–4581. [CrossRef]
16. Yoon, E.-J.; Yoo, K.-Y. Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. *J. Supercomput.* 2013, 63, 235–255. [CrossRef]
17. Kim, H.; Kim, H.; Jeon, W.; Jeon, W.; Lee, K.; Lee, K.; Lee, Y.; Lee, Y.; Won, D.; Won, D. Cryptanalysis and Improvement of a Biometrics-Based Multi-server Authentication with Key Agreement Scheme. In Proceedings of the International Conference on Computational Science and Its Applications, Salvador, Brazil, 18–21 June 2012.
18. He, D.; Wang, D. Robust Biometrics-Based Authentication Scheme for Multiserver Environment. *IEEE Syst. J.* 2014, 9, 816–823. [CrossRef]
19. Odelu, V.; Das, A.K.; Goswami, A. A Secure Biometrics-Based Multi-Server Authentication Protocol Using Smart Cards. *IEEE Trans. Inf. Forensics Secur.* 2015, 10, 1953–1966. [CrossRef]
20. Tsai, J.-L.; Lo, N.-W. A Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services. *IEEE Syst. J.* 2015, 9, 805–815. [CrossRef]
21. He, D.; Kumar, N.; Khan, M.K.; Wang, L.; Shen, J. Efficient Privacy-Aware Authentication Scheme for Mobile Cloud Computing Services. *IEEE Syst. J.* 2016, 12, 1621–1631. [CrossRef]
22. Kumari, S.; Li, X.; Wu, F.; Das, A.K.; Choo, K.-K.R.; Shen, J. Design of a provably secure biometrics-based multi-cloud-server authentication scheme. *Future Gener. Comput. Syst.* 2017, 68, 320–330. [CrossRef]
23. Wu, F.; Xu, L.; Li, X. A New Chaotic Map-Based Authentication and Key Agreement Scheme with User Anonymity for Multi-server Environment. In Proceedings of the International Conference on Frontier Computing, Kuala Lumpur, Malaysia, 3–6 July 2018.
24. Feng, Q.; He, D.; Zeadally, S.; Wang, H. Anonymous biometrics-based authentication scheme with key distribution for multi-server multi-server environment. *Future Gener. Comput. Syst.* 2018, 84, 239–251. [CrossRef]
25. Wang, P.; Zhang, Z.; Wang, D. Revisiting Anonymous Two-Factor Authentication Schemes for Multi-server Environment. In Proceedings of the International Conference on Information and Communications Security, Lille, France, 8 June 2018.
26. Maqbool, S. An efficient hash-based authenticated key agreement scheme for multi-server architecture resilient to key compromise impersonation. *Digit. Commun. Netw.* 2021, 7, 140–150. [CrossRef]
27. Ying, B.; Nayak, A. Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography. *J. Netw. Comput. Appl.* 2019, 131, 66–74. [CrossRef]
28. Kumar, A.; Om, H. An improved and secure multiserver authentication scheme based on biometrics and smartcard. *Digit. Commun. Netw.* 2018, 4, 27–38. [CrossRef]
29. Irshad, A.; Sher, M.; Ahmad, H.F.; Alzahrani, B.A.; Chaudhry, S.A.; Kumar, R. An improved Multi-server Authentication Scheme for Distributed Mobile Cloud Computing Services. *KSII Trans. Internet Inf. Syst.* 2016, 10, 6092–6115. [CrossRef]
30. Chatterjee, S.; Roy, S.; Das, A.K.; Chattopadhyay, S.; Kumar, N.; Vasilakos, A.V. Secure Biometric-Based Authentication Scheme Using Chebyshev Chaotic Map for Multi-Server Environment. *IEEE Trans. Dependable Secur. Comput.* 2018, 15, 824–839. [CrossRef]
31. Zhang, L. Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos Solitons Fractals* 2008, 37, 669–674. [CrossRef]
32. Veyrat-Charvillon, N.; Veyrat-Charvillon, N.; Standaert, F.-X.; Standaert, F.-X. Generic Side-Channel Distinguishers: Improvements and Limitations. In Proceedings of the Annual Cryptology Conference, Santa Barbara, CA, USA, 14–18 August 2011.
33. Das, M.L. Two-factor user authentication in wireless sensor networks. *IEEE Trans. Wirel. Commun.* 2009, 8, 1086–1090. [CrossRef]
34. Wang, D.; Li, W.; Wang, P. Measuring Two-Factor Authentication Schemes for Real-Time Data Access in Industrial Wireless Sensor Networks. *IEEE Trans. Ind. Informatics* 2018, 14, 4081–4092. [CrossRef]
35. Bresson, E.; Chevassut, O.; Pointcheval, D. Security proofs for an efficient password-based key exchange. In Proceedings of the 10th ACM Conference on Computer and Communications Security, Washington, DC, USA, 27–31 October 2003.
36. Borja, M.C.; Haigh, J. The birthday problem. *Significance* 2007, 4, 124–127. [CrossRef]
37. Zhang, L.; Tang, S.; Cai, Z. Cryptanalysis and improvement of password-authenticated key agreement for session initiation protocol using smart cards. *Secur. Commun. Netw.* 2014, 7, 2405–2411. [CrossRef]
38. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. *Proc. R. Soc. Lond. A Math. Phys. Sci.* 1989, 426, 233–271.
39. Lee, T.-F.; Diao, Y.-Y.; Hsieh, Y.-P. A ticket-based multi-server biometric authentication scheme using extended chaotic maps for telecare medical information systems. *Multimedia Tools Appl.* 2019, 78, 31649–31672. [CrossRef]
40. Irshad, A.; Sher, M.; Chaudhary, S.A.; Naqvi, H.; Farash, M.S. An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging Registration Centre. *J. Supercomput.* 2016, 72, 1623–1644. [CrossRef]
41. Braeken, A.; Kumar, P.; Liyanage, M.; Hue, T.T.K. An efficient anonymous authentication protocol in multiple server communication networks (EAAM). *J. Supercomput.* 2017, 74, 1695–1714. [CrossRef]
42. Shin, S.; Kwon, T. A Lightweight Three-Factor Authentication and Key Agreement Scheme in Wireless Sensor Networks for Smart Homes. *Sensors* 2019, 19, 2012. [CrossRef]
43. He, D.; Kumar, N.; Lee, J.-H.; Sherratt, R.S. Enhanced three-factor security protocol for consumer USB mass storage devices. *IEEE Trans. Consum. Electron.* 2014, 60, 30–37. [CrossRef]