A Secure and Efficient Remote User Authentication Scheme for Multi-server Environments Using ECC

Junsong Zhang[1], Jian Ma[2], Xiong Li[3], and Wendong Wang[2]
[1] School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China
[e-mail: zhangjs2002@gmail.com]
[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
Beijing 100876, China
[e-mail: jian.j.ma@gmail.com, wdwang@bupt.edu.cn]
[3] School of Computer Science and Engineering, Hunan University of Science and Technology
Xiangtan 411201, China
[e-mail: lixiongzhq@163.com]
*Corresponding author: Xiong Li

Received October 24, 2013; revised December 27, 2013; revised April 10, 2014; revised June 11, 2014; accepted July 14, 2014; published August 29, 2014

Abstract

With the rapid growth of the communication technology, intelligent terminals (i.e. PDAs and smartphones) are widely used in many mobile applications. To provide secure communication in mobile environment, in recent years, many user authentication schemes have been proposed. However, most of these authentication schemes suffer from various attacks and cannot provide provable security. In this paper, we propose a novel remote user mutual authentication scheme for multi-server environments using elliptic curve cryptography (ECC). Unlike other ECC-based schemes, the proposed scheme uses ECC in combination with a secure hash function to protect the secure communication among the users, the servers and the registration center (RC). Through this method, the proposed scheme requires less ECC-based operations than the related schemes, and makes it possible to significantly reduce the computational cost. Security and performance analyses demonstrate that the proposed scheme can solve various types of security problems and can meet the requirements of computational complexity for low-power mobile devices.

Keywords: User authentication, elliptic curve cryptography, smart card, hash function

http://dx.doi.org/10.3837/tiis.2014.08.021

1. Introduction

Nowadays, mobile devices (i.e. smart phones, PDAs) are popular and widely used in many mobile applications, such as online shopping, mobile pay-TV, and electronic transactions. Along with the increasing number of mobile applications, the security issues have been received more and more attention. Generally, the user must be authenticated by the remote server before he accesses the services provided by the remote servers. The password-based authentication scheme is one of widely used mechanisms to verify the validity of the remote users over an insecure communication channel, and is a protective barrier that can prevent unauthorized personnel from accessing services provided by the application server.

In 1981, Lamport [1] first proposed a password-based remote authentication scheme for insecure communication. In Lamport's scheme, the user submits a message which contains his/her identity and password to the remote server, and then he/she can access to the remote server if the submitted information is matched with the information stored in the server's memory. This method is simple and practical, but it has some inherent drawbacks. The most serious problem is that the server has to store and maintain a verification table to verify the validity of the users, and then it will suffer from the password related attacks, such as stolen-verifier attack, off-line password dictionary attack, password table tampering and corruption attack. Later, some research activities [2] [3] focused on the password-based authentication schemes without verification tables. However, only password-based user authentication scheme is not sufficient to fulfill the security requirements of various applications. Since the smart card has the functions of storage, encryption, computation, etc., many smart card based password authentication schemes have been proposed [4] [5] [6] [7] [8] [21] by researchers. The smart card based password authentication schemes can not only solve the problem of storing the password table in the server, but can enhance the security of their schemes.

However, most of smart card based user authentication schemes are designed for the single server environments. In the practical application, it is extremely hard for a user to remember these different numerous identities and passwords when he/she uses the single-server authentication scheme to login and access to different remote servers. For this reason, there are also some other research activities [9] [10] [11] [12] [20] contributed to the authentication schemes for multi-server environments. The multi-server authentication scheme resolves the repeated registration problem of single-server authentication scenario where the user has to register at different servers to access to different types of network services. In 2001, Li *et al*. [9] proposed a multi-server user authentication protocol based on neural networks, however, it was found that the computational complexity of their scheme is too high [10]. To remedy the efficiency problem of Li *et al*.'s scheme, Juang [10] proposed a new multi-server password authentication protocol using the hash function and symmetric key cryptosystem. However, Chang and Lee [11] pointed out that Juang's protocol lacks efficiency and is vulnerable to off-line dictionary attack. Therefore, Chang and Lee proposed a novel authentication scheme to remedy these weaknesses. Nevertheless, their protocol was found to be vulnerable to insider attack, spoofing attack and registration center spoofing attack [12]. Then, Tsai [12] proposed an efficient multi-server authentication protocol based on one-way hash function.

However, all of the above smart card based multi-server authentication schemes are based on static ID, which gives the adversary a chance to trace a legal user. Consequently, to deal with this problem, some dynamic identity based user authentication schemes for

multi-server environments have been proposed [13] [14] [15] [19] [22]. In 2009, Liao and Wang [13] proposed a dynamic identity based authentication scheme for multi-server architecture. They claimed that their scheme can resist various attacks and can achieve mutual authentication. However, Hsiang and Shih [14] pointed out that Liao-Wang's scheme is still suffering from a variety of different attacks. Besides, Hsiang and Shih found that Liao-Wang's scheme cannot achieve mutual authentication. To solve these problems, Hsiang and Shih [14] proposed an improved protocol on Liao-Wang's protocol. Unfortunately, Lee *et al*. [15], Sood *et al*. [16], and Chuang and Tseng [17] respectively pointed out that Hsiang and Shih's scheme still suffers from different kinds of attacks, and they are all proposed the enhanced scheme. Later, Li *et al*. [18] and Li *et al*. [19] respectively pointed out Lee *et al*.'s scheme [15] and Sood *et al*.'s scheme [16] are both not secure to resist attacks.

Recently, many pairing-based or ECC-based remote user authentication schemes with smart cards were proposed in [17] [23] [24] [25]. In 2006, Das *et al*. [23] proposed a pairing-based remote client authentication scheme with smart cards. However, their scheme suffered from a forgery attack. Later, Fang and Huang [24] proposed an improvement to overcome drawback of Das *et al*.'s scheme. However, Giri and Srivastava [25] found that Fang and Huang's scheme is still suffering from another type of forgery attack. Then, Giri and Srivastava [25] presented another improvement to withstand the forgery attack. In 2012, Chuang and Tseng [17] proposed an ID-based mutual authentication and key agreement scheme based on bilinear maps for mobile multi-server environment. In 2013, Liao and Hsiao [4] proposed a pairing-based remote user authentication scheme which uses secure key distribution based on self-certified public keys among the service servers. However, the computational cost of pairing operation is approximately 20 times higher than that of the scalar multiplication with the same order. Then, several ID-based authentication protocols on ECC are proposed. In 2009, Yang and Chang [26] proposed an ID-based remote mutual authentication with key agreement protocol on ECC. In 2012, He *et al*. [5] propose an ID-based remote mutual authentication with key agreement scheme on ECC. However, He et al.'s scheme cannot support the multi-server environment.

In general, a secure and efficient remote user authentication scheme should satisfy the following requirements [18].

•User anonymity: The adversary cannot track a special user by the user's identity, therefore it can protect the user's privacy.

•Single registration: The scheme allows the user to register only once at the registration center and then the user can access to all the permitted services in the whole network.

•No password table: The registration center should be without the user's password table to authenticate the users.

•Low computation and communication cost: Due to the limited energy, processing and storage resources of mobile devices, the design of authentication scheme must take computation efficiency into consideration.

