Two-Factor Remote Authentication Protocol with User Anonymity Based on Elliptic Curve Cryptography

L. Zhang, Shanyu Tang*, *Senior Member IEEE*, J. Chen, S. Zhu

School of Computer Science, China University of Geosciences, Wuhan, 430074, China

*Corresponding author: shanyu.tang@gmail.com, carolyn321@163.com,Tel/Fax:+862767848563

ABSTRACT

In order to provide secure remote access control, a robust and efficient authentication protocol should realize mutual authentication and session key agreement between clients and the remote server over public channels. Recently, Chun-Ta Li proposed a password authentication and user anonymity protocol by using smart cards, and they claimed that their protocol has satisfied all criteria required by remote authentication. However, we have found that his protocol cannot provide mutual authentication between clients and the remote server. To realize 'real' mutual authentication, we propose a two-factor remote authentication protocol based on elliptic curve cryptography in this paper, which not only satisfies the criteria but also bears low computational cost. Detailed analysis shows our proposed protocol is secure and more suitable for practical application.

KEY WORDS: authentication; key agreement; user anonymity; elliptic curve; smart card

1. INTRODUCTION

A secure and efficient remote authentication mechanism is needed to make clients and the remote server authenticate each other and generate a session key for future communications. Since the password is always chosen by the client freely and easy to remember, it is usually employed to achieve the authentication requirement between clients and the remote server over public channels. The first password-based remote authentication protocol was proposed by Lamport [1]. This protocol achieved good performance since only a one-way hash function was used in the protocol. Followed their work, many password-based authentication protocols have been proposed to improve the protocol's security and enhance their functionality [2-5]. However, these protocols require the server maintain a password table for verification purposes, thereby making them suffer from possible attacks such as insider attacks, password disclosure attacks, stolen-verifier attacks and server-spoofing attacks. When large number of client registers on the remote server, maintenance and protection of the verification table will become an intractable problem. Once the verification table is stolen by the adversary, the security of the system is crumbled [6]. Moreover, since the password or verification tables are usually very large, maintaining the tables and the reset passwords makes these solutions hard to scale up. Furthermore, passwords might be forgotten or divulged. So the password-based authentication protocols cannot provide high security since its security solely depends on the sensitive verification tables.

In order to improve the security, some two-factor authentication protocols have been proposed [7-12]. In the two–factor authentication protocols, smart cards are employed to store some secrets and the clients only need to remember the passwords. Since the smart cards can show what you have, and the passwords can verify what you know, through combining the password and the smart card, two different data types, these two-factor authentication protocols can provide strong security authentication. However, most of these two-factor protocols cannot resist impersonation attacks, many logged-in users' attacks, password guessing attacks, and so on. The use of smartcards also evokes some new security problems such as stolen/lost smartcard attacks, and an evicted client may use an overdue smartcard to access the server et al. [13]. Furthermore, the secrets stored in smart cards could be extracted by monitoring the power consumption and analyzing the leaked information in the smart cards [14-15].

On the basis of the above description and references [16-17], a robust and efficient remote authentication protocol should satisfy the following criteria: (1) The server needs not to maintain any security-sensitive password or verification table; (2) Provide security in choosing and updating passwords freely; (3) Provide user anonymity to protect the user's privacy; (4) Provision of mutual authentication and session key agreement to protect their future communications; (5) Provision of perfect forward secrecy; (6) Prevention of clock synchronization problem and time-delay; (7) Provision of practicability and efficiency, that is low computation and communication cost; (8) Resistance to replay, man-in-middle, and modification attacks; (9) Resistance to client impersonating and server spoofing attacks; (10)

Resistance to offline dictionary attacks with/without smartcards, password disclosure, stolen-verifier, known-key, corrupt insider and many logged-in users' attacks; (11) Resistance to parallel session and stolen/lost smart card attacks.

Recently, Chun-Ta Li [17] analyzed the authentication protocol proposed by Islam and Biswas [18], proposed an authentication protocol by using password and smartcard based on elliptic curve cryptography, and they claimed that their protocol satisfied the criteria mentioned above. However, in this paper we prove that Chun-Ta Li's protocol cannot provide mutual authentication between clients and the remote server. We also propose a robust and efficient two-factor authentication protocol with key agreement based on the elliptic curve discrete logarithm problem [19] which satisfies the above criteria at the same time. In addition, we use the GNY logic [20] to prove the security of our proposed protocol. Compared with the previous schemes [16-18], our protocol not only satisfies more security requirements, but also achieves better performance.

The rest of this paper is organized as follows. In Section 2 a brief review of Chun-Ta Li's protocol is given. Section 3 describes a cryptanalysis of Chun-Ta Li's protocol. Our authenticated key agreement protocol is presented in Section 4. In Section 5, the security of our proposed protocol is discussed. The performance of the protocol is examined in Section 6, and the paper is concluded in Section 7.

2. REVIEW OF CHUN-TA LI'S PROTOCOL

In this section, we briefly review Chun-Ta Li's password authentication protocol [17]. Their protocol consists of four phases: registration, password authentication, password change and session key distribution phase, described as follows (as shown in Figure 1):

2.1. Registration phase

When a client *A* wants to become a new legal client, it performs the following steps with the server *S*.

*Step R*1: $A \rightarrow S : ID_A, U_A$

The client *A* chooses its identity ID_A, its password pw_A and a random integer r_A. Next, it computes a password-verifier $U_A = pw_A \cdot r_A \cdot G$ and keeps r_A secretly.

*Step R*2: $S \rightarrow A : \text{SMART CARD}$

The server *S* stores each legal client's identity, password-verifier and a *status-bit* in a write protected file. Then it stores $\{G, U_S, H(\cdot), E_K(\cdot), D_K(\cdot)\}$ in a smart card and delivers this smart card to the user *A* in a secure channel.

*Step R*3: The user *A* stores r_A in the smart card. Then the smart card contains $\{r_A, G, U_S, H(\cdot), E_K(\cdot), D_K(\cdot)\}$.

2.2. Password authentication phase

When a client *A* wishes to login to the server *S*, the client *A* has to insert its smart card into the card reader and type its identity ID_A and password pw_A. And then the smart card and the server *S* perform the following steps:

*Step A*1: $A \to S: ID_A, E_{K_x}(ID_A, R_A, W_A, U'_A)$

The smart card selects a random integer r'_A, and computes $R_A = r_A U_S = r_A d_s G$, $W_A = r_A r_A PW_A G$, $U'_A = pw_A r'_A G$ and $E_{K_x}(ID_A, R_A, W_A, U'_A)$, where U_S is the public key of the server *S* and K_x is the x coordinate of $K = pw_A r_A U_S = pw_A r_A d_s G = (K_x, K_y)$. Then it sends ID_A and $E_{K_x}(ID_A, R_A, W_A, U'_A)$ to the server *S*.

*Step A*2: $S \to A: (W_A + W_S), H(W_S, U'_A)$

After receiving the message, the server *S* computes $K = d_s U_A = pw_A r_A d_S G = (K_x, K_y)$ to obtain K_x, and then it decrypts $E_{K_x}(ID_A, R_A, W_A, U'_A)$ by using K_x to reveal (ID_A, R_A, W_A, U'_A). Next, the server *S* compares the decrypted ID_A with the received ID_A and $\hat{e}(d_s R_A, U_A)$ with $\hat{e}(W_A, U_S)$. If all the conditions are satisfied, it chooses a random integer r_s and computes $W_S = r_s U_S = r_s d_s G$. Then it sends $(W_A + W_S)$ and $H(W_S, U'_A)$ to the client *A*.

*Step A*3: $A \to S: ID_A, H(W_A, W_S, U'_A)$

The client *A* obtains W_S from the received information $(W_A + W_S)$ and computes $H(W_S, U'_A)$. Next it verifies whether the computed value $H(W_S, U'_A)$ is equal to the received $H(W_S, U'_A)$. If so, the client *A* computes $H(W_A, W_S, U'_A)$ and sends it to the server *S*.

