A Provable One-way Authentication Key Agreement Scheme with User Anonymity for Multi-server Environment

Hongfeng Zhu
Software College, Shenyang Normal University
No.253, HuangHe Bei Street, HuangGu District, Shenyang, P.C 110034 - China
zhuhongfeng1978@163.com

Received August 20, 2014; revised November 21, 2014; accepted December 16, 2014; published February 28, 2015

Abstract

One-way authenticated key agreement protocols, aiming at solving the problems to establish secure communications over public insecure networks, can achieve one-way authentication of communicating entities for giving a specific user strong anonymity and confidentiality of transmitted data. Public Key Infrastructure can design one-way authenticated key agreement protocols, but it will consume a large amount of computation. Because one-way authenticated key agreement protocols mainly concern on authentication and key agreement, we adopt multi-server architecture to realize these goals. About multi-server architecture, which allow the user to register at the registration center (***RC***) once and can access all the permitted services provided by the eligible servers. The combination of above-mentioned ideas can lead to a high-practical scheme in the universal client/server architecture. Based on these motivations, the paper firstly proposed a new one-way authenticated key agreement scheme based on multi-server architecture. Compared with the related literatures recently, our proposed scheme can not only own high efficiency and unique functionality, but is also robust to various attacks and achieves perfect forward secrecy. Finally, we give the security proof and the efficiency analysis of our proposed scheme.

Keywords: One-way authentication, Key agreement, Multi-server architecture, Anonymity, Chaotic maps

This research was supported by Liaoning Provincial Natural Science Foundation of China (Grant No. 20102202, 201102201) and Liaoning Baiqianwan Talents Program (Grant No.2011921046).

http://dx.doi.org/10.3837/tiis.2015.02.19 **ISSN : 1976-7277**

1. Introduction

Authenticated key exchange (AKE) is one of the most important cryptographic components which is used for establishing an authenticated and confidential communication channel. Based on the number of participants, we can divide AKE protocols into three categories: two-party AKE protocols [1-3], three-party AKE protocols [4], and *N*-party AKE protocols [5-7]. Furthermore, based on the respective features in detail, the previous AKE protocols [3–17] can be classified many categories, we use two-party AKE protocols to set an example: such as using smart card [1], password-based [1–2], chaotic map-based [3], ID-based [7], anonymity [4, 8], secret sharing [9] and so on. Recently many researchers achieve AKE in the multi-server environment called multi-server authenticated key agreement (MSAKA) protocols. MSAKA protocols allow the user to register at the registration center (RC) once and can access all the permitted services provided by the eligible servers. In other words, users do not need to register at numerous servers repeatedly. MSAKA protocols mainly want to solve the problems in a traditional single server with authentication schemes [10] which lead to the fact that user has to register to different servers separately. On a macro level MSAKA protocols can be divided into three phases in chronological order: ***Creative Phase***: The pioneer work in the field was proposed by Li et al. [11] in 2001. However, Lin et al. [12] pointed out that Li et al.'s scheme takes long time to train neural networks and an improved scheme based on ElGamal digital signature and geometric properties on the Euclidean plane has also been given. ***Development Phase***: the main work in this phase is amended repeatedly. For example, Tsai [13] also proposed an efficient multi-server authentication scheme based on one-way hash function without a verification table. Because Tsai's scheme only uses the nonce and one-way hash function, the problems associated with the cost of computation can be avoided in the distributed network environment. However, the literature [14] pointed out that Tsai's scheme is also vulnerable to server spoofing attacks by an insider server and privileged insider attacks, and does not provide forward secrecy. ***Diversification Phase***: the research emphasis shifts to functionality. Therefore, identity-based MSAKA protocols, based on bilinear pairings or elliptic curve cryptosystem (ECC) MSAKA protocols, dynamic identity-based MSAKA protocols and other MSAKA protocols came up recently [14]. **Fig. 1** shows the history of AKE protocols development.

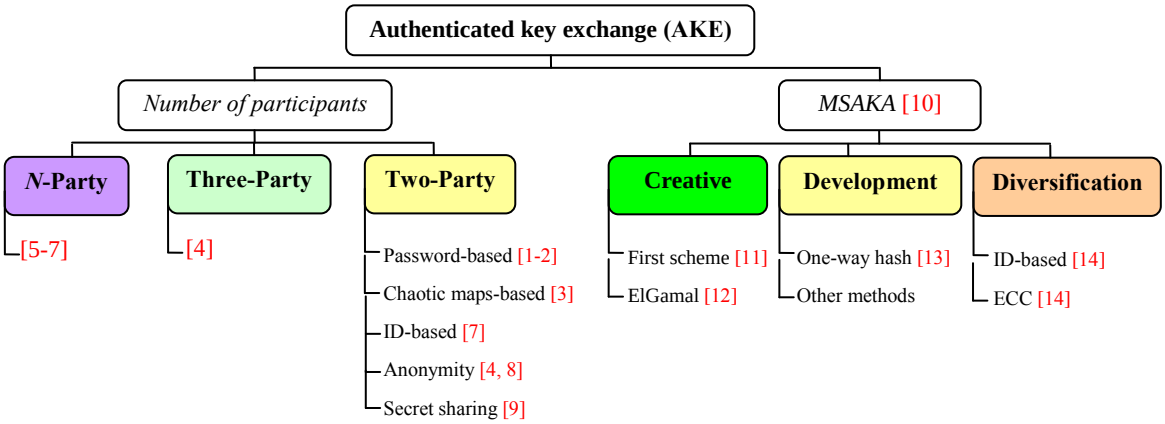

Fig. 1. History of AKE protocols development

However, most existing AKE or MSAKA protocols have emphasized mutual authentication, in which both parties authenticate themselves to their peer. There are many scenes need not

mutual authentication at all and we just need one-way authentication. We can take some facts as examples which are shown in the **Fig. 2**. (1) Readers-to-journalists model: Readers act upon the perceived reputation of a news source, so reputation is a valuable commodity for journalists. No further authentication is required and since the information is public, channel secrecy is not required and does not affect the actions of either party. (2) Patient-to-expert model: On Internet, patients requiring medical advice may wish to do so anonymously, while still ensuring the confidentiality of their request and assurance that the medical advice received comes from an authentic, qualified source.

Fig. 2. No need for mutual authentication environment on Internet

The key idea of one-way AKE is that one party wishes for no one to be able to determine his/her identity, including all the authorities. However, only a few protocols have considered the problem of one-way authentication. Goldberg [15] gave a specialized one-way AKE security definition for the Tor[1] authentication protocol. The literature [16] described an identity-based anonymous authenticated key exchange protocol but with a limited session key secrecy definition based on key recovery, not indistinguishability. Morrissey et al. [17] analyzed the security of the Transport Layer Security (TLS) protocol in the context of one-way authentication, but with specialized security definitions. Recently, Goldberg and Stebila [18] provided an intuitive set of goals and present a formal model that captures these goals. Usually, public key encryption can be used for one-way AKE protocols, for example by having the client encrypt a session key under the server's public key. This mechanism is widely used, for example in the RSA-based cipher suites in TLS [19] and in the KAS1 protocol in NIST SP800-56B [20].