•Mutual authentication and session key agreement: In mutual authentication scheme, the server and the user must be able to authenticate each other via message exchange (mutual authentication). Through the mutual authentication procedure, the server and the user can negotiate and agree on a session key that is used in the following secret communication.

•Security: The authentication scheme must be able to prevent from various kinds of attacks.

In this paper, we propose a novel remote user mutual authentication scheme with key agreement protocol for mobile multi-server environments. Unlike other ECC-based researches, the proposed scheme uses a secure hash function to protect both the users' and the servers' secret values. Besides, the proposed scheme uses ECC in combination with a secure hash

function to protect the secure communication among the users, the servers and the registration center (RC for short, which is responsible for system initialization, user and server registration and authentication).. Through this method, the proposed scheme requires less ECC-based operations than the related schemes. Performance analysis is made to show that our scheme has better performance than the related schemes [5] [17] [26]. The security analysis shows that the proposed scheme can withstand various possible attacks. Compared with the related works, the proposed scheme is more secure and efficient for remote authentication.

The remainder of this paper is organized as follows. In Section 2, we briefly review the basic concepts of ECC and some related mathematical preliminaries. Section 3 describes the proposed authentication and key agreement scheme. Then, we present the security analysis and performance evaluation about the proposed scheme in Section 4 and Section 5, respectively. Finally, Section 6 concludes the paper.

2. Preliminaries

In this section, we briefly introduce the basic concepts of the elliptic curve and its related mathematical properties. The elliptic curve cryptography (ECC) was first proposed by Miller and Koblitz (1985), respectively. The security of ECC is dependent on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP). Due to the difficulty in breaking its encryption, Elliptic Curve Cryptography can provide the same level of RSA encryption at a greatly reduced bit size [26] [27].

2.1 Elliptic Curve Cryptography (ECC)

Let p be a prime number, and let $GF(p)$ denote the field of integers modulo p. An elliptic curve E over $GF(p)$ is defined by an equation of the form:

$$y^2 = x^3 + ax + b \tag{1}$$

where $a, b \in GF(p)$, and satisfies $4a^3 + 27b^2 \neq 0 \pmod p$. A pari $Q(x_1, y_1)$ is a point on the curve if $Q(x_1, y_1)$ satisfies the equation (1), where $x_1, y_1 \in GF(p)$. The point at infinity, denoted by ∞, is also said to be on the curve. Any point pair $Q(x, y)$ which satifies the above equation together with ∞, called 'point at infinity', form an additive cyclic group $E(F_p) = \{(x, y) \mid x, y \in GF(p)$ satisfy $y^2 = x^3 + ax + b \pmod p\} \cup \infty$. The scalar multiplication of a point P over $E_p(a, b)$ is computed by repeated addition as, $k \cdot P = P + P + \cdots + P$ (k times), where k is a constant integer and P is a point on the curve E. For simplicity, we omit the details and the readers can refer to the related references [27].

2.2 Computational Problems

To prove the security of our proposed protocol, we present some important mathematical properties of ECC as follows.

Definition 1. Elliptic Curve Discrete Logarithm Problem (ECDL Problem): Given an elliptic curve E defined over a finite field $GF(p)$, and two points $Q, P \in E$, it is hard to find an integer $k \in Z_q^*$ such that $Q = k \cdot P$.

Definition 2. Computational Diffie-Hellman Problem (CDH Problem): Given an elliptic curve E defined over a finite field $GF(p)$, a point $P \in E$ of order q, and points $A = a \cdot P$; $B = b \cdot P$, it is hard to find the point $C = abP$.

Definition 3. The Elliptic Curve Decision Diffie-Hellman Problem (ECDDH Problem): Given an elliptic curve E defined over a finite field $GF(p)$, a point P of order q, and points $A =$

$a \cdot P$, $B = b \cdot P$, and $C = c \cdot P$, determine whether or not $C = abP$, equivalently, whether $c \equiv a \cdot b \pmod{p}$ or not.

To the best of our knowledge, there is no polynomial time algorithm to solve any of the above-mentioned problems with non-negligible probability [28].

2.3 Hash function

Hash function [29] is an algorithm that takes an arbitrary block of data and returns a fixed-size bit string, and it is easy to compute on every input but hard to compute the input from a given output. Here "easy" and "hard" are to be understood in the sense of computational complexity theory. The one-way hash function has the following properties.

- The hash function can be applied to a data block of all sizes.
- For any given input x, it is esay to compute the output.
- It is infeasible to deriving x from the given value $y = h(x)$.
- It is infeasible to find two different inputs with the same output.
- It is infeasible to modify an input without changing the output.

In the proposed scheme, two kinds of hash function will be used: $h(\cdot)$ and $H(\cdot)$. The hash function $h(\cdot)$ maps binary strings of an arbitrary length to an integer of fixed bit length (h: $\{0, 1\}^* \rightarrow Z_q^*$). The hash function $H(\cdot)$ maps a point over the elliptic curve E to a binary strings of a fixed length (H: $E_p(a, b) \rightarrow \{0, 1\}^l$, l is the length of the string). In our scheme, $H(\cdot)$ can be constructed by the hash function $h(\cdot)$: Let P_i be a point over the elliptic curve E, let P_x and P_y be the X-coordinate and the Y-coordinate of the point P_i, respectively. Then, $H(\cdot)$ can be sonstructed as $H(P_i) = h(P_x) \| h(P_y)$, where $\|$ is string concatenation operation.

3. The Proposed Scheme

In this section, we propose a novel authentication scheme using ECC to avoid various attacks. ECC is one of the most efficient public key systems nowadays. It has been shown that the ECC has been proven to be successful on the limited hardware [30]. The main idea of the proposed scheme is using the ECC in combination with a secure hash function to encrypt the exchange messages among the user, the server and the registration center (*RC*). Without loss of generality, the proposed authentication scheme consists of four phases: the registration phase, the login phase, the authentication and session key agreement phase and the password change phase. The login and authentication phases are depicted in **Fig. 1**, and more details of all these phases are describe as follows.

For the sake of clarity, the notations used in this paper are summarized and defined in **Table 1**.

Table 1. List of notations

Notation	Description
U_i	The ith user
ID_i	The $i$dentity of U_i
PW_i	The password of U_i
RC	Registration center
S_j	The jth server
SID_j	The identity of S_j

x, y	The secret keys maintained by registration center
TU_i	U_i's time stamp
$h(\cdot)$	A one-way hash function, where h: $\{0, 1\}^* => Z_p^*$
$H(\cdot)$	A hash function that maps a point to a string, H: $E_p(a, b) => \{0, 1\}^l$, l is the length of the string
$\oplus$	The bitwise XOR operation
$\|\|$	String concatenation operation
=>	A secure channel
$\rightarrow$	A common channel

3.1 System Initialization

Without loss of generality, it is assumed that the multi-server environment includes three kinds of participants: a trusted registration center (*RC*), users and servers. *RC* is responsible for system initialization, user and server registration and authenticaion. The detailed initialization steps are performed as follows.

