ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2014, T. 43, Nr. 4

An Efficient Password Authentication Scheme Using Smart Card Based on Elliptic Curve Cryptography

Yuanyuan Zhang

School of Mathematics and Statistics, Wuhan University, Wuhan, China
e-mail: circle0519@hotmail.com

Jianhua Chen

School of Mathematics and Statistics, Wuhan University, Wuhan, China
e-mail: chenjh_ecc@163.com

Baojun Huang

School of Mathematics and Statistics, Wuhan University, Wuhan, China
e-mail: huangbj_ecc@163.com

Cong Peng

NO. 722 research institute of CSIC
e-mail: licavier@163.com

crossref http://dx.doi.org/10.5755/j01.itc.43.4.6579

Abstract. Recently, Li proposed a new password authentication and user anonymity scheme based on elliptic curve cryptography. In this paper, we will show that Li's scheme is vulnerable to the impersonation attack and the denial of service attack. Moreover, we also point out that there is an error in his scheme. To overcome the weaknesses of Li's scheme, we proposed an efficient password authentication scheme based on elliptic curve cryptography. The proposed scheme improves the security and efficiency of the authentication process.

Keywords: password authentication; elliptic curve cryptography; smart card; attack.

1. Introduction

Communication networks have brought convenience to people, along with the increase of the security problems. An attacker could crack the remote servers, eavesdrop communication content, modify authentication messages and impersonate identities. User authentication is the essential security mechanism for remote login systems in which a password-based authentication scheme is the most commonly used technique to provide authentication between the clients and the server [1-7].

In 1981, Lamport [8] proposed the first remote authentication scheme based on the passwords. From then on, a series of authentication schemes [9-14] have been proposed to improve system security and computation efficiency in recent two decades. However, most of them still have security problems.

In 2003, Lin and Hwang proposed a password authentication scheme with secure password updating. But Islam and Biswas [15] showed that Lin and Hwang's [16] scheme suffers from insider attack, impersonation attack, known session-specific temporary information attack, many logged-in users' attack and stolen-verifier attack. So an improved password authentication scheme was proposed by Islam and Biswas to overcome the weaknesses of Lin and Hwang's scheme. The security of Islam and Biswas's scheme is based on elliptic curve cryptography, collision-resistant one-way hash functions and symmetric key cryptosystem. However, Li pointed out that Islam and

Biswas's scheme is vulnerable to offline password guessing attack, stolen-verifier attack and insider attack [17]. In 2012, Li proposed a new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card to improve Islam and Biswas's scheme. Nevertheless, we find that Li's scheme has some security loopholes. In this paper, we will demonstrate these corresponding attacks and propose an efficient password authentication scheme.

The rest of this paper is organized as follows: Section 2 reviews the scheme of Li, and Section 3 analyses the weaknesses of this scheme. Section 4 presents our efficient password authentication scheme. The security proof is introduced in Section 5. Section 6 analyses the security of the proposed scheme and compares the performance of our scheme with Li's scheme. Finally, Section 7 presents our conclusions.

2. Review of Li's scheme

Li proposed an advanced smart card-based password authentication and update scheme with user anonymity. The scheme consists of six phases: registration phase, password authentication phase, password change phase, session key distribution phase, user eviction phase and user anonymity phase. Fig. 1 shows the entire protocol structure of Li's scheme.

We first introduce the notations throughout this paper as follows:

- A : the client
- S : the remote server
- ID_A : the identity of client A
- U_A : the password-verifier of client A
- pw_A : the password of client A
- G : bases point of the elliptic curve group of order n such that $n \cdot G = 0$, where n is a large prime number
- U_S : the public key of S, where $U_S = d_S \cdot G$
- d_S : the secret key, which is kept secret and only known by S
- K_x : secret key computed either using $K = d_S \cdot U_A = (Kx, Ky)$ or $K = pw_A \cdot r_A \cdot U_S = (Kx, Ky)$
- $E_K(\cdot)/D_K(\cdot)$: the symmetric encryption/decryption function with key K
- $h(\cdot)$: one-way hash function
- SK : the session key
- $\hat{e}(\cdot,\cdot)$: a bilinear pairing
- $+/-$: elliptic curve point addition/subtraction

2.1. Registration phase

During the registration phase, a client A requests to be a legal user and the server does the following operations:

R1. $A \rightarrow S$: ID_A, U_A

A registers to the server with his/her identity ID_A and password-verifier $U_A = pw_A \cdot r_A \cdot G$, where r_A is a secret random number.