The main contributions are shown as below: The paper firstly presents a new one-way authentication key agreement scheme towards multi-server architecture. Furthermore, the proposed protocol is based on chaotic maps without using modular exponentiation and scalar multiplication on an elliptic curve. In Security aspect, the protocol can resist all common attacks, such as impersonation attacks, man-in-the-middle attacks, etc. About functionality, the protocol also has achieved some well-known properties, such as perfect forward secrecy and execution efficiency.

The rest of the paper is organized as follows: Some preliminaries are given in Section 2. Next, a one-way AKE towards multi-server architecture is described in Section 3. Then, the security analysis and efficiency analysis are given in Section 4 and Section 5. This paper is finally concluded in Section 6.

2. Preliminaries

2.1 Multi-server architecture

Fig. 3 shows the multi-server environment. In the multi-server environment [11], each user

[1] The Onion Router, an anonymous Internet communicaton system.

must perform authentication procedure to login the server for a transaction. If the user is in a single authentication architecture, then the user must register at various servers and memorize the corresponding identifications and passwords, which could not be convenient for a user. In order to make the registration to various servers easier for users, each user must register with the registration center to obtain a secure account. Then the user uses the secure account to perform the login and authentication procedures with various servers.

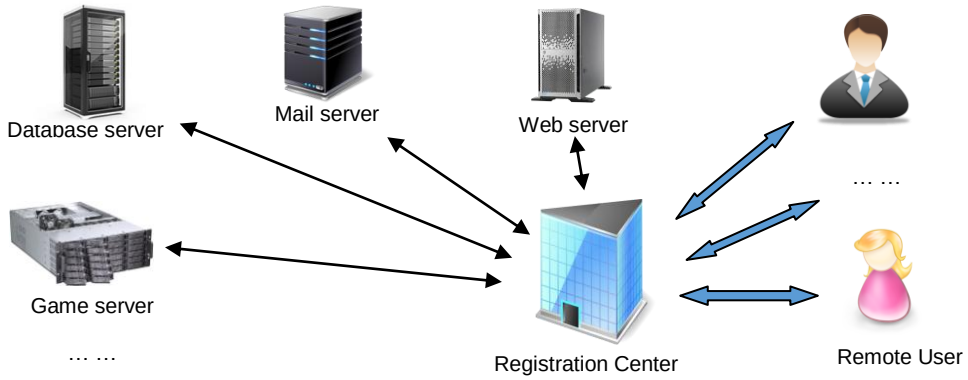

Fig. 3. The multi-server communication architecture

2.2 Security requirements

Secure communication schemes for remote one-way authentication and session key agreement for the multi-server architecture should provide security requirements [24]:

(1) One-way authentication and key agreement: One-way authentication and key agreement refers to only one party authenticating the other suitably and getting the session key simultaneously.

(2) Impersonation attack: An impersonation attack is an attack in which an adversary successfully assumes the identity of one of the legitimate parties in a system or in a communications protocol.

(3) Man-in-the-middle attack: The man-in-the-middle attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

(4) Replay attack: A replay attack is a form of network attack in which a valid data transmission is repeated or delayed maliciously or fraudulently.

(5) Known-key security: Known-key security is that a protocol can protect the subsequent session keys from disclosing even if the previous session keys are revealed by the intendant user.

(6) Perfect forward secrecy: An authenticated key establishment protocol provides perfect forward secrecy if the compromise of both of the node's secret keys cannot results in the compromise of previously established session keys.

(7) Session key security: A communication protocol exhibits session key security if the session key cannot be obtained without any long-term secrets.

(8) Resistance to stolen-verifier attacks: An adversary gets the verifier table from servers or RC by a hacking way, and then the adversary can launch any other attack which called stolen-verifier attacks.

(9) No verification table: there is no verification table at the RC or the server at all.

(10) Securely chosen password and time synchronization: Guarantee securely chosen password and no need for time synchronization among parties.

(11) Authentication: one way authentication or mutual authentication in different phase in our protocol.

3. The Proposed One-Way AKE towards Multi-Server Architecture

In this section, under the multi-server architecture, a chaotic maps-based one-way authentication key agreement scheme is proposed which consists of two phases: the servers registration phase, one-way authentication key agreement phase.

Remark 1: Because our proposed protocol is a one-way authentication scheme, there is no password update phase for users in our protocol.

3.1 Notations and Chebyshev chaotic maps

In this section, any server *i* has its identity ID_{S_i}. Only *RC* has its identity ID_{RC} and public key $(x, T_k(x))$ and a secret key k based on Chebyshev chaotic maps, a secure one-way hash function $H(\cdot)$, and a pair of secure symmetric encryption/decryption functions $E_K() / D_K()$ with key *K*. The concrete notations used hereafter are shown in **Table 1**.

Table 1. Notations

Symbol	Definition
SID_A	a temporary session
S_i, ID_{S_i}	The *i*th server， the identity of the *i*th server, respectively
a, r_i	nonces
$(x, T_k(x))$	public key based on Chebyshev chaotic maps
k	secret key based on Chebyshev chaotic maps
RC, ID_{RC}	registration center and its identity
$E_K() / D_K()$	a pair of secure symmetric encryption/decryption functions with the key *K*
H	A secure one-way hash function
$\|\|$	concatenation operation

Let n be an integer and let x be a variable with the interval $[-1,1]$. The Chebyshev polynomial [21] $T_n(x) : [-1,1] \to [-1,1]$ is defined as $T_n(x) = \cos(n \cos^{-1}(x))$. Chebyshev polynomial map $T_n : R \to R$ of degree n is defined using the following recurrent relation:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad (1)$$

where $n \geq 2$, $T_0(x) = 1$, and $T_1(x) = x$.

The first few Chebyshev polynomials are:

$T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3x$, $T_4(x) = 8x^4 - 8x^2 + 1$, … …

One of the most important properties is that Chebyshev polynomials are the so-called semi-group property which establishes that

$$T_r(T_s(x)) = T_{rs}(x). \quad (2)$$

An immediate consequence of this property is that Chebyshev polynomials commute under composition

$$T_r(T_s(x)) = T_s(T_r(x)) . \tag{3}$$

In order to enhance the security, Zhang [22] proved that semi-group property holds for Chebyshev polynomials defined on interval (-∞,+∞). The enhanced Chebyshev polynomials are used in the proposed protocol:

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) (\mathrm{mod}\, N), \tag{4}$$

where $n \geq 2$, $x \in (-\infty, +\infty)$, and N is a large prime number. Obviously,

$$T_{rs}(x) = T_r(T_s(x)) = T_s(T_r(x)) . \tag{5}$$

Definition 3.1. Semi-group property of Chebyshev polynomials:

$$T_{rs}(x) = T_r(T_s(x)) = \cos(r\cos^{-1}(s\cos^{-1}(x))) = \cos(rs\cos^{-1}(x)) = T_s(T_r(x)) = T_{sr}(x) .$$

Definition 3.2. Given *x* and *y*, it is intractable to find the integer *s*, such that $T_s(x) = y$. It is called the Chaotic Maps-Based Discrete Logarithm problem (CMBDLP).