At the beginning, *RC* chooses a prime number p and an elliptic curve equation E over the finite field $GF(p)$, and then selects a base point P with the order q over E, where q is a 512-bit prime number and $q \leqslant p$. Then, *RC* selects a random number $s \in Z_q^*$ as the master secret key, and computes the public key $P_{pub} = s \cdot P$.

Next, *RC* chooses two secret keys x, $y \in Z_q^*$, and chooses two secure hash functions $h(\cdot)$ and $H(\cdot)$, where h: $\{0, 1\} => Z_p^*$, and H: $E_p(a, b) => \{0, 1\}^l$, l is the length of the string.

Then, *RC* computes $h(SID_j \| y)$ and $h(SID_j \oplus y)$, and shares these authentication certificate with the corresponding server S_j via a secure channel.

In the end, *RC* publishes the public system parameters as $\{E, GF(p), q, P, P_{pub}, h(\cdot), H(\cdot)\}$ and keeps the master secret key s and values x and y secretly.

3.2 The Registration Phase

When a user U_i wants to use this system, he/she has to submit his/her identity and password to *RC* for registration. The steps of the registration phase are as follows:

Step R1. $U_i \rightarrow RC$: $\{ID_i, h(b \oplus PW_i)\}$.

The user U_i chooses his/her identity ID_i and password PW_i freely. U_i's terminal generates a random number b. Then, U_i computes $h(b \oplus PW_i)$ and sends the message $\{ID_i, h(b \oplus PW_i)\}$ to *RC* via a common channel.

Step R2. *RC* computes

$T_i = h(ID_i \| x)$,

$Q_i = h(T_i)$,

$V_i = T_i \oplus h(ID_i \| h(b \oplus PW_i))$

$R_i = h(ID_i \| h(b \oplus PW_i) \| x)$.

Step R3. *RC* issues a smart card to U_i, and the card contains $\{V_i, H_i, R_i, h(\cdot), H(\cdot), E, P\}$.

Step R4. U_i's terminal inserts the value b into the smart card. At last, the smart card contains $\{V_i, H_i, R_i, b, h(\cdot), H(\cdot), E, P\}$.

3.3 The Login Phase

When the user U_i wants to login to the server S_j, he/she inserts his/her smart card to a terminal (smart phone or PDA) and inputs his/her identity ID_i and password PW_i.

Step L1. The smart card computes $T_i^* = V_i \oplus h(ID_i \| h(b \oplus PW_i))$, $Q_i^* = h(T_i^*)$, and checks whether Q_i^* is equal to Q_i. If they are equal, the terminal proceeds to the next step. Otherwise, the smart card rejects this request.

Step L2. $U_i \rightarrow S_j$: $\{C_1, C_2, TU_i\}$. The smart card chooses a random number $k \in Z_q^*$ and computes:

$w_i = TU_i \oplus (ID_i \| h(b \oplus PW_i) \| SID_j \| R_i)$,

$C_1 = k \cdot P$, $P_{ur} = k \cdot P_{pub}$,

$C_2 = w_i \oplus H(P_{ur})$,

where TU_i denotes the current timestamp and SID_j denotes the identity of the destination server. Then, U_i sends the login request $\{C_1, C_2, TU_i\}$ to the server S_j for authentication.

3.4 The Authentication and Session Key Agreement Phase

Step A1. $S_j \rightarrow RC$: $\{SID_j, C_1, C_2, C_3, N_j, TU_i, aP\}$.

When the server S_j receives the login request $\{C_1, C_2, TU_i\}$, it checks whether $TU_i - T_s \leqslant \Delta T$, where T_s is the time when S_j receives this request and ΔT is the expected time interval for the transmission delay. If so, S_j generates a random number $N_j \in Z_q^*$, then computes $m_1 = h(SID_j \oplus y) \| N_j$, $C_3 = m_1 \oplus H(a \cdot P_{pub})$, where a is the server S_j's master key.

Then, S_j sends the registration request $\{SID_j, C_1, C_2, C_3, N_j, TU_i, aP\}$ to RC via a common channel.

Step A2. Upon receiving $\{SID_j, C_1, C_2, C_3, N_j, TU_i, aP\}$, RC checks whether $TU_i - T_{RC} \leqslant \Delta T$, where T_{RC} is the time when RC receives the above message. If so, RC computes:

$P_{ru} = s \cdot C_1$,

$w_i^* = C_2 \oplus H(P_{ru})$,

$m_1^* = C_3 \oplus H(s \cdot (aP))$.

The above formulae can be deduced as follows:

$w_i^* = C_2 \oplus H(P_{ru}) = C_2 \oplus H(s \cdot C_1) = C_2 \oplus H(s \cdot (k \cdot P)) = C_2 \oplus H(k \cdot (s \cdot P)) = C_2 \oplus H(k \cdot P_{pub}) = C_2 \oplus H(P_{ur}) = w_i = TU_i \oplus (ID_i \| h(b \oplus PW_i) \| SID_j \| R_i)$,

$m_1^* = C_3 \oplus H(s \cdot (aP)) = C_3 \oplus H(a \cdot (sP)) = C_3 \oplus H(a \cdot P_{pub}) = m_1 = h(SID_j \oplus y) \| N_j$.

Therefore, RC can extract the user's identity ID_i and $h(b \oplus PW_i)$ by computing $w_i \oplus TU_i$. At this point, RC has both U_i's and S_j's identity. RC computes $h(SID_j \oplus y)$ and compares it with the value extracted from m_1^*. If they are equal, RC considers S_j as a legal server. Then RC computes $h(ID_i \| h(b \oplus PW_i) \| x)$ and compares it with the value R_i which is extracted from w_i^*. If $h(ID_i \| h(b \oplus PW_i) \| x) = R_i$, RC considers U_i as a legal user. Then RC sends a message to S_j in the next step.

Step A3. $RC \rightarrow S_j$: $\{C_4, C_5, TU_i\}$. The RC computes:

$m_2 = h(SID_j \oplus y) \| ID_i \| h(ID_i \| x)$,

$m_3 = h(ID_i \| x) \| SID_j \| N_j \| h(SID_j \| y)$,

$C_4 = m_2 \oplus H(s \cdot (aP))$,

$C_5 = m_3 \oplus H(P_{ru})$.

Then, RC sends the message $\{C_4, C_5, TU_i\}$ to the server S_j via a common channel.

Step A4. $S_j \rightarrow U_i$: $\{C_5, N_j, TU_i\}$. Upon receiving $\{C_4, C_5, TU_i\}$, S_j computes $m_2^* = C_4 \oplus H(a \cdot (P_{pub}))$, where a is S_j's master key. Then, S_j extracts the value $h(SID_j \oplus y)$ from m_2^* and compares it with of $h(SID_j \oplus y)$ stored in its memory to verify RC. If they are equal, S_j considers U_i as a legal user and sends the message $\{C_5, N_j, TU_i\}$ to the user U_i. S_j further computes $SK = h(h(ID_i \| x) \| h(SID_j \| y) \| TU_i \| N_j)$ as the session key for securing communications with U_i.