Client A		Server S
Registration Phase:		
Select ID_A, pw_A, r_A , compute $U_A = pw_A r_A G$		
	$\{ID_A, U_A\}$ $\rightarrow$	Store $\{G, U_S, H(\cdot), E_K(\cdot)/D_K(\cdot)\}$
	$\leftarrow$ SMART CARD	
Store r_A into SMART CARD		
Password Authentication Phase:		
Input ID_A, pw_A, r'_A		
Compute $R_A = r_A U_S, W_A = r_A r_A pw_A G, U'_A = pw_A r'_A G$		
	$\{ID_A, E_{K_x}(ID_A, R_A, W_A, U'_A)\}$ $\rightarrow$	Decrypt $E_{K_x}(ID_A, R_A, W_A, U'_A)$
		Verify $\hat{e}(d_S R_A, U_A) \stackrel{?}{=} \hat{e}(W_A, U_S)$
		$W_S = r_S U_S$
	$\leftarrow$ $\{(W_A + W_S), H(W_S, U'_A)\}$	
Retrieve W_S from $(W_A + W_S)$		
Verify $H(W_S, U'_A)$		
	$\{ID_A, H(W_A, W_S, W'_A)\}$ $\rightarrow$	Verify $H(W_A, W_S, W'_A)$
		Replace U_A with U'_A
	$\leftarrow$ Access Granted/Denied	
Replace r_A with r'_A		
Password Change Phase:		
Input ID_A, pw_A, pw', r'_A	$\{ID_A, E_{K_x}(ID_A, R_A, W_A, U'_A)\}$ $\rightarrow$	
	$\leftarrow$ $\{(W_A + W_S), H(W_S, U'_A)\}$	
	$\{ID_A, H(W_A, W_S, W'_A), H(W_S + W_A + U'_A)\}$ $\rightarrow$	
	$\leftarrow$ Password Change Granted/Denied	
Session Key Distribution Phase:		
Input ID_A, pw_A, pw', r'_A	$\{ID_A, E_{K_x}(ID_A, R_A, W_A, U'_A)\}$ $\rightarrow$	
	$\leftarrow$ $\{(W_A + W_S), H(W_S, U'_A, SK)\}$	
	$\{ID_A, H(W_A, W_S, U'_A, SK)\}$ $\rightarrow$	
$SK = (r_A r_A pw_A) W_S$	$\leftarrow$ Key Distribution Granted/Denied	$SK = (r_S d_S) W_A$

Fig.1. Chun-Ta Li's protocol

Step $A4$: $S \rightarrow A$: Access Granted/Denied

The server S computes $H(W_A, W_S, U'_A)$ by using its own W_S and the received information (W_A, U'_A). Then it checks whether the hashed result of $H(W_A, W_S, U'_A)$ is equal to the received $H(W_A, W_S, U'_A)$. If so, the server S grants A's login request and replaces $U_A = pw_A r_A G$ with $U'_A = pw_A r'_A G$, otherwise denies A's login request. Finally, if all the conditions are satisfied, the client A's smart card will replace r_A with r'_A.

After finishing the password authentication phase, the verifier table is updated.

2.3. Password authentication phase

When a client A wants to change its password, A must notify the server S to update the old password verifier $U_A = pw_A r_A G$ with a new password verifier $U'_A = pw'_A r'_A G$.

*Step P*1: $A \to S : ID_A, E_{K_x}(ID_A, R_A, W_A, U'_A)$

*Step P*2: $S \to A : (W_A + W_S), H(W_S, U'_A)$

*Step P*3: $A \to S : ID_A, H(W_A, W_S, U'_A), H(W_S + W_A + U'_A)$

*Step P*4: $S \to A :$ Password Change Granted/Denied

2.4. Session key distribution phase

The client A and the server S choose two random numbers $r_A, r_S \in [1, n-1]$, respectively. After above four phases, the server S replaces U_A with U'_A and A's smart card replaces r_A with r'_A. And the client A and the server S can compute the session key $SK = r_A r_A pw_A r_S d_S G$ for a new password verifier $U'_A = pw'_A r'_A G$.

*Step S*1: $A \to S : ID_A, E_{K_x}(ID_A, R_A, W_A, U'_A)$

*Step S*2: $S \to A : (W_A + W_S), H(W_S, U'_A, SK)$

*Step S*3: $A \to S : ID_A, H(W_A, W_S, U'_A, SK)$

*Step S*4: $S \to A :$ Key distribution Granted/Denied

3. CRYPTANALYSIS OF CHUN-TA LI'S PROTOCOL

In this section, we demonstrate that the protocol proposed by Chun-Ta Li [17] cannot provide mutual authentication between the clients and the remote server.

Proof

$\because R_A = r_A U_S = r_A d_S G$, $U_A = pw_A r_A G$

$\therefore \hat{e}(d_S R_A, U_A) = \hat{e}(d_S r_A d_S G, pw_A r_A G) = \hat{e}(G,G)^{r_A r_A pw_A d_S d_S}$

$\because W_A = r_A r_A pw_A G \; U_S = d_S G$

$\therefore \hat{e}(W_A, U_S) = \hat{e}(r_A r_A pw_A G, d_s G) = \hat{e}(G,G)^{r_A r_A pw_A d_S}$

$\because \hat{e}(G,G)^{r_A r_A pw_A d_S d_S} \neq \hat{e}(G,G)^{r_A r_A pw_A d_S}$

$\therefore \hat{e}(d_S R_A, U_A) \neq \hat{e}(W_A, U_S)$

So, the verification $\hat{e}(d_S R_A, U_A) = \hat{e}(W_A, U_S)$ provided by Chun-Ta Li [17] is not correct. Even if the client is a legal client, she or he cannot pass the verification of the server *S*. Under this case, the server *S* will not send any message to the legal client and will reject the legal client's login request.

Moreover, since the extended protocol with user anonymity provided by Chun-Ta Li is based on the protocol described above, the extended protocol also cannot provide mutual authentication.

According to above analysis, the protocols proposed by Chun-Ta Li cannot provide mutual authentication between clients and the server.

4. OUR PROPOSED PROTOCOL

This section presents our newly designed two-factor (password, smart card) authentication key agreement protocol with user anonymity. The proposed protocol consists of five phases: system setup phase, registration phase, pre-computation phase, authentication phase and password changing phase. The procedures of the proposed protocol are described in detail as follows (as shown in Fig. 2):

4.1. System setup phase

*Step S*1: The server *S* chooses an elliptic curve equation $E_p(a,b): y^2 = x^3 + ax + b \pmod p$ over a prime finite field F_p, where $a, b \in F_p$ and $4a^3 + 27b^2 \neq 0 \pmod p$. Then it selects a base point P over $E_p(a,b)$.

*Step S*2: The server *S* chooses a random integer $s \in_R Z_p^*$ as a secret key and computes public key P_{pub}=sP, and chooses one secure one-way hash functions $h(\cdot):\{0,1\}^* \rightarrow \{0,1\}^k$.

*Step S*3: The server *S* keeps s secret and publishes the public information $\{E_p(a,b), P, P_{pub} h(\cdot)\}$.

4.2. Registration phase

When the client U wants to register with the server *S*, it performs the following steps with the server *S*.

*Step R*1: $U \rightarrow S: (ID, h(PW \oplus r))$

The client *U* selects its password PW and identity ID freely and chooses a random integer $r \in_R Z_p^*$. Next, the client *U* computes $h(PW \oplus r)$ and sends $\{ID, h(PW \oplus r)\}$ to the server *S* over a secure channel.

*Step R*2: After receiving the message from the client *U*, the server *S* computes the secret information $R = E_s(h(PW \oplus r) \| ID)$ for the client U. Then the server *S* records ID in an identity table as shown in Table 1 which consists of two columns one is for storing the client's *ID* and the other *Status* is used for checking whether the login *ID* is registered, revoked or not. Next it stores *R* in the memory of a smart card and delivers the smart card to the client *U* in a secure manner.