Definition 3.3. Given *x*, $T_r(x)$ and $T_s(x)$, it is intractable to find $T_{rs}(x)$. It is called the Chaotic Maps-Based Diffie-Hellman problem (CMBDHP).

3.2 Servers registration phase

Concerning the fact that the proposed scheme mainly relies on the design of Chebyshev chaotic maps-based in multi-server architecture, it is assumed that the servers can register at the registration center in some secure way or by secure channel. The same assumption can be set up for servers. **Fig. 4** illustrates the server registration phase.

Step 1. When a server(or an expert) wants to be a new legal service provider, she chooses her identity ID_{S_i} with her identification card in law. Then the server submits ID_{S_i} to the ***RC*** via a secure channel.

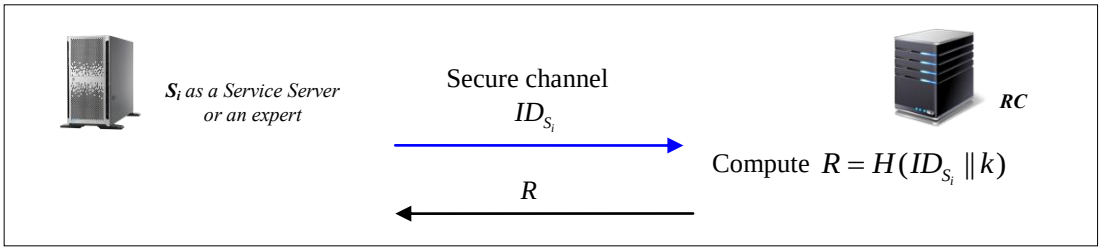

Fig. 4. Server or a authenticated expert registration phase

Step 2. Upon receiving ID_{S_i} from the server, the ***RC*** computes $R = H(ID_{S_i} \| k)$, where k is the secret key of ***RC***. Then the server stores *R* in a secure way via a secure channel.

3.3 One-way authenticated key agreement phase

In this phase, one-way authenticated means that the server or the ***RC*** can be authenticated by the other two peers, but the user can not be authenticated by the the server or the ***RC*** to keep the user complete anonymity in the multi-server architecture. This concrete process is presented in the following **Fig. 5**.

Step 1. If Alice(assume Alice as an anonymous user) wishes to consult some personal issues establish with S_i(or an expert) in a secure way, she will choose a random integer

number a and a temporary session SID_A. Then the device of Alice will compute $K_{A-RC} = T_a T_k(x)$, $H_A = H(SID_A \| ID_{S_i} \| T_a(x))$, $C_1 = E_{K_{A-RC}}(SID_A \| ID_{S_i} \| H_A)$. After that, Alice sends $m_1 = \{SID_A, T_a(x), C_1\}$ to S_i where she wants to get the server's service.

Step 2. After receiving the message $m_1 = \{SID_A, T_a(x), C_1\}$ from Alice, S_i will do the following tasks to ask ***RC*** for helping Alice to authenticate itself: S_i selects random r_i and computes $T_{r_i}(x)$ and $C_2 = H(ID_{S_i} \| m_1 \| R \| T_{r_i}(x))$. And then sends the message m_2 to ***RC***.

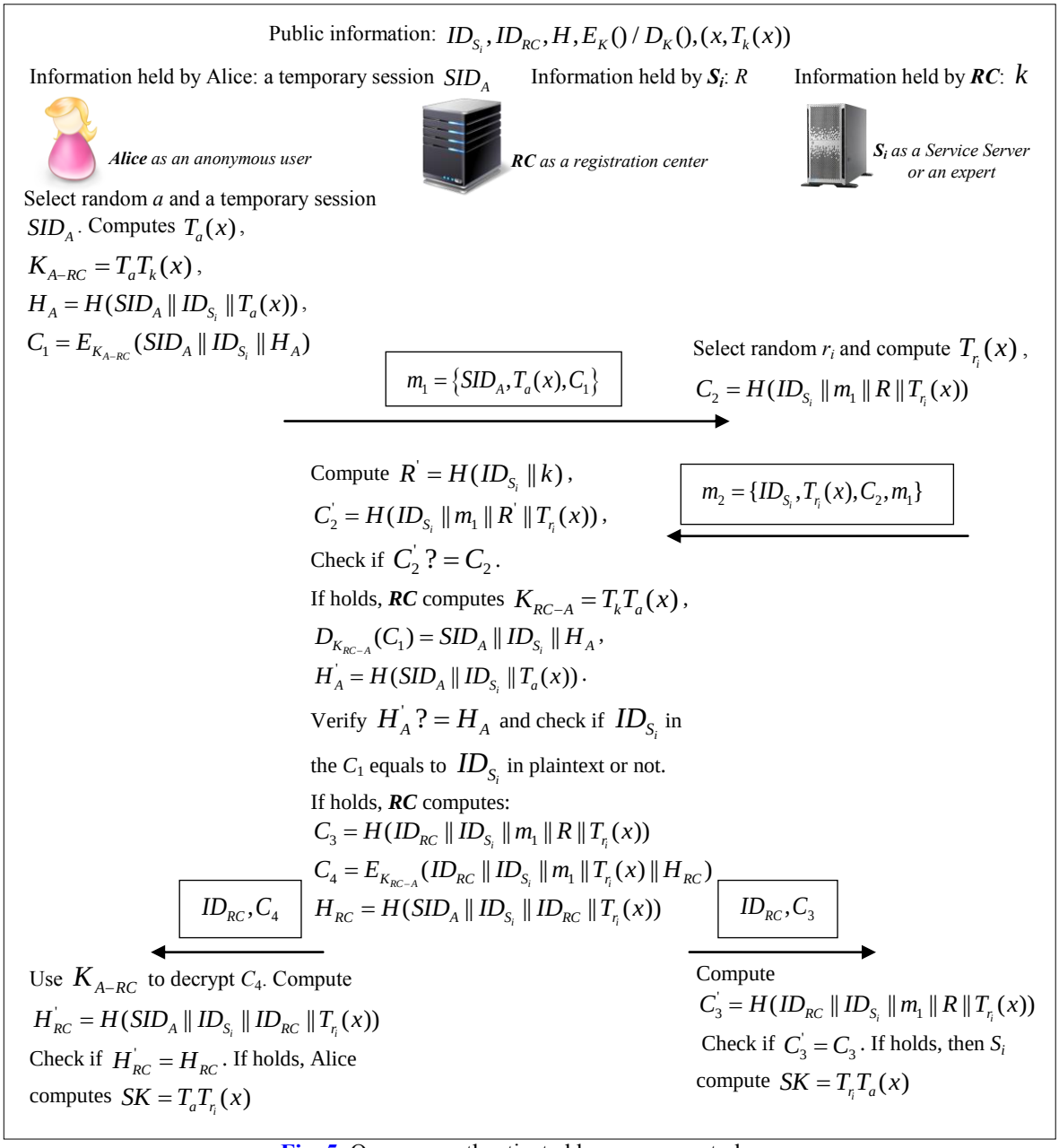

Fig. 5. One-way authenticated key agreement phase

Step 3. Next, ***RC*** will help Alice to authenticate S_i and verify the temporary information by helping them to compute the session key. After receiving the message $m_2 = \{ID_{S_i}, T_{r_i}(x), C_2, m_1\}$, ***RC*** will do the following tasks:

(1) Authenticate S_i: Based on ID_{S_i}, ***RC*** can compute $R^{'} = H(ID_{S_i} \| k)$. Then ***RC*** computes $C_2^{'} = H(ID_{S_i} \| m_1 \| R^{'} \| T_{r_i}(x))$ and check if $C_2^{'} ? = C_2$. If above equation holds, that means S_i are legal participants in this instance because only S_i own R.