Step A5. After receiving $\{C_5, N_j, TU_i\}$, U_i computes $m_3^* = C_5 \oplus H(k \cdot P_{pub})$, then U_i extracts the value $h(ID_i \| x)$ from m_3^* and computes $H_i^* = h(h(ID_i \| x))$. Next, U_i compares H_i^* with H_i. If they are equal, U_i considers S_j as a legal server and further computes $SK = h(h(ID_i \| x) \| h(SID_j \| y) \| TU_i \| N_j)$ as the session key for securing communications with S_j.

3.5 The Password Change Phase

This phase is invoked whenever U_i wants to change his/her password PW_i to a new password PW_{new}. The steps of the password change phase are as follows.

Step P1. U_i inserts his/her smart card into the terminal and then keys his/her ID_i and PW_i.

Step P2. The smart card computes $T_i = V_i \oplus h(ID_i \| h(b \oplus PW_i))$, $Q_i^* = h(T_i)$ and checks whether Q_i^* is equal to Q_i, If they are equal, the smart card lets U_i choose a new password PW_{new} and a new random number b_{new} to compute:

$h(b_{new} \oplus PW_{new})$,

$V_{new} = T_i \oplus h(ID_i \| h(b_{new} \oplus PW_{new}))$,

$C_6 = (ID_i \| h(b_{new} \oplus PW_{new})) \oplus H(k \cdot P_{pub})$.

Then, U_i sends the message $\{ID_i, C_1, C_2, C_6, TU_i\}$ to RC via a secure channel. Step P3. Upon receiving $\{ID_i, C_1, C_2, C_6, TU_i\}$, RC checks whether the time stamp TU_i is valid. If so, RC determines the legitimacy of U_i using the values C_1 and C_2. Then, RC computes $C_6 \oplus H(s \cdot C_1) = C_6 \oplus H(s \cdot (k \cdot P)) = C_6 \oplus H(k \cdot P_{pub}) = ID_i \| h(b_{new} \oplus PW_{new})$.

Step P4. RC calculates $R_{new} = h(ID_i \| h(b_{new} \oplus PW_{new}) \| x)$, $C_7 = (R_{new} \| h(ID_i \| x) \| TU_i) \oplus H(s \cdot C_1)$ and then sends the message $\{ID_i, C_7, TU_i\}$ to U_i.

Step P5. After receiving $\{ID_i, C_7, TU_i\}$, U_i computes $C_7 \oplus H(k \cdot P_{pub}) = C_7 \oplus H(s \cdot (k \cdot P)) = C_7 \oplus H(s \cdot C_1) = R_{new} \| h(ID_i \| x) \| TU_i$. Then, U_i computes $Q_i^* = h(h(ID_i \| x))$ and compares it with Q_i stored in his/her smart card to check the validity of this message. If they are the same, the smart card replaces b, V_i, R_i with b_{new}, V_{new}, and R_{new}, respectively.

4. Security Analysis of the Proposed Scheme

In this section, we use the random oracle model to analyze the security of our scheme. The random oracle model is often used to verify the security of the key establishment protocol or the signature scheme. In the model, each participant appeared in the authentication and key agreement scheme is treated as an oracle, and the adversary can access these oracles by sending some queries.

4.1 Adversarial Model

In this section, we define the adversarial model of a mutual authentication and key agreement protocol. Readers can refer to [33] to get the detailed descriptions about the following definitions. Assume that the multi-server environment contains three types of participants: n users $\mathbf{U} = \{U_1, U_2, \ldots, U_n\}$, m servers $\mathbf{S} = \{S_1, S_2, \ldots, S_m\}$ and a registration center (RC). The lth instance of U_i (resp. S_j) is denoted by Π_U^l (resp. Π_S^l). An adversary A is a probabilistic polynomial time machine. It is assumed that A is able to potentially control all common communications in the proposed scheme via accessing to a set of oracles (as defined below). The proposed scheme's public parameters are known by each participant (including the users, servers and RC).

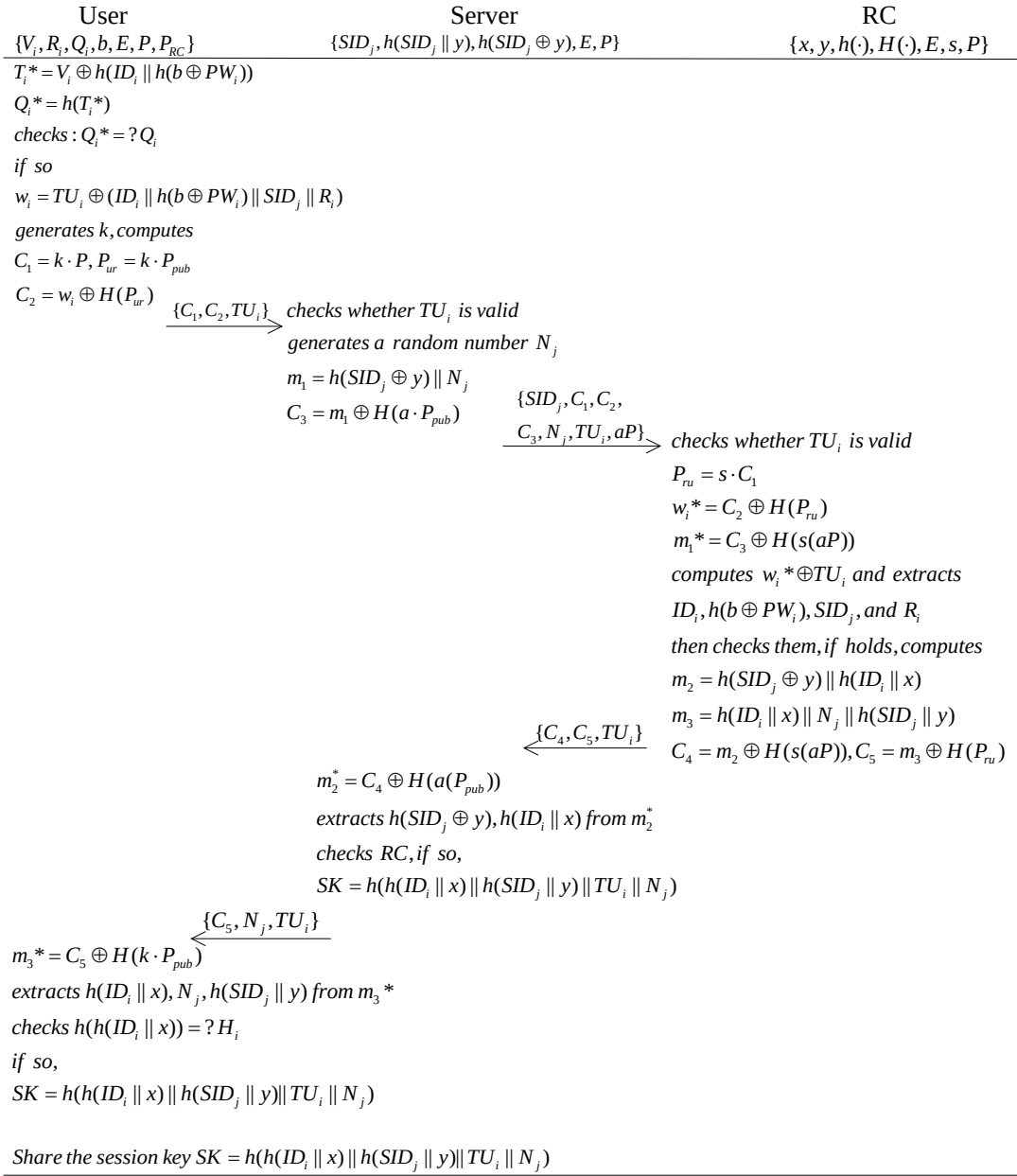

Fig. 1. The login and authentication phases of our scheme

•*Extract*(ID_i): In this query model, an adversary A is able to get the private key of ID_i.