*Step R*3: Upon receiving the smart card, the client *U* stores the secret random integer *r* in the smart card. Then the memory of the smart card contains (R, r). The client *U* keeps the password PW, ID and the smart card secretly for the registration process.

Table 1. Identity table

Identity	Status
ID_1	0/1
ID_2	0/1
……	……
ID_i	0/1
……	……

4.3. Authentication phase

In the authentication phase, the smart card and the server S perform the following four steps.

*Step A*1: $U \rightarrow S : (W, Z)$

The client U first inputs its password PW and its identity ID. It then selects a random integer $r_1 \in_R Z_p^*$, and computes $V = h(PW \oplus ID \oplus r)r_1P$, $W = h(PW \oplus ID \oplus r)r_1P_{pub}$ and $Z = E_{V_{co\_x}}(R \| h(PW \oplus r) \| ID)$, where $E_{V_{co\_x}}$ is an encryption function with the x-coordinate of elliptic curve point V. Finally, it sends (W, Z) to the server S.

*Step A*2: $S \rightarrow U : (X, r_3)$

After receiving the message, the server S computes $V^* = s^{-1}W = s^{-1}h(PW \oplus ID \oplus r)r_1sP$ $= h(PW \oplus ID \oplus r)r_1P$ by using the server's secret key s. It then obtains the x-coordinate of elliptic curve point V^* denoted as $V^*_{co\_x}$ and decrypts Z by using it to get information $R, h(PW \oplus r)$ and ID. Next, the server S checks whether ID is valid according to the identity table. If not, terminate the authentication session. Otherwise, it decrypts R by using its secret key s to obtain $h(PW \oplus r)$ and ID. Then the server S compares the value of the ID in Z with that of the ID in R to make sure that the message was indeed sent by the client U. Next it checks whether the value $h(PW \oplus r)$ in Z is equal to the value $h(PW \oplus r)$ getting from R. If so, it generates two random integers $r_2, r_3 \in_R Z_p^*$, and computes the session key $SK = h(V^*_{co\_x} \| r_2)$ and authentication message $Auth_s = h(V^*_{co\_x} \| r_3)$. Then the server S encrypts the secret random integer r_2 and authentication message $Auth_s$ by using $h(h(PW \oplus r) \oplus ID)$. At last it sends the encrypted message $X = E_{h(h(PW \oplus r) \oplus ID)}(Auth_s \| r_2)$ and random integer r_3 to the client U.

*Step A*3: $U \rightarrow S : (Auth_u)$

Upon receiving the message, the client U inputs its password PW and identity ID to compute the decryption key $h(h(PW \oplus r) \oplus ID)$. And then it can decrypt the message X to obtain the authentication message $Auth_s$ and the integer r_2. After that the client can compute the session key $SK' = h((h(PW \oplus ID \oplus r)r_1P)_{co\_x} \| r_2)$. Then the client U checks whether the equation $Auth_s \stackrel{?}{=} h(V_{co\_x} \| r_3)$ holds. If the equation holds, it computes $Auth_u = h(SK' \| (r_3 + 1))$ and sends $Auth_u$ to the server S. Otherwise, it deletes the received information and the protocol stops.

*Step A*4: After receiving the response message, the server S verifies whether the following equation holds $Auth_u \stackrel{?}{=} h(SK \| (r_3+1))$. If the message is authenticated, the server S sets SK as the shared session key with the client U ; otherwise, it deletes the received information and the protocol stops.

Client A		Server S
Registration Phase:		
Select $ID, PW, r \in_R Z_q^*$, compute $h(PW \oplus r)$		
	$\{ID, h(PW \oplus r)\}$ $\longrightarrow$	
		Compute $R = E_s(h(PW \oplus r) \| ID)$
	$\longleftarrow$ SMART CARD (R)	
Store r into SMART CARD		
Authentication Phase:		
Input ID, PW and choose $r_1 \in_R Z_q^*$		
Compute $V = h(PW \oplus ID \oplus r)r_1P, W = h(PW \oplus ID \oplus r)r_1sP$		
$Z = E_{V_{co\_x}}(R \| h(PW \oplus r) \| ID)$		
	$\{W, Z\}$ $\longrightarrow$	$V^* = s^{-1}W = h(PW \oplus ID \oplus r)r_1P$
		Decrypt Z by using $V_{co\_x}^*$
		Check ID in the identity table
		Decrypt R by using s
		Compare two IDs and two values of $h(PW \oplus r)$
		Select $r_2, r_3 \in_R Z_q^*$
		Compute $SK = h(V_{co\_x}^* \| r_2)$
		$Auth_s = h(V_{co\_x}^* \| r_3)$
	$\longleftarrow$ $\{X, r_3\}$	$X = E_{h(h(PW \oplus r) \oplus ID)}(Auth_s \| r_2)$
Decrypt X by using $h(h(PW \oplus r) \oplus ID)$		
Compute $SK' = h((h(PW \oplus ID \oplus r)r_1P)_{co\_x} \| r_2)$		
$Auth_s \stackrel{?}{=} h(V_{co\_x} \| r_3)$		
Compute $Auth_u = h(SK' \| (r_3+1))$		
	$\{Auth_u\}$ $\longrightarrow$	$Auth_u \stackrel{?}{=} h(SK \| (r_3+1))$
Password Change Phase:		
Select $PW^*, N, r^* \in_R Z_q^*$		
Compute $E_{SK}(h(PW^* \oplus r^*) \| ID \| N \| h(h(PW^* \oplus r^*) \| ID \| N))$		
	$\{N, E_{SK}(h(PW^* \oplus r^*) \| ID \| N \| h(h(PW^* \oplus r^*) \| ID \| N))\}$ $\longrightarrow$	Verify the tag $h(h(PW^* \oplus r^*) \| ID \| N)$
		Compute $R^* = E_s(h(PW^* \oplus r^*) \| ID)$
	$\longleftarrow$ $E_{SK}(R^* \| h(N+1 \| R^*))$	$E_{SK}(R^* \| h(N+1 \| R^*))$
Verify the tag $h(N+1 \| R^*)$		
Store (R^*, r^*) into SMART CARD		

Fig.2. The proposed protocol

4.4. Password changing phase

If the client U wants to change its password PW to a new password PW^* , it needs to agree on a session key with the server S via the authentication phase in advance. And the client U must notify the server S to update the old secret $R = E_s(h(PW \oplus r)\|ID)$ with a new secret $R^* = E_s(h(PW^* \oplus r^*)\|ID)$. All steps of the password changing phase are executed as follows:

*Step P*1: The client U selects a new random integer $r^* \in_R Z_q^*$, a new password PW^* and a nonce N for freshness checking. Then it encrypts the new password message $h(PW^* \oplus r^*)\|ID$ by using the session key SK . Next, it sends $E_{SK}(h(PW^* \oplus r^*)\|ID\|N\|h(h(PW^* \oplus r^*)\|ID\|N))$ and N to the server S.

*Step P*2: After receiving the message, the server S uses the session key SK to decrypt the encryption message and verifies whether the authentication tag $h(h(PW^* \oplus r^*)\|ID\|N)$ is valid. If true, it computes a new secret value $R^* = E_s(h(PW^* \oplus r^*)\|ID)$ and sends $E_{SK}(R^*\|h(N+1\|R^*))$ to the client U . Otherwise, it rejects the password changing requirement.

*Step P*3: Upon receiving the message, the client U decrypts the message and verifies whether the authentication tag $h(N+1\|R^*)$ is valid. If so, the client U stores (R^*, r^*) in its smart card. Otherwise, it stops the password updating.

5. SECURITY ANALYSIS

5.1 Authentication proof based on Gong-Needham-Yahalom logic

In this section, first we introduce some formulae and statements that will be used in the Gong-Needham-Yahalom (GNY) logic; and then set the goals and list the assumptions of the protocol; finally we use the GNY logic to prove our proposed protocol. GNY logic has been widely used to formally analyze the completeness of authentication protocols. It has successfully found defects in several protocols and disclosed redundancies in many protocols. So in this paper, we adopt the GNY logic to evaluate the security of the proposed protocol.