(2) Confirm S_i is the server that Alice wants to consult with: ***RC*** computes $K_{RC-A} = T_k T_a(x)$ and then decrypts C_1 to get $SID_A \| ID_{S_i} \| H_A$. Next, ***RC*** computes $H_A^{'} = H(SID_A \| ID_{S_i} \| T_a(x))$. ***RC*** verifies $H_A^{'} ? = H_A$ and checks if ID_{S_i} in the C_1 equals to ID_{S_i} in plaintext or not. If holds, that means S_i is the server that Alice wants to consult with.

(3) Help S_i and Alice to get the session key: ***RC*** computes $C_3 = H(ID_{RC} \| ID_{S_i} \| m_1 \| R \| T_{r_i}(x))$, $C_4 = E_{K_{RC-A}}(ID_{RC} \| ID_{S_i} \| m_1 \| T_{r_i}(x) \| H_{RC})$ and $H_{RC} = H(SID_A \| ID_{S_i} \| ID_{RC} \| T_{r_i}(x))$. Then ***RC*** sends the message ID_{RC}, C_4 to Alice and sends the message ID_{RC}, C_3 to S_i.

If any authenticated process does not pass, the protocol will be terminated immediately.

Step 4. For Alice: After receiving the message ID_{RC}, C_4, Alice uses K_{A-RC} to decrypt C_4. Next Alice computes $H_{RC}^{'} = H(SID_A \| ID_{S_i} \| ID_{RC} \| T_{r_i}(x))$. Check if $H_{RC}^{'} = H_{RC}$. If holds, Alice computes $SK = T_a T_{r_i}(x)$. For S_i: After receiving the message ID_{RC}, C_3, S_i computes $C_3^{'} = H(ID_{RC} \| ID_{S_i} \| m_1 \| R \| T_{r_i}(x))$ and checks if $C_3^{'} = C_3$. If holds, then S_i computes $SK = T_{r_i} T_a(x)$.

Remark 2: We can view the servers and RC as an integrated system for the user, so from the perspective of the user, we adopt one-way authentication, that means only user authenticated the integrated system (the server and RC) but there is no authentication for the user. However, from the inside integrated system, for providing the reliable service in multi-server architecture, and we must make the server and RC to authenticate each other, that is the mutual authentication.

4. Security Analysis

The section analyzes the security of our proposed protocol. Let us assume that there are three secure components, including the two problems CMBDLP and CMBDHP cannot be solved in polynomial-time, a secure one-way hash function, and a secure symmetric encryption. Assume that the adversary has full control over the insecure channel including eavesdropping, recording, intercepting, modifying the transmitted messages.

4.1 Security proof of the proposed scheme

(1) One-way authentication and key agreement

Definition 4.1. One-way authentication and key agreement refers to only one party authenticating the other suitably and getting the session key simultaneously.

Theorem 4.1. *The proposed protocol can achieve one-way authentication and key*

agreement.

Proof: In our proposed protocol, one-way authentication means that ***RC*** helps Alice (an anonymous user) to authenticate S_i. So we can divide the one-way authentication process into three steps:

(a) Alice authenticates ***RC***: Because only ***RC*** has the secret k, ***RC*** can computes $K_{RC-A} = T_k T_a(x)$ which equals to $K_{A-RC} = T_a T_k(x)$. So if Alice decrypts C_4 to get the necessary information and check if $H'_{RC} = H_{RC}$. If above equation is equal, then that means Alice authenticates ***RC***.

(b) ***RC*** and S_i authenticate each other: We can use the shared key R to achieve the task. Firstly, based on ID_{S_i}, ***RC*** can compute $R' = H(ID_{S_i} \| k)$ by its private key k. Then ***RC*** computes $C'_2 = H(ID_{S_i} \| m_1 \| R' \| T_{r_i}(x))$ and checks if $C'_2 = C_2$. If above equation is equal, then that means ***RC*** authenticates S_i. After receiving the messages $\{ID_{RC}, C_3\}$, S_i computes $C'_3 = H(ID_{RC} \| ID_{S_i} \| m_1 \| R \| T_{r_i}(x))$ and chesks if $C'_3 = C_3$. If holds, we can say S_i authenticates ***RC***.

(c) Alice authenticates S_i: If Alice already authenticates ***RC***, then she can authenticate S_i based on the information $ID_{RC} \| ID_{S_i} \| m_1 \| T_{r_i}(x) \| H_{RC}$ which were decrypted by ***RC*** in C_4. The trust flow is $\text{Alice} \rightarrow RC \rightarrow S_i$.

As for the key agreement, after authenticating each other, the temporary $T_a(x)$, $T_{r_i}(x)$ and the $SID_A \| ID_{S_i} \| ID_{RC}$ were already authenticated by ***RC***. So finally Alice and S_i can make the key agreement simultaneously.

(2) Impersonation attack

Definition 4.2. An impersonation attack is an attack in which an adversary successfully assumes the identity of one of the legitimate parties in a system or in a communications protocol.

Theorem 4.2. *The proposed protocol can resist impersonation attack.*

Proof: An adversary cannot impersonate anyone of the S_i and ***RC***. The proposed scheme has already authenticated each other between S_i and ***RC***, and Alice authenticates S_i and ***RC*** (in section **4.1.(1)**) based on the secrets k, R and the nonces a, r_i. So there is no way for an adversary to have a chance to carry out impersonation attack.

Remark 3: Because Alice is an anonymous user, an adversary impersonates "Alice" is meaningless for the S_i and ***RC***.

(3) Man-in-the-middle attack

Definition 4.3. The man-in-the-middle attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

Theorem 4.3. *The proposed protocol can resist Man-in-the-middle attack.*

Proof: Because $C_i (1 \le i \le 4)$ contain the participants' identities or an anonymous user's temporary session ID, a man-in-the-middle attack cannot succeed.

(4) Replay attack

Definition 4.4. A replay attack is a form of network attack in which a valid data transmission is repeated or delayed maliciously or fraudulently.

Theorem 4.4. *The proposed protocol can resist replay attack.*

Proof: That any message of Alice was replayed by an adversary is meaningless. Because "Alice" is an anonymous user, the adversary can as an anonymous user to initiate the protocol legally as his wish.