•Send(Π_c^k, M): The adversary A can send a message M to the oracle Π_c^k, where c $\in$ {**U**, **S**, *RC*}. When receiving the message M, Π_c^k responds to A according to the proposed scheme.

•$h(m_i)$: When an adversary A makes this hash query with the message m_i, the oracle Π_c^k returns a random number r_1 and records (m_i, r_1) into a list L_h. The list is initially empty.

•$H(P_i)$: When an adversary A makes this hash query with a point P_i over the curve E, the oracle Π_c^k returns a random number r_2 and records (P_i, r_2) into a list L_H. The list is also empty initially.

•*Reveal*(Π_c^k): In this query model, the adversary A can obtain a session key SK from the oracle Π_c^k if the oracle has accepted. Otherwise, Π_c^k returns a null to A.

•*Corrupt*(ID_i): The adversary A can issue this query to ID_i and gets back its private key.

•*Test*(Π_c^k): When A asks this query to an oracle Π_c^k, the oracle chooses a random bit $b \in \{0, 1\}$. If $b = 1$, then Π_c^k returns the session key. Otherwise, it returns a random value. This query measures the semantic security of the session key.

In this model, the adversary A can make *Send*, *Reveal*, *Corrupt* and *Test* queries. Note that the capabilities of the adversary can make finite queries under adaptive chosen message attacks [33].

4.2 Security Analysis

In this subsection, we first demonstrate that the proposed scheme can achieve the secure authentication and key agreement under the random oracle model. Then, we analyze the other security properties about the proposed scheme.

4.2.1 Formal analysis under the random oracle model

In this subsection, we show that the proposed scheme can provide the secure authentication and key agreement under the computational Diffie-Hellman problem (CDH) assumption.

Theorem 1. Assume that an adversary A can violate the proposed scheme with a non-negligible advantage ε and makes at most q_u, q_s, q_h, and q_H queries to the oracle of the user Π_U^i, oracle of the server Π_S^j, h, and H, respectively. Then we can construct an algorithm to solve the CDH problem with a non-negligible advantage.

Proof. We first classify the types of attack into two categories: impersonating the user to communicate with *RC* and impersonating the server to communicate with *RC*. Then we can construct an algorithm to solve the CDH problem.

For an instance of the CDH problem {q, P, $Q_1 = xP$, $Q_2 = yP$}, B simulates the system initializing algorithm to generate the system parameters {E, q, P, $P_{pub} = Q_2$, h, H}. h and H are random oracles controlled by B. Then, B gives the system parameters to A. B interacts with A as follows.

h-query: B maintains a list L_h of tuples (str_i, h_i). When A queries the oracle h on (str_i, h_i), B responds as follows:

If str_i is on L_h, then B responds with h_i. Otherwise, B randomly chooses an integer h_i that is not found in L_h, and adds (str_i, h_i) into L_h, then responds with h_i.

H-query: B maintains a list L_H of tuples (P_c, H_c). When A queries the oracle H on (P_c, H_c), B responds as follows:

If P_c is on L_H, then B responds with H_c. Otherwise, B randomly chooses an integer H_c that is not found in L_H, and adds (P_c, H_c) into L_H, then responds with H_c.

Reveal-query: When the adversary *A* makes a *Reveal*(Π_c^m) query, *B* responds as follows. If Π_c^m is not accepted, *B* responds none. Otherwise, *B* examines the list L_h and responds with the corresponding h_i.

Send-query: When the adversary *A* makes a *Send*(Π_c^m, "start") query, *B* responds as follows. If $\Pi_c^m = \Pi_u^m$, *B* sets $C_1 \leftarrow Q_1$, and randomly generates the values C_2 and TU_i, and responds with $\{C_1, C_2, TU_i\}$. Otherwise, *B* generates a random number *k* and computes $C_1 = kP$, $C_2 = w_i \oplus kP_{pub}$ and responds with $\{C_1, C_2, TU_i\}$, where w_i is generated by *B*. The simulation works correctly since *A* cannot distinguish whether w_i is valid or not unless *A* knows the *RC*'s private key.

When the adversary *A* makes a *Send*(Π_c^m, (SID_j, C_1, C_2, C_3, N_j, TU_i, aP)) query, *B* responds as follows. If the Π_c^m matches the Π_u^m, *B* cancels the game. Otherwise, *B* computes $w_i = C_2 \oplus H(P_{ru})$, where P_{ru} is generated in *Send*(Π_c^m, "start"). Then *B* responds the corresponding message according to the description of the proposed scheme.

When the adversary *A* makes a *Send*(Π_c^m, (C_5, N_j, TU_i)) query, *B* responds as follows. If the Π_c^m matches the Π_s^m, *B* cancels the game. Otherwise, *B* computes $m_3^* = C_5 \oplus H(_{pur})$ and then checks the *RC*. If so, *B* computes the session key $SK = h(h(ID_i \| x) \| h(SID_j \| y) \| TU_i \| N_j)$, where $h(SID_j \| y)$ is extracted from m_3^*, and $h(ID_i \| x)$ is extracted from L_h.

If the adversary *A* can violate a user to the *RC* authentication, it means that *A* can get the values of $h(ID_i \| x)$ and $R_i = h(ID_i \| h(b \oplus PW_i) \| x)$ from the list L_h and then compute the values of C_1 and C_2. In such case *B* can extract the corresponding $H_c = s \cdot (kP)$ in L_H using $Q_2 = P_{pub}$, $C_1 = kP$ with non-negligible probability. Therefore, if the adversary *A* can violate a user to the *RC* authentication, then *B* is able to solve the CDH problem with non-negligible probability. It is a contradicting to the intractability of the CDH problem. In addition, *x* is the *RC*'s secret value and *h* is a secure hash function. Therefore, the probability that *A* can compute the value $R_i = h(ID_i \| h(b \oplus PW_i) \| x)$ is negligible. From the above analysis, we can see that the probability that *A* can violate the user to the *RC* authentication is negligible.

If the adversary *A* can violate the server to *RC* authentication, it means that *A* can get $h(SID_j \oplus y)$ and $H_c = aP$ from the lists L_h and L_H, respectively, and further computes $C_3 = m_1 \oplus H(a \cdot P_{pub})$ and makes a *Send*(Π_c^m, (SID_j, C_1, C_2, C_3, N_j, TU_i, aP)) query. Then *B* must have made the corresponding $H_c = a \cdot (P_{pub}) = s \cdot (aP)$ from the list L_H with non-negligible probability. Consequently, *B* is able to solve the CDH problem with non-negligible probability, which is a contradicting to the intractability of the CDH problem. Besides, *y* is the *RC*'s secret value that the adversary cannot obtain, the hash functions *h* and *H* are collision-resistant hash functions. Therefore, the probability that *A* can violate the server to the *RC* authentication is negligible.