5.1.1 Formulae and statements

A formula is a name used to refer to a bit string, which has a particular value in a run [20]. Let *X* and *Y* range over formulae. First we introduce some formulae used in our authentication proof and the complete list of all logical postulates is shown in [20].

(1) (*X*, *Y*): conjunction of two formulae *X* and *Y*.

(2) $\{X\}_K$ and $\{X\}_K^{-1}$: symmetrically encrypt and decrypt *X* with the key *K*.

(4) *h*(*X*): a one-way function of *X*.

(5) **X*: *X* is not originated here.

A basic statement reflects some property of a formula [20]. Let *P* and *Q* range over principals. Then we introduce some statements as follows:

(1) $P \lhd X$: *P* is told formula *X*.

(2) $P \ni X$: *P* possesses formula *X*.

(2) $P|\sim X$: *P* once conveyed formula *X*.

(3) $P|\equiv \#(X)$: *P* believes that *X* is fresh.

(4) $P|\equiv \phi(X)$: *P* believes that *X* is recognizable.

(5) $P \xleftrightarrow{S} Q$: *P* believes that *S* is a suitable secret for *P* and *Q*.

5.1.2 Protocol description

We first change some notations to fit the GNY logic and transform our protocol into the form of P→ *Q*:(*X*). In the following transformation, the notation *x_V* is regarded as V_{co-x} in our protocol.

(1) $U \to S : (h(PW \oplus ID \oplus r)r_1 sP, \{R \| h(PW \oplus r) \| ID\}_{x\_h(PW \oplus ID \oplus r)r_1 P})$

(2) $S \to U : (\{h(x\_h(PW \oplus ID \oplus r)r_1 P \| r_3) \| r_2\}_{h(h(PW \oplus r) \oplus ID)}, r_3)$

(3) $U \to S : (h(h(x\_h(PW \oplus ID \oplus r)r_1 P \| r_2) \| (r_3 + 1)))$

5.1.3 Goals

We descript our goals as follows:

(1) Message content authentication

Goal 1: *S* believes that the message in the first run is recognizable.

$S|\equiv \phi\{h(PW \oplus ID \oplus r)r_1 sP, \{R \| h(PW \oplus r) \| ID\}_{x\_h(PW \oplus ID \oplus r)r_1 P}\}$

Goal 2: U believes that the message in the second run is recognizable.

$$U \mid\equiv \phi\{\{h(x\_h(PW \oplus ID \oplus r)r_1P \| r_3) \| r_2\}_{h(h(PW \oplus r) \oplus ID)}, r_3\}$$

Goal 3: S believes that the message in the third run is recognizable.

$$S \mid\equiv \phi\{h(h(x\_h(PW \oplus ID \oplus r)r_1P \| r_2) \| (r_3 + 1))\}$$

(2) Message origin authentication

Goal 4: U believes S conveyed the message in the second run.

$$U \mid\equiv S \mid\sim \{\{h(x\_h(PW \oplus ID \oplus r)r_1P \| r_3) \| r_2\}_{h(h(PW \oplus r) \oplus ID)}\}$$

Goal 5: S believes U conveyed the message in the third run.

$$S \mid\equiv U \mid\sim \{h(h(x\_h(PW \oplus ID \oplus r)r_1P \| r_2) \| (r_3 + 1))\}$$

(3) Session key material establishment

Goal 6: U believes that S believes that r_2 is a secret shared between U and S.

$$U \mid\equiv S \mid\equiv U \xleftrightarrow{SK} S$$

Goal 7: U believes that r_2 is a secret shared between U and S.

$$U \mid\equiv U \xleftrightarrow{SK} S$$

Goal 8: S believes that U possesses SK.

$$S \mid\equiv U \ni SK$$

Goal 9: S believes that U believes that SK is a secret shared between U and S.

$$S \mid\equiv U \mid\equiv U \xleftrightarrow{SK} S$$

5.1.4 Assumption list

To derive our goals, some assumptions are made as follows:

(1) S possesses the secret key s.

$$S \ni s$$

(2) Since S keeps the identity table, S believes that ID is recognizable.

$$S \mid\equiv \phi(ID)$$

(3) The random integer r_1 and r are generated by U in the protocol, so U possesses r_1 and r. The password PW and ID are chosen by U in the protocol and are keep secret, therefore U possesses PW and ID and believes that ID is fresh. Moreover, the point P is public, and then U possesses P.

$U \ni r_1, U \ni r, U \ni PW, U \ni P, U \ni ID, U \mid\equiv \#(ID)$

(4) We assume that U believes $h(h(PW \oplus r) \oplus ID)$ is a suitable secret for himself and S.

$U \mid\equiv U \xleftrightarrow{h(h(PW \oplus r) \oplus ID)} S$

(5) The SK generated by S is a temporal session key in the current run. So we assume that S believes that SK is a suitable secret between U and S.

$S \mid\equiv U \xleftrightarrow{r_2} S$

(6) The random integer r_2 and r_3 are generated by S in the protocol, so S possesses r_2, r_3 and believes that r_2 and r_3 are fresh.

$S \ni r_2, S \mid\equiv \#(r_2)$, $S \ni r_3, S \mid\equiv \#(r_3)$

(7) U believes that the server S is an authority on generating a suitable session key material SK shared between U and S.

$U \mid\equiv S \mid\Rightarrow U \xleftrightarrow{SK} S$

5.1.5 Authentication proof by using Gong-Needham-Yahalom (GNY) logic

In this subsection, we use the GNY logic to analyze the proposed protocol. A complete list of all logical postulates and the index in the list is provided [20], such as ($T1$, $P1$), to show how to achieve the goals defined in subsection 5.1.3.

(1) The first run:

$$\frac{S \lhd\ h(PW \oplus ID \oplus r) r_1 sP, S \ni s}{S \ni x\_h(PW \oplus ID \oplus r) r_1 P} \quad (P1, P2, P3) \qquad (1)$$

If S is told a formula $h(PW \oplus ID \oplus r) r_1 sP$ and possesses a key s (Assumption 1), then it is consider to possess the formula $h(PW \oplus ID \oplus r) r_1 P$, as well as possess x-V.

$$\frac{S \mid\equiv \phi(ID), S \ni s}{S \mid\equiv \phi\{h(PW) \oplus r, ID\}_s} \quad (R1, R2) \qquad (2)$$

If S believes that ID is recognizable (Assumption 2) and possesses the secret key s (Assumption 1), then S believes that the encrypted message $\{h(PW \oplus r), ID\}_s$ with ID as a component in it, is recognizable. So, S can recognize the secret $R = E_s(h(PW \oplus r) \| ID)$ in the smart card of the client U.

$$\frac{S|\equiv\phi(R), S \ni x\_V}{S|\equiv\phi\{R, h(PW \oplus r), ID\}_{x\_V}} \quad (R1, R2) \qquad (3)$$

If S believes that R is recognizable (A2) and possesses the key x_V (A1), then S believes that the encrypted $\{R, h(PW \oplus r), ID\}_{x\_V}$ with R as a component in it, is recognizable. So, S can recognize $\{R, h(PW \oplus r), ID\}_{x\_V}$.

$$\frac{S|\equiv\phi(\{R, h(PW \oplus r), ID\}_{x\_V})}{S|\equiv\phi(h(PW \oplus ID \oplus r)r_1 sP, \{R, h(PW \oplus r), ID\}_{x\_V})} \quad (R1) \qquad (4)$$

If S believes $\{R, h(PW \oplus r), ID\}_{x\_V}$ is recognizable (A3), then it believes that $(h(PW \oplus ID \oplus r)r_1 sP, \{R, h(PW \oplus r), ID\}_{x\_V})$ of which $\{R, h(PW \oplus r), ID\}_{x\_V}$ is a component, is recognizable.