For the messages between S_i and ***RC***, an adversary cannot start a replay attack against our scheme because there were the fresh nonces a, r_i in each session. If $T_a(x)$ and $T_{r_i}(x)$ have appeared before or the status shows in process, any of the participants in instance protocol will reject the session request. If the adversary wants to launch the replay attack successfully, it must compute and modify $T_a(x)$, $T_{r_i}(x)$ and $C_i(1 \le i \le 4)$ correctly which is impossible.

(5) Known-key security

Definition 4.5. Known-key security is that a protocol can protect the subsequent session keys from disclosing even if the previous session keys are revealed by the intendant user.

Theorem 4.5. *The proposed protocol can achieve known-key security.*

Proof: Since the session key $SK = T_a T_{r_i}(x) = T_{r_i} T_a(x)$ is depended on the random nonces a and r_i, and the generation of nonces is independent in all sessions, an adversary cannot compute the previous and the future session keys when the adversary knows one session key. And in the secrets update phase, any session key is only used once, so it has known-key security attribute.

(6) Perfect forward secrecy

Definition 4.6. An authenticated key establishment protocol provides perfect forward secrecy if the compromise of both of the node's secret keys cannot results in the compromise of previously established session keys.

Theorem 4.6. *The proposed protocol can achieve perfect forward secrecy.*

Proof: In the proposed scheme, the session key $SK = T_a T_{r_i}(x) = T_{r_i} T_a(x)$ is related with a and r_i, which were randomly chosen by Alice and the server S_i respectively. So any session key has not related with the secret key (such as k) of each of participants. Furthermore, because of the intractability of the CMBDLP and CMBDHP problem, an adversary cannot compute the previously established session keys.

(7) Session key security

Definition 4.7. A communication protocol exhibits session key security if the session key cannot be obtained without any long-term secrets.

Theorem 4.7. *The proposed protocol can achieve session key security.*

Proof: In the authenticated key agreement phase, a session key SK is generated from a and r_i. These parameter values are different in each session, and each of them is only known by Alice and S_i. Whenever the communication ends between S_i and Alice, the key will immediately self-destruct and will not be reused. Therefore, assuming the attacker has obtained a session key, and Alice will be unable to use this session key to decode the information in other communication processes. Because the random point elements a and r_i

are all generated randomly and are protected by the CMBDLP, CMBDHP, and the secure symmetric encryption, a known session key cannot be used to calculate the value of the next session key. Additionally, since the values a and r_i of the random elements are very large, attackers cannot directly guess the values a and r_i of the random elements to generate session key. Therefore, the proposed scheme provides session key security.

(8) Resistance to stolen-verifier attacks

Definition 4.8. An adversary gets the verifier table from servers or ***RC*** by a hacking way, and then the adversary can launch any other attack which called stolen-verifier attacks.

Theorem 4.8. *The proposed protocol can resistance to stolen-verifier attacks.*

Proof: In the proposed scheme, neither the server nor the registration center maintains any verification table. Thus, the stolen-verifier attack is impossible to initiate in the proposed scheme.

Remark 4: One-way authentication AKE means that there is no verification table at the ***RC*** or the server at all. And securely chosen password is also no need for our protocol. Because our protocol is based on nonces, there is no need for time synchronization.

From the **Table 2**, we can see that the proposed scheme can provide secure session key agreement, perfect forward secrecy and so on. As a result, the proposed scheme is more secure and has much functionality compared with the recent related scheme.

Table 2. Security of our proposed protocol

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
[26](2013)	Yes	Mutual	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
[27](2008)	Yes	Mutual	Yes	Yes	Yes	No	No	No	No	No	Yes	No
[28](2009)	Yes	No	Yes	Yes	Yes	ID hiding	No	No	No	Yes	Yes	No
[29](2009)	Yes	No	Yes	No	Yes	ID hiding	No	No	No	Yes	Yes	No
Our Scheme	Yes (Only Servers)	One-way for users, mutual for server and ***RC***	Yes	NN	Yes	Anonymity	Yes	Yes	Yes	Yes	Yes	Yes

S1: Single registration; **S2**: Authentication; **S3**: No verification table; **S4**: Securely chosen password;
S5: Session key agreement; **S6**: Privacy protection for a user; **S7**: Freedom from time synchronization;
S8: Session key secrecy; **S9**: Perfect forward secrecy; **S10**: Resistance to replay attack;
S11: Resistance to stolen-verifier attack; **S12**: Resistance to masquerading attack
Yes/No: Support/Not support the security **NN**: No need

Remark 5: Some qualitatively discuss about the difference between the proposed scheme and [26-29] as followed: (1) Our protocol is one way authentication AKE for users, so only servers need to registration at the ***RC***. (2) About authentication, one-way authentication for users and mutual authentication for server and ***RC*** (see **Remark 2**). (3) Our proposed protocol can hold the security **S1-S12**, but the [26-29] have some defects. (4) Our protocol is anonymity, and [28-29] only assure ID hiding, and [26-27] have no privacy protect at all. The main differences about anonymity and ID hiding, and please see the Appendix.

4.2 The provable security of the proposed scheme

We recall the definition of session-key security in the authenticated-links adversarial model of Canetti and Krawczyk [25]. The basic descriptions are shown in **Table 3**.

Table 3. Descriptions the model of Canetti and Krawczyk

Symbol	Definition
parties $P_1,...P_n$	Modeled by probabilistic Turing machines.
Adversary Λ	A probabilistic Turing machine which controls all communication, with the

	exception that the adversary cannot inject or modify messages (except for messages from corrupted parties or sessions), and any message may be delivered at most once.
Send query	The adversary can control over Parties' outgoing messages via the **Send** query. Parties can be activated by the adversary launching **Send** queries.
Two sessiosn matching	If the outgoing messages of one are the incoming messages of the other

We allow the adversary access to the queries **SessionStateReveal**, **SessionKeyReveal**, and **Corrupt**.

(1) **SessionStateReveal(*s*)**: This query allows the adversary to obtain the contents of the session state, including any secret information. ***s*** means no further output.

(2) **SessionKeyReveal(*s*)**: This query enables the adversary to obtain the session key for the specified session ***s***, so long as ***s*** holds a session key.

(3) **Corrupt(P_i)** : This query allows the adversary to take over the party P_i, including long-lived keys and any session-specific information in P_i's memory. A corrupted party produces no further output.

(4) **Test(*s*)**: This query allows the adversary to be issued at any stage to a completed, fresh, unexpired session ***s***. A bit *b* is then picked randomly. If $b = 0$, the test oracle reveals the session key, and if $b = 1$, it generates a random value in the key space. The adversary Λ can then continue to issue queries as desired, with the exception that it cannot expose the test session. At any point, the adversary can try to guess *b*. Let $GoodGuess^{\Lambda}(k)$ be the event that the adversary Λ correctly guesses *b*, and we define the advantage of adversary Λ as $Advantage^{\Lambda}(k) = \max\{0, |\Pr[GoodGuess^{\Lambda}(k)] - \frac{1}{2}|\}$, where *k* is a security parameter.

A session ***s*** is locally exposed with P_i: if the adversary had issued **SessionStateReveal(*s*)**, **SessionKeyReveal(*s*)**, **Corrupt(P_i)** before ***s*** would have expired.