If *B* is able to win such a game, then *B* must have made the corresponding *h* query of the form (str_i, h_i) to find the correct h_i (the session key) with non-negligible probability since *h* is a random oracle. The string str_i contains the values of $h(ID_i \| x)$, $h(SID_j \| y)$, TU_i and N_j. If *B* can get the value $h(ID_i \| x)$, it can find the corresponding H_c in L_H with the probability $1/q_H$ and output the corresponding P_c as a solution to the CDH problem. Therefore, if the adversary *A* can compute the correct session key with non-negligible probability ε, then the probability that B solves the CDH problem is $\varepsilon / q_h \cdot q_H$. It is a contradicting to the intractability of the CDH problem.

4.2.2 Other security features

•Replay attack

A replay attack is a form of network attack where an authentication session is replayed by an attacker to fool a computer into granting access. In the proposed scheme, the time stamp TU_i is used to resist this type of attack. After incepting the previous login request $\{C_1, C_2, TU_i\}$ from the user U_i, the adversary may replay the same message to the service S_j. However, he/she cannot launch a replay attack. Since the time stamp TU_i is a parameter of the value C_2, the replay attack can be filtered obviously by the server. After receiving the message $\{C_1, C_2, TU_i\}$, the server firstly checks the validity of time stamp TU_i. If the time stamp is not valid, the request will be rejected. If the adversary modifies the time stamp TU_i with a new time stamp TU_i^*, he/she cannot compute the corresponding C_2^* without the knowledge of $H(k \cdot P_{pub})$. Thus, the bogus TU_i^* can be found by checking C_2.

In addition, when an attacker replays the message $\{SID_j, C_1, C_2, C_3, N_j, TU_i, aP\}$ or $\{C_5, N_j, TU_i\}$ to impersonate as a legal server S_j, he/she has no capacity to compute the value C_3 or C_5. In the proposed scheme, the values of C_3 and C_5 are computed as $C_3 = m_1 \oplus H(a \cdot P_{pub})$, $C_5 = m_3 \oplus H(s \cdot C_1)$. Since N_j is a parameter of m_1 and m_3, any change of N_j can affect the values of C_3 and C_5, indirectly. When the adversary modifies the value of N_j, *RC* or users can detect the replay attack immediately. From the above analysis, we can see that our scheme can resist replay attack.

•Stolen smart card attack

We assume that the user U_i's smart card has been lost or stolen, so the attacker can breach the values $\{V_i, H_i, R_i, b, h(\cdot), H(\cdot), E, P\}$ which are stored in the smart card. It is important to note that the adversary cannot get the *RC*'s secret value x, hence he/she cannot guess the U_i's identity ID_i and password PW_i from the breached values in polynomial time since they are protected by a secure hash function. Consequently, the adversary cannot compute $h(b \oplus PW_i)$ and further compute the correct $w_i = TU_i \oplus (ID_i \| h(b \oplus PW_i) \| SID_j \| R_i)$. As a result, the adversary cannot use the lost or stolen smart card to carry out impersonation attack. From the above analysis, it can be seen that the proposed scheme effectively resists the stolen smart card attack.

•Impersonation attack

In the proposed scheme, every legal user must be registered in the *RC* when he/she wants to use this authentication scheme. Then, the *RC* issues U_i a unique value $R_i = h(ID_i \| h(b \oplus PW_i) \| x)$, where x is the *RC*'s secret value and PW_i is U_i's password. If an adversary wants to impersonate as the user U_i to communicate with the *RC*, he/she cannot compute the correct C_2 without the knowledge of R_i. Thus, the attacker has no way to impersonate a legitimate user to make a request.

In addition, every server has a unique value $h(SID_j \oplus y)$ and only the *RC* knows all the servers' authentication certificates. If an adversary wants to impersonate as the server to communicate with the *RC*, he/she cannot compute the correct C_3 without the authentication certificate of $h(SID_j \oplus y)$. Thus, the adversary cannot impersonate as a server to communicate with other entities. From the user's or the server's perspective, only the legal *RC* has the secret values x and y. Consequently, there is no way for an attacker to impersonate as the *RC*. From the above analysis, it can be seen that the proposed scheme can withstand this type of attack.

•User's anonymity

In the registration phase and password change phase of our authentication scheme, the user communicates with the *RC* via a secure channel. Therefore, other entities (including the adversary and other users) cannot obtain the user's identity. In the login phase of the proposed scheme, the user U_i submits the message $\{C_1, C_2, TU_i\}$ to the server S_j for authentication.

However, U_i's identity ID_i is contained in the value C_2. The adversary cannot extract U_i's identity without the knowledge of $H(P_{ur})$. In the authentication and session key agreement phase of the proposed scheme, the *RC* responds the message $\{C_4, C_5, TU_i\}$ to the server S_j. The value $h(ID_i \| x)$ is contained in C_4 and C_5, respectively. The adversary cannot extract $h(ID_i \| x)$ unless he/she gets the values of $H(s\cdot(aP))$ and $H(P_{ru})$. In addition, since the time stamp TU_i is a parameter of the value C_2, C_2 will be different for each session when the user logins to the system at different time. Therefore, the attacker cannot distinguish a certain user from different sessions. From the above analysis, it can be seen that our proposed protocol can provide user's anonymity.

•Man-in-the-middle attack

A man-in-the-middle attack would be able to successfully only when the attacker can impersonate each party to cheat other parties. In the proposed scheme, the exchange values C_2, C_3, C_4, and C_5 are XORed by $H(P_{ur})$ and $H(a\cdot P_{pub})$, respectively. Since the adversary cannot compute the values of $H(P_{ur})$ and $H(a\cdot P_{pub})$, he/she cannot extract the detailed information (ID_i, $h(b \oplus PW_i)$, $h(SID_j \oplus y)$, etc.) from the values of C_2, C_3, C_4, and C_5. Thus, it is clear that the adversary cannot launch the man-in-the-middle attack to cheat the user or the server.

5. Performance Evaluation

In this section, we evaluate the performance of our scheme and compare it with other related authentication schemes. It is generally known that most of the mobile devices have limited power resources and computing capability. Therefore, one of the most important concerns of design authentication scheme in mobile environment is power consumption (include computation cost and communication cost). We first evaluate the computational cost and then focus on the communication cost of the proposed scheme. Since exclusive-OR operation requires extremely small computational cost, we neglect its computation cost.

For the convenience of evaluating the computational cost, we define some notations as follows.

•T_{mul}: The time of executing a scalar multiplication operation of point.

•T_{add}: The time of executing an addition operation of points.

•T_{pair}: The time of executing a bilinear pairing operation.

•T_h: The time of executing a one-way hash function.

•T_{mtph}: The time of executing a map-to-point hash function.

•T_H : The time of executing a hash function $H(\cdot)$ used in this paper.

•T_{exp}: The time of executing a modular exponential operation.