R2. $S \rightarrow A$: $G, U_S, h(\cdot), E_K(\cdot)/D_K(\cdot)$

After receiving the message from A, S stores A's identity, password-verifier and a *status-bit* in a write protected file as depicted in Table 1. Then, S issues a smart card which contains $\{G, U_S, h(\cdot), E_K(\cdot)/D_K(\cdot)\}$ and sends it to A through a secure channel.

R3. A enters r_A into his/her smart card and the smart card contains $\{r_A, G, U_S, h(\cdot), E_K(\cdot)/D_K(\cdot)\}$.

Table 1. Verifier table of S after finishing the registration phase

Identity	Password-verifier	*Statu-bit*
ID_A	$U_A = pw_A \cdot r_A \cdot G$	0/1
ID_B	$U_B = pw_B \cdot r_B \cdot G$	0/1
ID_C	$U_C = pw_C \cdot r_C \cdot G$	0/1
……	……	……

2.2. Password authentication phase

When A wants to access the server, he/she should perform the following steps:

PA1. $A \rightarrow S$: $ID_A, E_{Kx}(ID_A, R_A, W_A, U'_A)$

A inserts the smart card into the card reader. Then, he/she inputs the identity ID_A and the password pw_A. The smart card generates a new random number r'_A, computes $R_A = r_A \cdot U_S = r_A \cdot d_S \cdot G$, $W_A = r_A \cdot r_A \cdot pw_A \cdot G$, $U'_A = pw_A \cdot r'_A \cdot G$ and $E_{Kx}(ID_A, R_A, W_A, U'_A)$, where encryption key K_x is the x coordinate of $K = pw_A \cdot r_A \cdot U_S = pw_A \cdot r_A \cdot d_S \cdot G = (Kx, Ky)$. Then, A sends ID_A and $E_{Kx}(ID_A, R_A, W_A, U'_A)$ to S.

PA2. $S \rightarrow A$: $(W_A + W_S), h(W_S, U'_A)$

S achieves the decryption key Kx by computing $K' = d_S \cdot U_A = pw_A \cdot r_A \cdot d_S \cdot G = (K_x, K_y)$, and decrypts $E_{Kx} = (ID_A, R_A, W_A, U'_A)$ to reveal (ID_A, R_A, W_A, U'_A). Then S checks whether the revealed ID_A equals the received ID_A and whether $\hat{e}(d_S \cdot R_A, U_A)$ equals $\hat{e}(W_A, U_S)$. If all of those hold, S generates a random number r_S, computes $W_S = r_S \cdot U_S = r_S \cdot d_S \cdot G$, $(W_S + W_S)$ and $h(W_S, U'_A)$. Then, S sends $(W_A + W_S)$ and $h(W_S, U'_A)$, to A.

PA3. $A \rightarrow S$: $ID_A, h(W_A, W_S, U'_A)$

A achieves W_S by subtracting W_A from $(W_A + W_S)$ and computes $h(W_S, U'_A)$. Then, A checks whether the hashed result of (W_S, U'_A) equals the received $h(W_S, U'_A)$. If holds, A computes (W_A, W_S, U'_A) and sends it with ID_A to S.

PA4. $S \rightarrow A$: Access Granted/Denied

S computes $h(W_A, W_S, U'_A)$ by its own W_S and (W_A, U'_A) which is received from A in PA1. Then

S checks whether the hashed result of (W_A, W_S, U_A') equals the received $h(W_A, W_S, U_A')$. If this holds, S grants A's login request and replaces old password-verifier U_A with new password-verifier $U_A' = pw_A \cdot r_A' \cdot G$, otherwise denies A's login request. Finally, A's smart card replaces r_A with r_A' if all of the conditions are satisfied.