Definition 4.9. An authenticator exchange protocol Π_1 in security parameter *k* is said to be authentication secure in the adversarial model of Canetti and Krawczyk if for any polynomial-time adversary Λ,

Algorithm 1 CMBDHP distinguisher

Input: $H, E_K() / D_K(), (x, T_k(x))$

1: $r \xleftarrow{R} \{1, ..., k\}$, where k is an upper bound on the number of sessions activated by Λ in any interaction.

2: Invoke Λ and simulate the protocol to Λ, except for the $r-th$ activated protocol session.

3: For the $r-th$ session, let a user send $\{i, SID_A, T_a(x), C_1\}$ to a server $\boldsymbol{S_i}$, and let the server $\boldsymbol{S_i}$ send $\{i, ID_{S_i}, T_{r_i}(x), C_2, m_1\}$ to the ***RC***, where i is the session identifier. The ***RC*** can compute the encrypted messages $\{C_3, C_4\}$ with the authenticators locally after authenticating the server $\boldsymbol{S_i}$ by one-round messages and public information.

4: **if** the $r-th$ session is chosen by Λ as the test session **then**

5: Provide Λ as the answer to the test query.

6: $d \leftarrow \Lambda$'s output.

7:**else** $d \xleftarrow{R} \{0,1\}$.

8:**end if**

Output: d

(1) If two uncorrupted parties have completed matching sessions, these sessions produce the same key as output;

(2) $Advantage^{\Lambda}(k)$ is negligible.

Theorem 4.9. *Under the CMBDHP assumption, using the Algorithm* 1 *to compute* an *authenticator is authentication secure in the adversarial model of Canetti and Krawczyk* [25].

Proof. The proof is based on the proof given by Refs. [25]. There are two uncorrupted parties in matching sessions output the same session key, and thus the first part of **Definition 4.9.** is satisfied. To show that the second part of the definition is satisfied, assume that there is a polynomial-time adversary Λ with a non-negligible advantage ε in standard model. We claim that Algorithm 1 forms a polynomial-time distinguisher for CMBDHP having non-negligible advantage.

Probability analysis. It is clear that Algorithm 1 runs in polynomial time and has non-negligible advantage. There are two cases where the *r*-th session is chosen by Λ as the test session: (1) If the *r*-th session is not the test session, then Algorithm 1 outputs a random bit, and thus its advantage in solving the CMBDHP is 0. (2) If the *r*-th session is the test session, then Λ will succeed with advantage ε, since the simulated protocol provided to Λ is indistinguishable from the real protocol. The latter case occurs with probability $1/k$, so the overall advantage of the CMBDHP distinguisher is ε / k, which is non-negligible.

Definition 4.10. A composable key exchange protocol Π_2 in security parameter *k* is said to be session-key secure in the adversarial model of Canetti and Krawczyk if for any polynomial-time adversary Λ,

Algorithm 2 Proposed protocol simulator

Input: $H, E_K()/D_K(), (x, T_a(x)), (x, T_{r_i}(x), (x, T_k(x))$

1: $r \xleftarrow{R} \{1,...,k\}$, where k is an upper bound on the number of sessions activated by Λ in any interaction.

2: Invoke Λ and simulate the protocol to Λ, except for the $r-th$ activated protocol session.

3: For the $r-th$ session, After running the protocol Π_1, the ***RC*** can compute the encrypted messages $\{C_3, C_4\}$ with the authenticators locally. Then the ***RC*** continues to send messages $\{ID_{RC}, C_4\}$ and $\{ID_{RC}, C_3\}$ to the user and the server $\boldsymbol{S_i}$ respectively. Both the user and the server can compute the session key $SK = H(T_a T_{r_i}(x))$ locally after authenticating each other by ***RC***'s messages and public information.

4: **if** the $r-th$ session is chosen by Λ as the test session **then**

5: Provide Λ as the answer to the test query.

6: $d \leftarrow \Lambda$'s output.

7:**else** $d \xleftarrow{R} \{0,1\}$.

8:**end if**

Output: d

(3) If two uncorrupted parties have completed matching sessions with pre-distributed parameter, these sessions produce the same key as output;

(4) $Advantage^{\Lambda}(k)$ is negligible.

Theorem 4.10. *Under the CMBDHP assumption, using the Algorithm* 2 *to compute session key is session-key secure in the adversarial model of Canetti and Krawczyk* [25].

Proof. The proof's process is similar to **Theorem 4.9**. The protocol Π_2 is the composable instance of protocol multiple Π_1. Since **Theorem 4.9** is session-key secure, the protocol Π_2 is also session-key secure.

Probability analysis. It is similar to Algorithm 1. If we assume that Algorithm 2 forms a polynomial-time distinguisher for CMBDHP having non-negligible advantage, the overall advantage of the proposed protocol simulator with authenticated parameter is ε / k which is also non-negligible. Because the protocol Π_2 chooses different parameters to structure session keys in different phase which are secure independence of protocol Π_1.

5. Efficiency Analysis

Compared to RSA and ECC, Chebyshev polynomial computation problem offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings. In our proposed protocol, no time-consuming modular exponentiation and scalar multiplication on elliptic curves are needed. However, Wang [21] proposed several methods to solve the Chebyshev polynomial computation problem. For convenience, some notations are defined as follows.

- T_{hash}: The time for executing the hash function;
- T_{sym} : The time for executing the symmetric key cryptography;
- T_{XOR}: The time for executing the XOR operation;
- T_{Exp}: The time for a modular exponentiation computation;
- T_{CH}: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial using the algorithm in literature[30].

To be more precise, on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, where n and p are 1024 bits long, the computational time of a one-way hashing operation, a symmetric encryption/decryption operation, an elliptic curve point multiplication operation and Chebyshev polynomial operation is 0.0005s, 0.0087s, 0.063075s and 0.02102s separately [30]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations.

Table 4 shows performance comparisons between our proposed scheme and the literature of [26-29] in multi-server architecture. Therefore, as in **Table 4** the concrete comparison data as follows:

(1) The concrete computation cost. Based on the [21] [30], the conclusions can be drawn with different phase as follows:

Phase A: The computation cost of our proposed protocol is zero. And the computation cost of the literatures [26-29] is about 0.001s, 0.001s, 0.0025s, and 0.004s respectively.

Phase B: The computation cost of our proposed protocol is the same as all the related literature [26-29] which is about 0.0005s.

Phase C: The computation cost of our proposed protocol is about $1T_{sym} + 1T_{hash} + 1T_{CH} \approx$ 0.03022s. And the computation cost of the literatures [26-29] is about 0.064075s, 0.0005s, 0.0015s, and 0.0035s respectively.

Phase D: The computation cost of our proposed protocol is about $8T_{hash} + 3T_{CH} + 2T_{sym} \approx$ 0.08443s. And the computation cost of the literatures [26-29] is about 0.193725s, 0.008s, 0.0045s, and 0.0075s respectively.

Phase E: The computation cost of our proposed protocol is zero. And the computation cost of the literatures [26-29] is about 0.001s, 0.001s, 0.002s, and 0.002s respectively.