According to A4, in the proposed protocol, the server S can recognize the message $\{h(PW \oplus ID \oplus r)r_1 sP, \{R \| h(PW \oplus r) \| ID\}_{x\_h(PW \oplus ID \oplus r)r_1 P}\}$ in the first run. (Goal 1)

(2) The second run:

$$\frac{U \ni PW, U \ni ID, U \ni r}{U \ni h(PW \oplus ID \oplus r), U \ni h(PW \oplus r) \oplus ID, U \ni h(h(PW \oplus r) \oplus ID)} \quad (P2, P4) \qquad (5)$$

If U possesses PW, ID and r (Assumption 3), it is capable of possessing $h(PW \oplus ID \oplus r)$, $h(PW \oplus r) \oplus ID$ and $h(h(PW \oplus r) \oplus ID)$.

$$\frac{U \ni h(PW \oplus ID \oplus r), U \ni r_1, U \ni P}{U \ni x\_V} \quad (P2, P3) \qquad (6)$$

If U possesses $h(PW \oplus ID \oplus r)$ (A5), P and r_1 (Assumption 3), it is capable of possessing x_V.

$$\frac{U \ni x\_V}{U|\equiv\phi(x\_V, r_3)} \quad (P4, R6, R1) \qquad (7)$$

If U possesses x_V (A6), then U is entitled to believe that (x_V, r_3) is recognizable.

$$\frac{U|\equiv\phi(x\_V, r_3), U \ni x\_V, U \lhd r_3}{U|\equiv\phi(Auths)} \quad (P1, P2, R5) \qquad (8)$$

If U believes that (x_V, r_3) is recognizable (A7) and U possesses (x_V, r_3) ($P2$, since U is told r_3 and possesses x_V (A6), then it possesses (x_V, r_3)), then U is entitled to believe that $Auths = h(V_{co-x} \| r_3)$ is recognizable.

$$\frac{U \mid\equiv \phi(Auths), U \ni h(h(PW \oplus r) \oplus ID)}{U \mid\equiv \phi(\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)}), U \mid\equiv \phi(\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)}, r_3)} \quad (R1, R2) \qquad (9)$$

If U believes that the formula *Auths* (8) is recognizable and U possesses the key $h(h(PW \oplus r) \oplus ID)$ (5), then U is entitled to believe that $\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)}$, of which *Auths* is a component, is recognizable. And U also believe that ($\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)}$, r_3) of which $\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)}$ is a component is recognizable.

So, according to (9) we can say that in the proposed protocol, U can recognize the message $\{\{h(x\_h(PW \oplus ID \oplus r)r_1 P \| r_3) \| r_2\}_{h(h(PW\oplus r)\oplus ID)}, r_3\}$ in the second run. (Goal 2)

$$\frac{U \mid\equiv \#(ID), P \ni h(PW \oplus r) \oplus ID}{U \mid\equiv \# h(h(PW \oplus r) \oplus ID)} \quad (F1, F10) \qquad (10)$$

If U believes that ID (Assumption 3) is fresh then it believes that $h(PW \oplus r) \oplus ID$ is fresh. Since U also possesses $h(PW \oplus r) \oplus ID$ (A5), then it is entitled to believe that $h(h(PW \oplus r) \oplus ID)$ is fresh.

$$\frac{U \mid\equiv \phi(Auths), U \ni h(h(PW \oplus r) \oplus ID), U \mid\equiv \# h(h(PW \oplus r) \oplus ID)}{U \mid\equiv \#\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)}} \quad (F1, F7) \qquad (11)$$

If U believes that the formula *Auths* is recognizable (8) and U possesses the key $h(h(PW \oplus r) \oplus ID)$ (A5) and believes that $h(h(PW \oplus r) \oplus ID)$ is fresh (10), and then U is entitled to believe that the encrypted formula $\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)}$, of which *Auths* is a component, is fresh. Therefore, U can identify that the message in the second run of the proposed protocol is fresh.

$$\frac{\begin{array}{c} U < *\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)}, U \ni h(h(PW \oplus r) \oplus ID), \\ U \mid\equiv U \xleftrightarrow{h(h(PW\oplus r)\oplus ID)} S, U \mid\equiv \phi(Auths, r_2), U \mid\equiv \# h(h(PW \oplus r) \oplus ID) \end{array}}{U \mid\equiv S \mid\sim \{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)}, U \mid\equiv S \ni h(h(PW \oplus r) \oplus ID)} \quad (I1) \qquad (12)$$

If all of the following conditions hold: 1) U receives the formula (*Auths*, r_2) encrypted with the key $h(h(PW \oplus r) \oplus ID)$ and marked with a not-originated-here mark; 2) U possesses $h(h(PW \oplus r) \oplus ID)$ (5); 3) U believes that $h(h(PW \oplus r) \oplus ID)$ is a suitable secret for himself and S (Assumption 4); 4) U believes that the formula (*Auths*, r_2) is recognizable (9); and 5) U believes that $h(h(PW \oplus r) \oplus ID)$ is fresh (10).

Then U is entitled to believe that 1) S once conveyed (*Auths*, r_2) encrypted with $h(h(PW \oplus r) \oplus ID)$ and 2) U believes that the server S possesses $h(h(PW \oplus r) \oplus ID)$ (Goal 4)

According to the GNY logic, we assume that $U|\equiv S|\Rightarrow S|\equiv *$, that is, U believes that S is honest and competent, and then we can deduce the following statement:

$$\frac{U|\equiv S|\Rightarrow S|\equiv *, U|\equiv S|\sim(\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)} \sim> S|\equiv U \xleftrightarrow{SK} S), U|\equiv \#\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)}}{U|\equiv S|\equiv U \xleftrightarrow{SK} S} \quad (J2)$$

(13)

If U believes that S is honest and competent; and U receives a message $(\{Auths, r_2\}_{h(h(PW\oplus r)\oplus ID)} \sim> S|\equiv U \xleftrightarrow{SK} S)$ (Assumption 5), which it believes S conveyed (12), then U ought to believe that S really believes $U \xleftrightarrow{SK} S$.

According to (13), U believes that S believes that SK is a suitable secret between U and S. (Goal 6)

$$\frac{U|\equiv S|\Rightarrow U \xleftrightarrow{SK} S, U|\equiv S|\equiv U \xleftrightarrow{SK} S}{U|\equiv U \xleftrightarrow{SK} S} \quad (J1) \qquad (14)$$

If U believes that S is an authority on the statement $U \xleftrightarrow{SK} S$ (Assumption 7) and S believe in $U \xleftrightarrow{SK} S$ (13), then U ought to believe in $U \xleftrightarrow{SK} S$ as well.

According to (14), U believes that SK is a suitable secret between U and S. (Goal 7)

(3) The third flow:

$$\frac{S \ni r_2}{S|\equiv \phi(x\_V, r_2)} \quad \text{(Assumption 6, } P4, R6, R1) \qquad (15)$$

If S possesses r_2 (Assumption 6) then it believes that r_2 is recognizable, and (x_V, r_2) is recognizable.

$$\frac{S|\equiv \phi(x\_V, r_2), S \ni (x\_V, r_2)}{S|\equiv \phi(h(x\_V, r_2))} \quad \text{(A15, A1, Assumption 6, } P2, R5) \qquad (16)$$

If S believes that (x_V, r_2) is recognizable (15) and it also possesses (x_V, r_2), then S believes that h(x_V, r_2) is recognizable.

$$\frac{S|\equiv \phi(h(x\_V, r_2)), S \ni (x\_V, r_2), S \ni r_3}{S|\equiv \phi h(h(x\_V, r_2), (r_3+1))} \quad \text{(A16, } R1, \text{A1, Assumption 6, } P2, P4, R5) \qquad (17)$$

If S believes that h(x_V, r_2) is recognizable (16) , then it is entitled to believe that (h(x_V, r_2),(r_3+1)) is recognizable (R1). And if S possesses (x_V, r_2) (1, Assumption 6, P2) and r_3 (Assumption 6), then it is capable of possessing (h(x_V, r_2),(r_3+1)) (P4, P2) . So, according to R5, S is entitled to believe that h(h(x_V, r_2),(r_3+1)) is recognizable.