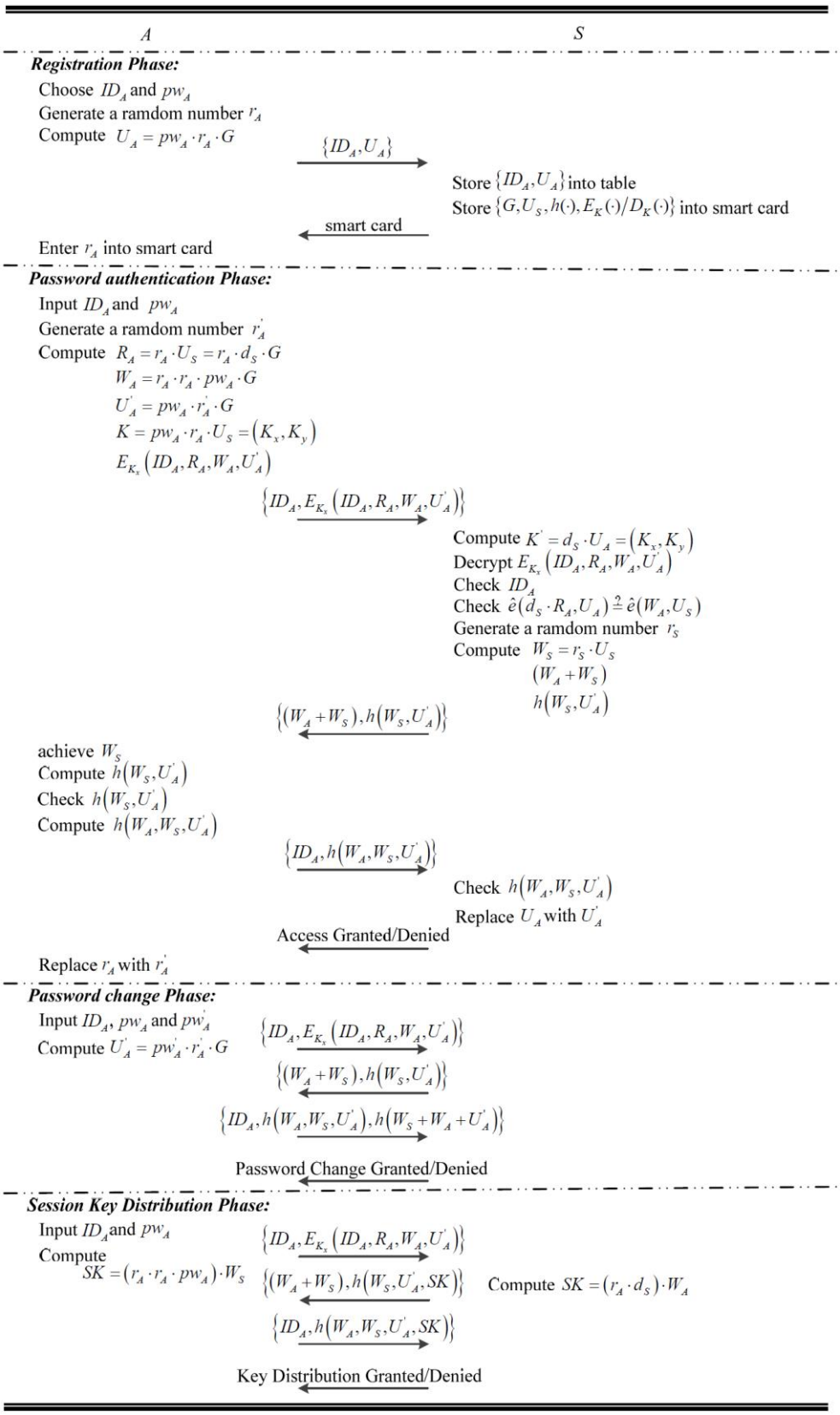

Figure 1. Li's scheme

After finishing the password authentication phase, the verifier table is updated and the content of the verifier table is shown in Table 2.

Table 2. Verifier table of S after finishing the password authentication phase

Identity	Password-verifier	*Statu-bit*
ID_A	$U'_A = pw_A \cdot r'_A \cdot G$	0/1
ID_B	$U'_B = pw_B \cdot r'_B \cdot G$	0/1
ID_C	$U'_C = pw_C \cdot r'_C \cdot G$	0/1
……	……	……

2.3. Password change phase

When A wants to change his/her password pw_A to a new password pw'_A, he/she must perform the following steps:

PC1. $A \rightarrow S$: $ID_A, E_{Kx}(ID_A, R_A, W_A, U'_A)$

PC2. $S \rightarrow A$: $(W_A + W_S), h(W_S, U'_A)$

PC3. $A \rightarrow S$: $ID_A, h(W_A, W_S, U'_A), h(W_S + W_A + U'_A)$

PC4. $S \rightarrow A$: Password Change Granted/Denied

In this phase, U'_A is equal to $pw'_A \cdot r'_A \cdot G$. In PC4, if the messages $h(W_A, W_S, U'_A)$ and $h(W_S + W_A + U'_A)$ from A are valid, S updates U_A with U'_A. Moreover, A's smart card replaces r_A with r'_A.