Total: The computation cost of our proposed protocol is 0.11415s. And the computation cost of the literatures [26-29] is about 0.2603s, 0.011s, 0.011s, and 0.0175s respectively.

(2) The concrete communication cost. The communication round of our proposed protocol is equal to the literature [28], and reduced by about 25%, 57%, and 40% with the literature [26] [27][29] respectively.

(3) Comparisons and Reasons

(a) The total computation cost of our proposed protocol is lower than the literatures [26]. The main reason is that the literatures [26] adopted modular exponentiation computation. At the same time, the literatures [26] cannot provide privacy protection for a user.

(b) The total computation cost of our proposed protocol is higher than the literatures [27-29]. Furthermore, the communication round of our proposed protocol is superior to the literature [27][29] and is equal to the literature [28]. The reasons are: one reason is our protocol mainly adopts Chebyshev chaotic maps but the literatures [27-29] mainly adopts one way hash funciton. At the same time, Chebyshev chaotic maps has more attributes which leading to reduce communication rounds. Furthermore, from the perspective of security, our protocol is more secure than the literatures [27-29]. From the **Table 3**, we can see that the literatures [26-29] cannot resist many attacks and the literatures [28-29] cannot afford any authentication method. Therefore, as in **Table 3** and **Table 4**, we can draw a conclusion that the proposed scheme has achieved the balance of efficiency and security.

Table 4. Efficiency of our proposed scheme

Phase		[26](2013)	[27](2008)	[28](2009)	[29](2009)	Ours
A		$2T_{hash} + 1T_{XOR}$	$2\ T_{hash} + 1T_{XOR}$	$5T_{hash} + 2T_{XOR}$	$8T_{hash} + 4T_{XOR}$	NN
B		$1T_{hash}$	$1\ T_{hash}$	$1T_{hash}$	$1T_{hash}$	$1T_{hash}$
C		$2T_{hash} + 1T_{XOR} + 1T_{Exp}$	$1T_{hash} + 2T_{XOR}$	$6T_{hash} + 3T_{XOR}$	$7T_{hash} + 7T_{XOR}$	$1T_{sym} + 1T_{hash} + 1T_{CH}$
D	User	$1T_{hash} + 1T_{Exp}$	$4T_{hash} + 3T_{XOR}$	$3T_{hash}$	$2T_{hash}$	$1T_{hash} + 1T_{CH}$
	Server	$2T_{hash} + 2T_{Exp}$	$6T_{hash} + 7T_{XOR}$	$6T_{hash} + 3T_{XOR}$	$8T_{hash} + 6T_{XOR}$	$2T_{hash} + 1T_{CH}$
	RC	$6T_{hash}$	$6T_{hash} + 5T_{XOR}$	0	$5T_{hash} + 7T_{XOR}$	$5T_{hash} + 1T_{CH} + 2T_{sym}$
	Total	$9T_{hash} + 3T_{Exp}$	$16T_{hash} + 15T_{XOR}$	$9T_{hash} + 3T_{XOR}$	$15T_{hash} + 13T_{XOR}$	$8T_{hash} + 3T_{CH} + 2T_{sym}$
E		$2T_{hash} + 2T_{XOR}$	$2T_{hash} + 2T_{XOR}$	$4T_{hash} + 5T_{XOR}$	$4T_{hash} + 4T_{XOR}$	NN
F		4 rounds	7 rounds	3 rounds	5 rounds	3 rounds

A: User registration **B**: Server registration **C**: Login phase **NN**: No Need
D: Authentication phase **E**: Password change phase **F**: Communication cost

There are few one-way AKE schemes. The literature [17] was a security analysis of the TLS handshake protocol which has some context was related with one-way AKE protocol, but there was no a concrete protocol process. So this paper choose the literature [18] as a compared paper. For simplify, we just compare with the literature [18] about authentication phase. **Table 5** presents the effciency in term of modular exponentiations(T_{Exp}), chebyshev polynomial(T_{CH}) and symmetric key cryptography(T_{sym}) computation of relevant one-way authentication key agreement protocol [18]. Therefore, from **Table 5** we can see that the our proposed scheme has achieved the tight security and good efficiency. The tight security will be given in **Section 4.2**. Our protocol is more efficient than the literature [18] because chebyshev polynomial and symmetric key cryptography computation is more efficient than modular exponentiations computation, and our protocol chooses the former algorithms. Moreover, our proposed scheme possesses expandability because it is realized in multi-server architecture.

Table 5. Efficiency in terms of modular exponentiations and chebyshev polynomial

Protocol	Efficiency (client)	Efficiency (server)	Efficiency (*RC*)	Authentication	Security Model	Architecture
NTOR[18]	$3T_{Exp}$	$3T_{Exp}$	0	one-way	tight	Two-party
Ours	$1T_{CH}$	$1T_{CH}$	$1T_{CH}$ $+2T_{sym}$	One-way for users, mutual for server and ***RC***	tight	Multi-Server
Tight: The only signicant factor between the difficulty of breaking the key agreement protocol and the difficulty of solving the underlying function is the factor that comes from guessing the correct test session. In brief, an adversary only solves some kind of hard problem, the protocol can be compromised.						

6. Conclusion

This work provides a new approach to one-way authenticated key establishment towards multi-server architecture. The core ideas of the proposed scheme are the mutual authentication between the servers and ***RC*** and the anonymity for the users. Subsequently, we explain the practical motivations for authentication and secrecy assurances of parties engaging in one-way AKE protocols and some related terms. Based on our discussion we proposed a suitable protocol that covers those goals and offered an efficient protocol that formally meets the proposed security definition. Finally, after comparing with related literatures (multi-server schemes and one-way protocols) respectively, we found our proposed scheme has satisfactory security, efficiency and functionality. Therefore, our protocol is more suitable for practical applications.

References

[1] Lamport L, "Password authentication with insecure communication," *Commun ACM*, Vol. 24, no. 11, pp. 770–772, 1981. Article (CrossRef Link).

[2] Jonathan Katz, Philip MacKenzie, Gelareh Taban, Virgil Gligor, "Two-server password-only authenticated key exchange," *Applied Cryptography and Network Security*, Vol. 3531, pp. 1-16, 2005. Article (CrossRef Link).

[3] M. S. Baptista, "Cryptography with chaos," *Physics Letters A,* vol. 240, no. 1, pp. 50-54, 1998. Article (CrossRef Link).

[4] Lee C, Li C, and Hsu C, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps," *Nonlinear Dyn*, Vol. 73, pp. 125–132, 2013. Article (CrossRef Link).

[5] E. Bresson, O. Chevassut and D. Pointcheval, "Group Diffie-Hellman key exchange secure against dictionary attack," *Asiacrypt*, Vol. 2501, pp. 497–514, 2002. Article (CrossRef Link).

[6] Hui Li, Chuan-Kun Wu, Jun Sun, "A general compiler for password-authenticated group key exchange protocol," *Information Processing Letters*, pp.160–167, 2010. Article (CrossRef Link).