In **Table 2**, we summarize the computation cost of the proposed user authentication and key agreement scheme. The user's terminal is a low-power computing device while the server and the *RC* are regarded as powerful devices. Since the authentication information contained by the user and the server (i.e. $R_i = h(ID_i \| h(b \oplus PW_i) \| x)$ and $h(SID_j \oplus y)$) are protected by a secure hash function $h(\cdot)$, the *RC* needs only to compute several hash functions to authenticate the legality of both the user and the server. From **Table 2**, we can see that the user requires only $2T_{mul} + 2T_h + T_H$ in login phase and $2T_h$ in authentication and key agreement phase. In authentication and key agreement phase, The *RC* and the server require only $2T_{mul} + 5T_h + 2T_H$ and $T_{mul} + 2T_h + T_H$, respectively.

Table 2. Computation costs of the proposed scheme

	User	**Server**	**RC**
Login phase	$2T_{mul} + 2T_h + T_H$	-	-
Authentication phase	$2T_h$	$T_{mul} + 2T_h + T_H$	$2T_{mul} + 5T_h + 2T_H$

Table 3. Computational cost on the client side and the server side

	T_{mul}	T_{add}	T_{pair}	T_h	T_{mtph}	T_{exp}
Client	0.13s	＜0.1s	0.38s	＜0.001s	＜0.01s	0.07s
Server	1.17ms	＜0.1ms	3.16ms	＜0.01ms	＜1ms	0.62ms

Table 4. Comparisons between the previously proposed schemes and our scheme

	Chuany and Tseng [17]	He *et al*. [5]	Yang and Chang [23]	Our scheme
Computational cost of client	$4T_{mul} + T_{exp} + T_{add} + 5T_h$	$3T_{mul} + 4T_h$	$4T_{mul} + 2T_{add} + T_{mtph} + 3T_h$	$2T_{mul} + 4T_h + 2T_H$
Execution time of client	≈0.69s	≈0.4s	≈0.73s	≈0.26s
Computational cost of server	$2T_{pair} + 3T_{mul} + T_{exp} + T_{add} + 5T_h$	$3T_{mul} + 5T_h$	$4T_{mul} + 2T_{add} + T_{mtph} + 3T_h$	$T_{mul} + 2T_h + T_H$
Execution time of server	≈10.6ms	≈3.56ms	≈5.91ms	≈1.2ms
Computational cost of *RC*	$T_{add} + 2T_h$	-	-	$2T_{mul} + 5T_h + 2T_H$
Execution time of *RC*	≈0.12ms	-	-	≈2.4ms
Multi-server environment	YES	NO	NO	YES
Known attacks	Provably secure	Provably secure	Impersonation attack	Provably secure

In order to present an objective and detailed comparison between our scheme and other related authentication schemes, we make the computation costs analysis on the basis of the implementation results in [31] . In [31], The hardware platform is Philips HiPersmart card and Pentium IV offer maximum clock speeds of 36 MHz and 3 GHz, respectively. In order to provide adequate security, a popular and valid choice would be to use the elliptic curve $y^2 = x^3 + ax + b$ defined on a finite field F_p with $p = 512$ bits and a large prime order $q = 160$ bits. All the operations are built with MIRACLE [32], a standard cryptographic library. **Table 3** lists the experimental data for related pairing-based operations on the Philips HiPersmart card and on the Pentium IV processor, respectively.

In **Table 4**, we demonstrate the comparisons between our proposed scheme and the previously authentication schemes on elliptic curve cryptography in terms of the computation cost, the execution time and security properties. The execution time appeared in **Table 4** are measured based on the computational cost listed in **Table 3**. In **Table 4**, we assume that the execution time of the hash function $H(\cdot)$ used in our scheme is equal to T_h because $H(\cdot)$ can be constructed by the hash function $h(\cdot)$: Let P_i be a point over E, let P_x and P_y be the X-coordinate and the Y-coordinate of the point P_i, respectively. Then, we can construct $H(\cdot)$ as

$H(P_i) = h(P_x \| P_y)$, where $\|$ is string concatenation operation. Therefore, we can see that T_H is equal to T_h. From **Table 4**, we can see that Yang and Chang's scheme suffered from impersonation attack. Based on an overall consideration of efficiency and security, our scheme is better than the previously proposed schemes listed in **Table 4**. Therefore, the proposed scheme is more suitable for the mobile client server environments.

The communication cost of the proposed scheme includes the exchange messages involved in the login and authentication phase. While, in the proposed authentication scheme, only the user's terminal is a resource -constrained device. Thus, we do not consider the communication cost between the server and the *RC*. Therefore, the number of communication parameters includes $\{C_1, C_2, TU_i\}$ and $\{C_5, N_j, TU_i\}$. In order to provide adequate security, as mentioned above, we choose the elliptic curve E defined on a finite field F_p with p = 512 bits and a large prime order q = 160 bits. Thus, the bit-length of a point P is 1024 bits. Since the parameter C_1 is a point on the curve E, the bit-length of C_1 is 1024 bits, too. In addition, we set the bit-length of the time stamp (TU_i), the identity number (ID_i, SID_j) and the random number (N_j) are 32 bits, 128 bits and 128 bits, respectively. The bit-length of the hash functions $h(\cdot)$ and $H(\cdot)$ are 160 bits and 512 bits, respectively. Therefore, the communication cost $\{C_1, C_2, TU_i\}$ and $\{C_5, N_j, TU_i\}$ is 1568 (1024+512+32) bits and 672 (512+128+32) bits, respectively. Therefore, the communication cost of the proposed scheme is 3264 bits, which is very small in practice. From the above analysis, we can see that the proposed scheme is well suited for mobile client server environment in term of the communication cost.

6. Conclusion

In this paper, we proposed a secure and efficient ECC-based authentication and key agreement scheme for multi-server environments and low-power mobile devices. Under the CDH problem assumption and the random oracle model, we demonstrated that the proposed scheme is secure against impersonation attack and provides the session key protection. Then, we analyze some other security features about our scheme such as replay attack and stolen smart card attack. According to the comparisons in Section 5, the proposed scheme is more efficient and practical than the other related authentication schemes, and is more suitable for multi-server environment with low-power devices.

Acknowledgements

The authors are grateful to the editor and anonymous reviewers for their valuable suggestions which improved this paper. This research was partially supported by Foundation of China and National High-Tech (863) Programs Grant No. 2011AA01A101, the National Natural Science Foundation of China under Grant No. 61300220 & 61271041, the National Major Science and Technology Special Project of China under Grant No.2012ZX03005008, and FP7 Integrated Project iCore (Internet Connected Objects for Reconfigurable Eco-systems) under Grant No. 287708.