According to (17), we can say that S believes that the message $\{h(h(x\_h(PW \oplus ID \oplus r)r_1P \| r_2) \| (r_3+1))\}$ in the third run is recognizable. (Goal 3)

$$\frac{S \ni x\_V, S \ni r_2}{S \ni SK} \text{ (A1, Assumption 6, } P2, P4) \qquad (18)$$

If S possesses x_V ($A1$) and r_2 (Assumption 2), it is capable of possessing (x_V, r_2), and so it is capable of possessing $SK = h(x\_V \| r_2)$.

$$\frac{S \ni SK, S \ni r_3}{S \ni ((r_3+1), SK)} \text{ (}A18\text{, Assumption 6, } P2) \qquad (19)$$

If S possesses SK ($A18$) and r_3 (Assumption 6), then it is capable of possessing ((r_3+1), SK).

$$\frac{S | \equiv \#(r_2)}{S | \equiv \#(SK)} \text{ (Assumption 6, } F1) \qquad (20)$$

If S believes r_2 is fresh (Assumption 6) then it is entitled to believe that SK is fresh.

$$\frac{S \lhd *h((r_3+1), <SK>), S \ni ((r_3+1), SK), S | \equiv S \xleftrightarrow{SK} U, S | \equiv \#(SK)}{S | \equiv U | \sim ((r_3+1), <SK>), S | \equiv U | \sim h((r_3+1), <SK>)} \quad (I3) \qquad (21)$$

If all of the following conditions hold: 1) S receives a formula $Auth_u = h(SK \| (r_3+1))$ consisting of a one way function of (r_3+1) and SK marked with a not-originated-here mark; 2) S possesses (r_3+1) and SK (19); 3) S believes SK is a suitable secret for himself and U (Assumption 5); 4) U believes that SK is fresh (20). Then S is entitled to believe that U once conveyed ((r_3+1), SK) and h((r_3+1), SK).

According to (21), we can say that S believes that the message $Auth_u$ in the third run of the proposed protocol is conveyed from the U. (Goal 5)

$$\frac{S | \equiv U | \sim ((r_3+1), SK), S | \equiv \#(SK)}{S | \equiv U \ni SK} \quad (I6, I7, P3) \qquad (22)$$

If S believes that U once conveyed the formula ((r_3+1), SK) (21), then it is entitled to believe that U once conveyed SK ($I7$). Since S believes that SK is fresh (20) and U once conveyed SK, then S is entitled to believe that U possesses SK.

According to (22), S believes that SK is possessed by U. (Goal 8)

According to the GNY logic, we assume that $U | \equiv S | \Rightarrow S | \equiv *$, that is, S believes that U is honest and competent, and then we can deduce the following statement:

$$\frac{S | \equiv U | \Rightarrow U | \equiv *, S | \equiv U | \sim (Auth_u \leadsto U | \equiv U \xleftrightarrow{SK} S), S | \equiv \#(Auth_u)}{S | \equiv U | \equiv U \xleftrightarrow{SK} S} \quad (J2) \qquad (23)$$

If S believes that U is honest and competent, and S receives a message $Auth_u \sim> U | \equiv U \xleftrightarrow{SK} S$ which it believes U conveyed (21), then S ought to believe that U really believes $U \xleftrightarrow{SK} S$.

According to (23), we can say that in the proposed protocol, S believes that SK is a suitable secret between U and S. (Goal 9)

5.2 Discussion on possible attacks

5.2.1 Replay attacks

Supposing that an adversary Bob intercepts the client U's previous message (W, Z) in *Step A*1 and replays it to the server S to impersonate the client U. However, in *Step A*3， Bob cannot construct a valid $Auth_u$ to pass the verification process of the server S unless he can correctly guess the session key $SK = h(V_{co-x} \| r_2)$. In order to construct a valid $Auth_u$, Bob needs to extract V from the intercepted message W, which is equivalent to solving an elliptic curve discrete logarithm problem without the knowledge of the server's secret key s. Moreover, Bob cannot obtain V by decrypting the intercepted message Z since it is protected by a secure symmetric encryption algorithm. Furthermore, Bob cannot correctly guess the random integer r_2 from the intercepted message X without the knowledge of the client's password PW, the client's identifier ID and a secret random integer r. So, Bob cannot generate a valid $Auth_u$ to pass the verification process of the server S in *Step A*4. Therefore, the proposed protocol can resist the replay attacks.

5.2.2 Man-in-the-middle attacks

In the proposed protocol, a session key SK can be shared only after mutual authentication between the client U and the server S. So, if an adversary Bob attempts to make the server S believe that it is talking to the client U, he need to pass the verification process of the server S. But Bob cannot pass the verification without the knowledge of the client U's password, client's identifier ID and the secret random integer r. On the other hand, for the same reason, Bob cannot construct a valid $Auth_s$ to pass the verification process of the client U. So, Bob cannot cheat the client U to share a session key and make the client U believe that the key is shared with the server S. Therefore, Bob cannot launch the man-in-middle attack to cheat either the client U or the server S.

5.2.3 Impersonation attacks

Assuming an adversary Bob sends a fraud message (*W*', Z') to the server *S* to impersonate the client. But, Bob cannot construct a valid *R* without the knowledge of the server's secret key *s*. Therefore, the server *S* can easily find this attack by comparing the value of the *ID* in *Z* with that of the *ID* in *R*. Moreover, the server S can also detect this attack by checking whether the value $h(PW \oplus r)$ in *Z* is equal to the value $h(PW \oplus r)$ getting from *R*.

Supposing an adversary Bob modifies the message (*X*', r_3') and sends it to the client *U* to impersonate the server *S*. But Bob cannot correctly guess the decryption key $h(h(PW \oplus r) \oplus ID)$ and construct a valid authentication message $Auth_s$ to pass the verification process of the client *U*. Therefore the client *U* can easily detect the attack by checking whether the equation $Auth_s \overset{?}{=} h(V_{co-x} \| r_3)$ is hold.

If an adversary Bob forges an authentication message $Auth_u$' and sends it to the server *S* to impersonate the client *U*. The server *S* can find out that $Auth_u$' is not equivalent to its computed value, since Bob cannot correctly guess the value of *SK* without the knowledge of the client *U*'s password *PW*, identity *ID* and the secret random integer (r, r_1) or the server's secret key *s*. Therefore, the proposed protocol can resist the modification attacks.

5.2.4 Stolen-verifier attacks

In the proposed protocol, no password or verification tables are stored on the server *S*. Therefore, the protocol can resist the stolen-verifier attacks.

5.2.5 Offline dictionary attacks without the smart card

Suppose an adversary intercepts all messages relay between the client *U* and the server *S* through eavesdropping communications and then carrying out the offline dictionary attack. In order to obtain the client *U*'s password PW, Bob needs to extract $h(PW \oplus ID \oplus r)$ from $W = h(PW \oplus ID \oplus r)r_1sP$, which is equivalent to solving an instance of elliptic curve discrete logarithm problem. Even if Bob can obtain the value $h(PW \oplus ID \oplus r)$, he cannot correctly guess the client *U*'s password, since the entropy of the secret random integer *r* is very large. So Bob cannot launch the offline dictionary attack by using the intercepted message *W*. Furthermore, since the client *U*'s password *PW* is protected by a secure symmetric encryption algorithm and a high entropy random integer *r*, so Bob cannot derive it from the information Z, X or $Auth_u$. Therefore, the proposed protocol can withstand against the offline dictionary attack without the smart card.

5.2.6 Offline dictionary attacks with the smart card

Assuming an adversary Bob compromises the secret information (R, r) stored in the smart card of the client U and intercepts all the messages transmitted between the client U and the server U. Compared with the offline dictionary attack without the smart card, the addition information known by Bob in this attack is (R, r). However, without the knowledge of the server's secret key s, Bob cannot obtain $h(PW \oplus r)$ from R and know whether each of their guessed passwords is correct or not. Therefore, the proposed protocol can resist the offline dictionary attack with the smart card.

5.2.7 Insider Attacks

In the proposed protocol, the server S needs not to store a password or verification table, so a privileged-insider of the server S cannot access other servers by stealing the identity and password-verifier from the server S's verification table. Therefore, the proposed protocol can resist the insider attacks.