After finishing the password change phase, the verifier table is updated and the content of the verifier table is shown in Table 3.

Table 3. Verifier table of S after finishing the password change phase

Identity	Password-verifier	*Statu-bit*
ID_A	$U'_A = pw'_A \cdot r'_A \cdot G$	0/1
ID_B	$U'_B = pw'_B \cdot r'_B \cdot G$	0/1
ID_C	$U'_C = pw'_C \cdot r'_C \cdot G$	0/1
……	……	……

2.4. Session key distribution phase

S1. $A \rightarrow S$: $ID_A, E_{Kx}(ID_A, R_A, W_A, U'_A)$

S2. $S \rightarrow A$: $(W_A + W_S), h(W_S, U'_A, SK)$

S3. $A \rightarrow S$: $ID_A, h(W_A, W_S, U'_A, SK)$

S4. $S \rightarrow A$: Key Distribution Granted/Denied

In this phase, S computes the session key $SK = (r_S \cdot d_S) \cdot W_A$ and A computes the session key $SK = (r_A \cdot r_A \cdot pw_A) \cdot W_S$. Obviously, $SK = (r_S \cdot d_S) \cdot W_A = (r_A \cdot r_A \cdot pw_A) \cdot W_S$. In S4, if the message $h(W_A, W_S, U'_A, SK)$ from A is valid, S updates U_A with U'_A, where $U'_A = pw_A \cdot r'_A \cdot G$. Moreover, A's smart card replaces r_A with r'_A.

2.5. User eviction phase

If the server wants to evict a client A, S can delete (ID_A, U_A), form its verifier table and A cannot use (ID_A, U_A), to login S since ID_A cannot be found in the verifier table in the password authentication phase.

2.6. Provision of user anonymity

Li extends a user anonymity phase to provide the user anonymity. In this phase, the client's identity and location cannot be traced by attackers over the networks.

2.6.1. Registration phase

In this phase, A registers to the server with his/her password-verifier $U_A = pw_A \cdot r_A \cdot G$ and personal credentials, for example, National ID card. S generates a pseudonym IND_A, issues a smart card which contains $\{IND_A, G, U_S, h(\cdot), E_K(\cdot)/D_K(\cdot)\}$ and sends it to A through a secure channel. Afterward, A enters r_A into his/her smart card and the smart card contains $\{r_A, IND_A, G, U_S, h(\cdot), E_K(\cdot)/D_K(\cdot)\}$.

After finishing the registration phase, the content of the verifier table is shown in Table 4.

Table 4. Verifier table of S after finishing the registration phase

Identity	Password-verifier	*Statu-bit*
IND_A	$U_A = pw_A \cdot r_A \cdot G$	0/1
IND_B	$U_B = pw_B \cdot r_B \cdot G$	0/1
IND_C	$U_C = pw_C \cdot r_C \cdot G$	0/1
……	……	……

2.6.2. User anonymity phase

When a client A wants to access the server anonymously, he/she should perform the following steps:

U1. $A \rightarrow S$: $IND_A, E_{Kx}(IND_A, R_A, W_A, U'_A)$

A inserts the smart card into the card reader and inputs the password pw_A. The smart card computes R_A, W_A, U'_A, K_x and $E_{Kx}(IND_A, R_A, W_A, U'_A)$. Then, A sends IND_A and $E_{Kx}(IND_A, R_A, W_A, U'_A)$ to S.

U2. $S \rightarrow A$: $(W_A + W_S)$, $h(W_S, U'_A, SK, IND'_A)$, $E_{SK}(IND'_A)$

If IND_A, R_A and W_A are valid, S generates a new pseudonym IND'_A and a random number r_s, computes a session key $SK = (r_S \cdot d_S) \cdot W_A$ and sends $(W_A + W_S$, $h(W_S, U'_A, SK, IND'_A)$, and $E_{SK}(IND'_A)$ to A.