References

[1] L. Lamport, "Password authentication with insecure communication," *Communication of ACM*, vol. 24, pp. 770-772, 1981. Article (CrossRef Link)

[2] G. Horng, "Password authentication without using password table," *Information Processing Letters*, vol. 55, no. 5, pp. 247-250, 1995. Article (CrossRef Link)

[3] J. K. Jan and Y. Y. Chen, "Paramita Wisdom password authentication scheme without verification tables," *The Journal of Systems and Software*, vol. 42, no.1, pp. 45-57, 1998. Article (CrossRef Link)
[4] Y. P. Liao and C. M. Hsiao, "A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients," *Future Generation Computer Systems*, vol. 29, no. 3, pp. 886-900, 2013. Article (CrossRef Link)
[5] D. He, J. H. Chen and J. Hu, "An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security," *Information Fusion*, vol. 13, no. 3, pp. 223-230, 2012. Article (CrossRef Link)
[6] M. S. Hwang, S. K. Chong and T. Y. Chen. "DoS-resistant ID-based password authentication scheme using smart cards," *Journal of Systems and Software*, vol. 83, no. 1, pp. 163-172, 2010. Article (CrossRef Link)
[7] R. G. Song. "Advanced smart card based password authentication protocol," *Computer Standards & Interfaces*, vol. 32, no. 5-6, pp. 321-325, 2010. Article (CrossRef Link)
[8] X. Li, J. W. Niu, M.K. Khan and J. G. Liao. "An enhanced smart card based remote user password authentication scheme," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1365-1371, 2013. Article (CrossRef Link)
[9] L. H. Li, L. C. Lin and M. S. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498-504, 2001. Article (CrossRef Link)
[10] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Transaction on Consumer Electronics*, vol. 50, no. 1, pp. 251-255, 2004. Article (CrossRef Link)
[11] C. C. Chang and J. S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards," in *Proc. of the third international conference on cyberworlds*, pp. 417-22, November 2004. Article (CrossRef Link)
[12] J. L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Computers & Security*, vol. 27, no.3-4, pp. 115-21, 2008. Article (CrossRef Link)
[13] Y. P. Liao and S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standard & Interfaces*, vol. 31, no. 1, pp. 24-29, 2009. Article (CrossRef Link)
[14] H. C. Hsiang and W. K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards & Interfaces*, vol. 31, no. 6, pp. 1118-1123, 2009. Article (CrossRef Link)
[15] C. C. Lee, T. H. Lin and R. X. Chang, "A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13863-13870, 2011. Article (CrossRef Link)
[16] S. K. Sood, A. K. Sarje and K. Singh, "A secure dynamic identity based authentication protocol for multi-server architecture," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 609-618, 2011. Article (CrossRef Link)
[17] Y. H. Chuang and Y. M. Tseng, "Towards generalized ID-based user authentication for mobile multi-server environment," *International Journal of Communication System*, vol. 25, no. 4, pp. 447-460, 2012. Article (CrossRef Link)
[18] X. Li, J. Ma, W. D. Wang, Y. P. Xiong and J. S. Zhang, "A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments," *Mathematical and Computer Modelling*, vol. 58, no. 1-2, pp. 85-95, 2013. Article (CrossRef Link)
[19] X. Li, Y. P. Xiong, J. Ma and W. D. Wang. "An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 763-769, 2012. Article (CrossRef Link)
[20] C. Li, C. Lee and C. Weng, C. Fan, "An Extended Multi-Server-Based User Authentication and Key Agreement Scheme with User Anonymity," *KSII Transactions on Internet & Information Systems*, Vol. 7 no. 1, pp. 119-131, 2013. Article (CrossRef Link)

[21] X. Li, Y. Zhang, X. Liu and J. Cao, "A Lightweight Three-Party Privacy-preserving Authentication Key Exchange Protocol Using Smart Card," *KSII Transactions on Internet & Information Systems*, Vol. 7 no. 5, pp. 1313-1327, 2013. Article (CrossRef Link)
[22] J. Nam, K.K.R. Choo, M. Kim, J. Paik and D. Won, "Dictionary Attacks against Password-Based Authenticated Three-Party Key Exchange Protocols," *KSII Transactions on Internet & Information Systems*, Vol. 7, no. 12, pp. 3244-3260, 2013. Article (CrossRef Link)
[23] M. L. Das, A. Saxena, V. P. Gulati and D. B. Phatak, "A novel remote client authentication protocol using bilinear pairings," *Computer and Security*, vol. 25, no. 3, pp. 184-189, 2006. Article (CrossRef Link)
[24] G. F. Fang and G. X. Huang, "Improvement of recently proposed Remote User Authentication Schemes," http://eprint.iacr.org/2006/200.
[25] D. Giri and P.D. Srivastava, "An improved remote client authentication protocol with smart cards using bilinear pairings," http://eprint.iacr.org/2006/274.
[26] J. Yang and C. Chang, "An ID-based remote mutual authentication with key agreement protocol for mobile devices on elliptic curve cryptosystem," *Computers and Security*, vol. 28, no. 3-4, pp. 138-143, 2009. Article (CrossRef Link)
[27] D. Hankerson, A. Menezes and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer, New York, 2004. http://dl.acm.org/citation.cfm?id=940321
[28] F. Li, X. Xin and Y. Hu, "Identity-based broadcast signcryption," *Computer Standard and Interfaces*, vol. 30, no. 1-2, pp. 89-94, 2008. Article (CrossRef Link)
[29] P. Rogaway and T. Shrimpton, "Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance," *Lecture Notes in Computer Science*, vol. 3017, pp. 371-388, 2004. Article (CrossRef Link)
[30] J. M. David, W. Matt and D. S. Michael, "Implementing Public-Key Infrastructure for Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 4, pp. 1-23, 2008. Article (CrossRef Link)
[31] M. Scott, N. Costigan and W. Abdulwahab, "Implementing cryptographic pairings on smartcards," in *Proc. of Cryptographic Hardware and Embedded Systems - CHES 2006*, LNCS, vol. 4249, pp.134-147, Springer-Verlag, 2006. Article (CrossRef Link)
[32] Shamus Software, http://www.shamus.ie/index.php.
[33] M. Bellare, D. Pointcheval and P. Rogaway, "Authenticated key agreement secure against dictionary attacks," in *Proc. of the Advances in Cryptology - EUROCRYPT 2000*, LNCS, vol. 1807, pp. 139-155, Springer-Verlag, 2000. Article (CrossRef Link)

Junsong Zhang received his master's degree in computer software and theory from Zhengzhou University (ZZU) in 2008 and Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications (BUPT) in 2014. Dr. Zhang is a lecturer of Zhengzhou University of Light Industry (ZZULI). His research interests include information security and mobile network, etc.

Jian Ma was born in 1959, obtained his Ph.D degree at Helsinki University of Technology in 1994. He is now a professor of Beijing University of Posts and Telecommunications. His research interests span the area of mobile internet, mobile internet of things, wireless sensing network and social network analysis. He has published more than 300 papers in different journals and conferences.

Xiong Li received his master's degree in mathematics and cryptography from Shaanxi Normal University (SNNU) in 2009 and Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications (BUPT) in 2012. Dr. Li now is a lecturer of Hunan University of Science and Technology (HNUST). He has published more than 15 referred journal papers. His research interests include cryptography and information security, etc.

Wendong Wang received the B.S. and M.E. degrees, both in computer science, from Beijing University of Posts and Telecommunications (BUPT), Beijing, China in 1985 and 1991, respectively. He is currently a professor of State Key Lab of Networking and Switching Technology at BUPT. His research interests are IP QoS, next generation Internet, and next generation internet services.