5.2.8 Many logged-in users' attacks

Assuming that the client U's password PW and the identity ID are leaked to more than one adversary. At this case, when adversaries try to login the server S using the client U's password and identity at the same time, the server S will find the attacks by checking the client U's login status *Status* stored in the identity table. Therefore, the adversaries cannot launch many logged-in users' attacks successfully.

5.2.9 Password disclosure attacks

In our protocol, in the register phase, the client U sends $h(PW \oplus r)$ instead of its password to the server S, since the password is protected by a high entropy random integer r chosen by the client U and keep secret, so the server S cannot find an opportunity to obtain the client's password in the register phase. Therefore, the proposed protocol can resist the password disclosure attacks.

5.2.10 Provide session key security

In the proposed protocol, only the client U and the server S can compute the session key $SK = h_1((h(PW \oplus ID \oplus r)r_1P)_{co\_x} \| r_2)$ at the end of the key exchange. If the adversary Bob attempts to obtain the session key *SK*, he needs to extract $h(PW \oplus ID \oplus r)r_1P$ from the intercepted message *W*, which is equivalent to solving an instance of elliptic curve discrete logarithm problem. Furthermore, Bob cannot correctly guess the random integer r_2 without the knowledge of the client's password *PW*, the client's identifier *ID* and a secret random integer *r* or the server's secret key *s*. So, the session key *SK* is not known by anyone but only the client *U* and the server *S*. Therefore, the proposed protocol provides session key security.

5.2.11 Provide known-key security

In the proposed protocol, the session key *SK* of each session is not connected with the session keys of any other sessions, since the random numbers $r_1$and r_2 generated independently by the client *U* and the server *S* are different in each session process. Even if the adversary Bob compromises a session key *SK*, he cannot compute other session keys $SK = h_1((h(PW \oplus ID \oplus r)r_1^{'}P)_{co\_x} \| r_2^{'})$. This is because the high entropy random integers $r_1^{'}$ and $r_2^{'}$ are differ in every session. Therefore, the proposed protocol provides the known-key security.

5.2.12 Provide perfect forward secrecy

In the proposed protocol, the long-term private key of the client *U* is its password *PW*. Supposing that the adversary Bob compromises the client's password *PW* and intercepts all messages relays between the client *U* and the server *S*. But knowing above information is not enough for computing a previous session key *SK*, because Bob cannot correctly guess the valid random integer r_2 and compute a correct $h(PW \oplus ID \oplus r)r_1P$. When Bob tries to extract the value $h(PW \oplus ID \oplus r)r_1P$ from the message *W*, he will face an elliptic curve discrete logarithm problem. Furthermore, Bob cannot correctly guess the random integer r_2 without the knowledge of the client's identifier *ID* and the high entropy random integer *r*. Therefore, in the proposed protocol, even if the client's password *PW* is compromised, the secrecy of previous session keys established by them cannot be affected.

5.2.13 Provide mutual authentication

In the proposed protocol, the server *S* and the client *U* authenticate each other by checking $Auth_u$ and $Auth_s$, respectively. Therefore, the proposed protocol can provide mutual authentication.

5.2.14 Provide security in choosing and updating passwords

In order to helps users to remember their own passwords, the legitimate clients with the smartcards can freely choose their favorite passwords in the proposed protocol. Furthermore, an update password phase for users to change their password is also provided in the proposed protocol. Even if the smart card has lost or has been stolen, any other person cannot change or update the password, since they do not know the current session key *SK* shared between the client *U* and the server *S*.

5.2.15 Provide user anonymity

In the proposed protocol, the anonymity of the client *U* is obtained by hash function, symmetric encryption technique and elliptic curve discrete logarithm problem. In the proposed protocol, the client's identifier *ID* is protected by a secure symmetric encryption algorithm, elliptic curve discrete logarithm problem. So, even if an adversary Bob compromises the secret (R, r) stored in the smartcard and record the used messages transmitted between the client *U* and the server *S*, he cannot derive the real identifier *ID* of the client without the knowledge of secret key *s*.

6. COMPLEXITY ANALYSIS

In this section, we first evaluate the security and summarize the functionality of our protocol, and then compare the computational cost of our protocol with other related protocols.

As shown in Table 2, compared with [17] and [18], our proposed protocol can provide more unique features such as user anonymity and no password or verifier table. These new features are very important in implementing a practical and universal authenticated key agreement. Moreover, Table 2 also shows that the proposed protocol is more robust and secure than other related protocols. In the proposed protocol, since the server does not need to store any password table, the adversary cannot launch the stolen-password attack and the server spoofing attack successfully. The proposed protocol can also resist the attacks associated with client's identity, because the client's real identity *ID* is protected by hash function, symmetric encryption technique and elliptic curve discrete logarithm problem, so the adversary cannot guess the client's real identity correctly. In addition, the proposed protocol is secure against other attacks mentioned in Section 5, such as the password disclosure attack, the insider attack, the impersonation attack, the many logged-in users' attack, the stolen smartcard attack, et al.

Table 2. Security and functionality comparisons between our protocol and other protocols

	Wamg-Juang-Lei [16]	Islam-Biswas [18]	Chun-Ta [17]	Our protocol
No password or verifier table	√	×	×	√
Freely choosing and updating the password	√	√	√	√
User anonymity	√	×	×	√
Secure mutual authentication	√	√	×	√
Session key agreement	√	√	×	√
Secure to password disclosure attacks	√	×	N/A	√
Secure to password guessing attacks	√	×	N/A	√
Secure to server spoofing attacks	×	√	N/A	√
Secure to stolen-verifier attacks	√	×	N/A	√
Secure to insider attacks	√	×	N/A	√
Secure to impersonation attacks	√	×	N/A	√
Secure to many logged-in user's attacks	×	√	N/A	√
Secure to stolen smart card	√	N/A	N/A	√

Next, we discuss the computational costs of the proposed protocol in each phase. In the registration phase, one hash operations is required to compute $h(PW \oplus r)$ on the client side, and one symmetric key encryption operation is needed to obtain *R* on the server side. In the authentication phase, the client takes two scalar multiplication operations of elliptic curve to compute $V = h(PW \oplus ID \oplus r)r_1P$ and $W = h(PW \oplus ID \oplus r)r_1P_{pub}$; one symmetric key encryption operations to compute *Z*; one symmetric key decryption operation to decrypt *X*; and five one-way hash function operations to compute $h(h(PW \oplus r) \oplus ID)$, $h(PW \oplus ID \oplus r)$, *SK*, $Auth_u$ and $Auth_s$. The server takes one scalar multiplication operation of elliptic curve and one modular inversion operation to get V^*; one symmetric key encryption operations to compute *X*; two symmetric key decryption operation to decrypt *Z* and *R*; and four one-way hash function operations to compute $h(h(PW \oplus r) \oplus ID)$, *SK*, $Auth_u$ and $Auth_s$. In the password changing phase, the client takes three one-way hash function operations to compute $h(PW^* \oplus r^*), h(h(PW^* \oplus r^*) \| ID \| N)$ and $h(N+1 \| R^*)$; one symmetric key encryption operation and one symmetric key decryption operation. The server takes two one-way hash function operations to compute $h(h(PW^* \oplus r^*)\|ID\|N)$ and $h(N+1\|R^*)$; and one symmetric key decryption operation and two symmetric key encryption operation.

For the convenience of evaluating the computational cost, some notations are defined as follows:

(1) T_b : is the time for executing a bilinear map operation;

(2) T_m : is the time for executing a scalar multiplication operation of elliptic curve;

(3) T_a : is the time for executing a point addition operation of elliptic curve;

(4) T_h : is the time for executing a one-way hash function;

(5) T_{inv} : is the time for executing a modular inversion operation;

(6) T_e : is the time for executing a symmetric key encryption operation;

(7) T_d : is the time for executing a symmetric key decryption operation.