[7] T.Y. Wu, Y.M. Tseng, and T.T. Tsai, "A revocable ID-based authenticated group key exchange protocol with resistant to malicious participants," *Computer Networks*, Vol. 56, No. 12, pp. 2994-3006, 2012. Article (CrossRef Link).

[8] Tseng H, Jan R, Yang W, "A chaotic maps-based key agreement protocol that preserves user anonymity," in *Proc. of IEEE International Conference on Communications (ICC09)*, pp. 1–6, 2009. Article (CrossRef Link).

[9] M. Di Raimondo, R. Gennaro, "Provably secure threshold password-authenticated key exchange," *System Sci*, vol.72, No. 6, pp. 978–1001, 2006. Article (CrossRef Link).

[10] Khan MK, Zhang J, "Improving the security of a flexible biometrics remote user authentication scheme," *Comput Stand Interfaces*, vol. 29, No. 1, pp. 82–85, 2007. Article (CrossRef Link).
[11] Li LH, Lin IC, and Hwang MS, "A remote password authentication scheme for multi-server architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, No. 6, pp. 1498–1504, 2001. Article (CrossRef Link).
[12] Lin I C, Hwang MS, and Li LH, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, No. 1, pp. 13–22, 2003. Article (CrossRef Link).
[13] Tsai JL, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Comput Secur*, vol. 27, No. 3–4, pp.115–121, 2008. Article (CrossRef Link).
[14] Ravi SP, Jaidhar CD, Shashikala T, "Robust Smart Card Authentication Scheme for Multi-server Architecture," *Wireless Pers Commun*, vol. 72, Issue 1, pp. 729–745, 2013. Article (CrossRef Link).
[15] Ian Goldberg, "On the security of the Tor authentication protocol, In George Danezis and Philippe Golle," *Privacy Enhancing Technologies (PET)*, LNCS, vol.4258, pp. 316-331, 2006. Article (CrossRef Link).
[16] Aniket Kate, Greg M. Zaverucha, and Ian Goldberg, "Pairing-based onion routing with improved forward secrecy," *ACM Transactions on Information and System Security*, vol. 13, No. 4, pp. 29, 2010. Article (CrossRef Link).
[17] Paul Morrissey, Nigel P. Smart, and B. Warinschi, "A modular security analysis of the TLS handshake protocol," *Advances in Cryptology-Proc*, vol. 5350, pp. 55-73, 2008. Article (CrossRef Link).
[18] Ian Goldberg, Douglas Stebila, and Berkant Ustaoglu, "Anonymity and one-way authentication in key exchange protocols," *Codes Cryptogr*, vol. 67, pp. 245–269, 2013. Article (CrossRef Link).
[19]Tim Dierks and Christopher Allen, The TLS protocol version 1.0, 1999. Article (CrossRef Link).
[20] NIST National Institute of Standards and Technology, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography," *Special Publication 800-56B*, 2009. Article (CrossRef Link).
[21] Wang X, and Zhao J, "An improved key agreement protocol based on chaos," *Commun. Nonlinear Sci. Numer. Simul*, Vol. 15, pp. 4052–4057, 2010. Article (CrossRef Link).
[22] Zhang L, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos Solitons Fractals*, Vol. 37, no. 3, pp. 669–674, 2008. Article (CrossRef Link).
[23] Andreas Pfitzmann and Marit Hansen, "A terminology for talking about privacy by dataminimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," 2010. Article (CrossRef Link).
[24] Ravi SP, Jaidhar CD, and Shashikala T, "Robust Smart Card Authentication Scheme for Multi-server Architecture," *Wireless Pers Commun*, Vol. 72, pp. 729–745, 2013. Article (CrossRef Link).
[25] Ran Canetti, and Hugo Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," *EUROCRYPT*, Vol. 2045 of Lecture Notes in Computer Science, pp. 453-474, 2001. Article (CrossRef Link).
[26] Te-Yu Chen, Cheng-Chi Lee, Min-Shiang Hwang and Jinn-Ke Jan, "Towards secure and efficient user authentication scheme using smart card for multi-server environments," *J Supercomput* , vol. 66, Issue. 2, pp. 1008–1032, 2013. Article (CrossRef Link).
[27] Tsai JL, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Comput Secur*, Vol. 27, no. 3–4, pp. 115–121, 2008. Article (CrossRef Link).
[28] Liao YP, Wang SS, "A secure dynamic ID based remote user authentication scheme for multiserver environment," *Comput Stand Interfaces*, vol. 31, No. 1, pp. 24–29, 2009. Article (CrossRef Link).
[29] Hsiang HC, Shih WK, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Comput Stand Interfaces*, vol. 31, No. 6, pp. 1118–1123, 2009. Article (CrossRef Link).
[30] Kocarev L, and Lian S, "Chaos-Based Cryptography: Theory, Algorithms and Applications," pp. 53–54, 2011. Article (CrossRef Link).

Appendix. Explanation of some terms

(1) *Anonymity vs "OTP and ID hiding"*

Anonymity ensures that a user may use a resource or service without disclosing the user's identity completely.

ID hiding usually means that a user may use a resource or service without disclosing the user's identity during the protocol interaction, which is a kind of privacy protection partly. A *pseudonym* is an identifier of a subject other than one of the subject's real names. ID hiding usually uses *pseudonym* to realize. Because the server may store the user's identity.

OTP (one-time password) usually means that the password can be used only once but the *ID* is plaintext during the protocol interaction, so there is no privacy protection. The concrete differences are shown in **Table 6**.

Table 6. Comparisons among Anonymity, OTP and ID hiding

Terms	Privacy	Authentication	Security
	√ √ √	One way	√ √ √
	√	Mutual	√ √
	×	Mutual	√ √

×: Weak; √: Ordinary; √ √: Good; √ √ √: Excellent

(2) *Anonymity vs Unlinkability*

Unlinkability [23] of two or more items of interest (IOIs, e.g., messages, actions, ...) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not. So in the context of key exchange, unlinkability and anonymity are in a sense equivalent.

(3) *Anonymity vs Untraceability* [23]

Untraceability: The signer is unable to link the message-signature pair with the corresponding view after the blind signature has been revealed to the public by the requester. So anonymity is a general term and untraceability is used in signature usually.

(4) *Anonymity vs Undetectability*

Undetectability [23] of an item of interest (IOI) from an attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not. So we can defer undetectability is a kind of pseudor anonymity just like pseudorandom number and true random number.

(5) *One-way AKE vs One-flow AKE*

In brief, we can view a one-way AKE protocol as the complement of a one-flow AKE protocol. One-flow AKE protocols are designed to establish a session key using a single message from the client to the server. It can provide mutual authentication by using two static keys (one each from the client and the server) and one ephemeral key (from the client). In contrast, one-way AKE can use one static key (from the server) and two ephemeralkeys (one each from the client and the server), but provides no authentication to the server.

Hongfeng Zhu obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hongfeng Zhu is a full associate professor of the Kexin software college at Shenyang Normal University. He is also a master's supervisor. He has research interests in wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. Other things that interest him are number theory and the investigation of mathematics for designing secure and efficient cryptographic schemes. Dr. Zhu had published more than 60 international journal and international conference papers on the above research fields.