Experimental results [21–23] show that the time for executing a symmetric key encryption/decryption operation T_e/T_d, a bilinear pairing operation T_b, and a scalar multiplication operation of elliptic curve Tm are 0.0087, 0.38 and 0.13 s respectively. And the time for performing a point addition operation of elliptic curve T_a, a modular inversion operation T_v, and a hash function operation T_h are less than 0.1, 0.03, and 0.001 s respectively. According to Table 3, the total required computational time of Islam-Biswas's protocol, Chun-Ta's protocol, and our protocol are 1.3624, 1.5224 and 0.3038 s at the client side respectively, and 2.4924, 2.3924 and 0.2269 s at the server side respectively. Close analysis of the data in Table 3, shows that our proposed protocol is more efficient than Islam-Biswas's protocol [18] and Chun-Ta's protocol [17], because it eliminates the expansive operation bilinear pairing operations and reduces the numbers of the operations of scalar multiplication of elliptic curve.

Table 3. Computational comparisons between our protocol and others

	Computational cost	Islam-Biswas [18]	Chun-Ta [17]	Our protocol
Registration phase	Client	$1T_m \approx 0.13$ s	$1T_m \approx 0.13$ s	$1T_h \approx 0.001$ s
	Server			$1T_e \approx 0.0087$ s
Authentication phase (password authentication & Session key distribution)	Client	$4T_m+1T_a+1T_e+2T_h \approx$ 0.6307 s	$5T_m+1T_a+1T_e+2T_h \approx$ 0.7607 s	$2T_m+1T_e+1T_d+5T_h \approx$ 0.2824 s
	Server	$2T_b+3T_m+1T_a+1T_d+2T_h \approx$ 1.2607 s	$2T_b+3T_m+1T_a+1T_d+2T_h$ ≈ 1.2607 s	$1T_m+1T_{inv}+1T_e+2T_d+4T_h$ ≈ 0.1901 s

Password changing phase	Client	$3T_m+2T_a+1T_e+3T_h \approx$ 0.6017 s	$4T_m+1T_a+1T_e+3T_h \approx$ 0.6317 s	$1T_e+1T_d+3T_h \approx$ 0.0204 s
	Server	$2T_b+2T_m+2T_a+1T_d+3T_h \approx$ 1.2317 s	$2T_b+2T_m+1T_a+1T_d+3T_h \approx$ 1.1317 s	$2T_e+1T_d+2T_h \approx$ 0.0281 s
Overall	Client	$8T_m+3T_a+2T_e+5T_h \approx$ 1.3624 s	$10T_m+2T_a+2T_e+5T_h \approx$ 1.5224 s	$2T_m+2T_e+2T_d+9T_h \approx$ 0.3038 s
	Server	$4T_b+5T_m+3T_a+2T_d+5T_h \approx$ 2.4924 s	$4T_b+5T_m+2T_a+2T_d+5T_h \approx$ 2.3924 s	$1T_m+1T_{inv}+4T_e+3T_d+6T_h \approx$ 0.2269 s

From above discussion, we can conclude that our proposed protocol not only satisfies all the criteria required by mutual authentication and key agreement protocol, but also reduces the computational cost to an extent.

7. CONCULSION

In this paper, we have found that Chun-Ta Li's protocol cannot provide mutual authentication between the client and the server, and then we have proposed a robust and efficient authentication protocol based on elliptic curve cryptography by using the password and the smart card. The proposed protocol satisfies all the criteria aforementioned which are very important for mutual authentication and key agreement by using the password and the smart card. The security analysis has proved the proposed protocol can resist all possible attacks, and the performance analysis has shown that our protocol achieves better performance that other related protocols. So the proposed protocol is more suitable for practical application.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant Numbers 61303237, 61272469, 61075063), the Fundamental Research Funds for the Central Universities (Grant Number 2013199037), Wuhan Scientific Research Program (Grant Number 2013010501010144) and China Postdoctoral Fund (Grant Number 2012194091).

REFERENCES

1. Lamport, L. (1981). Password authenticationwith insecure communication. Communications of the ACM, 24(11), 770–772.

2. Peyravian,M.,&Zunic, N. (2006).Methods for protecting password transmission. Computers&Security, 19(2006), 466–469.

3. Peyravian, M., & Jeffries, C. (2006). Secure remote user access over insecure networks. Computer Communications, 29(5), 660–6673.

4. Chang, C.-C., Lee, C.-Y., & Chiu, Y.-C. (2009). Enhanced authentication scheme with anonymity for roaming service in global mobility networks. Computer Communications, 32(4), 611–618.

5. Rhee, H. S., Kwon, J. O., & Lee, D. H. (2009). A remote user authentication scheme without using smartcards. Computer Standards & Interfaces, 31(1), 6–13.

6. Juang, W.-S., & Nien, W.-K. (2008). Efficient password authenticated key agreement using bilinear pairings'. Mathematical and Computer Modelling, 47(2008), 1238–1245.

7. Liu, J.-Y., Zhou, A.-M., & Gao, M.-X. (2008). A new mutual authentication scheme based on nonce and smart cards. Computer Communications, 31(10), 2205–2209.

8. Wang, X. M., Zhang, W. F., Zhang, J. S., & Khan, M. K. (2007). Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards. Computer Stander & Interface, 29, 507–512.

9. Wang, Y., Liu, J., Xiao, F., & Dan, J. (2009). A more efficient and secure dynamic ID-based remote user authentication scheme. Computer Communications, 32(4), 583–585.

10. Chen, T.-H., Hsiang, H.-C., & Shih, W.-K. (2011). Security enhancement on an improvement on two remote user authenticaiton schemes using smartcards. Future Generation Computer Systems, 27(2011), 377–380.

11. Khan,M. K., Kim, S.-K.,&Alghathbar, K. (2011).Cryptanalysis and security enhancement of a 'more efficent & secure dynamic ID-based remte user authencitaion scheme'. Computer Communications, 34(2011), 305–309.

12. Zhang, L., Tang, S., & Cai, Z. (2013). Efficient and flexible password authenticated key agreement for VoIP session initiation protocol using smart card. International Journal of communication systems. doi:10.1002/dac.2499.

13. Fan, C.-I., & Lin, Y.-H. (2009). Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics. IEEE Transations on Information Forensics and Security, 4(4), 933–945.

14. Kocher, P., Jaffe, J., & Jun, B. (1999). Differential power analysis'. In Proceedings of advances in cryptology, CRYPTO'99 (pp. 388–397).

15. Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (2002). Examining smart card security under the threat of power analysis attacks. IEEE Transactions on Computers, 51(5), 541–552.

16. Wang, R.-C., Juang,W.-S.,&Lei, C.-L. (2011).Robust authtication and key agreement scheme preserving the privacy of secret key. Computer Communications, 34(2011), 274–280.

17. Li, C.-T. (2012). A new password authentication and user anonymity scheme based on elliptic curvecryptography and smart card. IET Inforamtion Security. doi:10.1049/ie-ifs.2012.0058.

18. Islam, S. H., & Biswas, G. P. (2013). Design of improved password authentication and update scheme based on elliptic curve cryptography. Mathematical and Computer Modelling, 57(11–12), 2703–2717.

19. Koblitz, N., Menezes, A., & Vanstone, S. (2000). The state of elliptic curve cryptography'. Designs, Codes and Cryptography, 19(2), 173–193.

20. Gong, L., Needham, R., & Yahalom, R. (1990). Reasoning about belif in cryptographic protcocols. In Proceedings of the 1990 IEEE computer society symposium research in security and privacy (pp. 234–246).

21. Scott, M., Costigan, N., & Abdulwahab,W. (2006). Implementing cryptographic pairings on smartcards. In Proceedings of the eighth workshop on cryptographic hardware and embedded systems. Yokohama, Japan (pp. 134–147).

22. Lee, J., & Chang, C. (2007). Secure communications for cluster-based ad hoc networks using node identities. Journal of Network and Computer Applications, 30(4), 1377–1396.

23. Li, C., Hwang, M., & Chung, Y. (2008). A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks. Computer Communication, 31, 2803–2814.