U3. $A \rightarrow S$: $IND_A, h(SK, IND'_A)$

A achieves W_S by subtracting W_A form $(W_A + W_S)$ and computes the session key $SK = (r_A \cdot$

$r_A \cdot pw_A) \cdot W_S$ to decrypt $E_{SK}(IND'_A)$. Then, A checks whether the hashed result of (W_S, U'_A, SK, IND'_A) equals the received $h(W_S, U'_A, SK, IND'_A)$. If this holds, A computes $h(SK, IND'_A)$, and sends it with IND'_A to S.

U4. $S \rightarrow A$: User Anonymity Granted/Denied

S checks whether the hashed result of (SK, IND'_A) equals the received $h(SK, IND'_A)$. If this holds, S grants A's anonymously login request and replaces (IND_A, U_A) with (IND'_A, U'_A), then A's smart card replaces (r_A, U_A) with (r'_A, U'_A), otherwise S denies A's anonymously login request.

After finishing the user anonymity phase, the content of the verifier table is shown in Table 5.

Table 5. Verifier table of S after finishing the user anonymity phase

Identity	Password-verifier	*Statu-bit*
IND_A	$U'_A = pw_A \cdot r'_A \cdot G$	0/1
IND_B	$U'_B = pw_B \cdot r'_B \cdot G$	0/1
IND_C	$U'_C = pw_C \cdot r'_C \cdot G$	0/1
……	……	……

3. Cryptanalysis of Li's scheme

Li claimed that their scheme had several security properties. However, we find that their scheme is flawed against impersonation attack and denial of service attack. Moreover, we also find that the verification $\hat{e}(d_S \cdot R_A, U_A) = \hat{e}(W_A, U_S)$ is not correct.

3.1. Attacks on Li's scheme

Because the PA1, PA2 in password authentication phase are similar to the PC1, PC2 in password change phase, Li's scheme is flawed against impersonation attack and denial of service attack when a legal client A wants to change his/her password pw_A to a new password pw'_A.

A1. $A \rightarrow Z \rightarrow S$: $ID_A, E_{Kx}(ID_A, R_A, W_A, U'_A)$

A sends ID_A and $E_{Kx}(ID_A, R_A, W_A, U'_A)$ to S, where $U'_A = pw'_A \cdot r'_A \cdot G$. The attacker Z intercepts the message from A and sends it as a login request to S.

A2. $S \rightarrow Z \rightarrow A$: $(W_A + W_S), h(W_S, U'_A)$

S sends $(W_A + W_S)$ and $h(W_S, U'_A)$ to Z. After receiving the message from S, Z sends it to A.

A3. $A \rightarrow Z$: $ID_A, h(W_A, W_S, U'_A), h(W_S + W_A + U'_A)$

$Z \rightarrow S$: $ID_A, h(W_A, W_S, U'_A)$

A sends ID_A, $h(W_A, W_S, U'_A)$ and $h(W_S + W_A + U'_A)$ to S. Z intercepts the message from A and only sends ID_A, $h(W_A, W_S, U'_A)$ to S.

A4. $S \rightarrow Z$: Access Granted

$Z \rightarrow A$: Password Change Granted/Denied.

Obviously, the message from Z is valid, S grants Z's login request and replaces old password-verifier U_A with new password-verifier U'_A. So, there is an impersonation attack in Li's scheme. In addition, if Z grants A's password change request, A's smart card replaces r_A with r'_A and A cannot notice the attack has happened only if he/she wants to login in the server before Z exiting the server. Because the server's verifier table and A's smart card are updated, Z can login in S using the new random number r'_A. But, if Z denies A's password change request, A's smart card would not replace r_A with r'_A. So A no longer can login in S, and the scheme is flawed against denial of service attack.

3.2. Error in Li's scheme

Li claimed that $\hat{e}(d_S \cdot R_A, U_A)$ equals $\hat{e}(W_A, U_S)$. A bilinear pairing was used to assure the correctness of their scheme and was given below:

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

However, we find that the verification $\hat{e}(d_S \cdot R_A, U_A) = \hat{e}(W_A, U_S)$ is erroneous.

In Li's scheme,

$R_A = r_A \cdot U_S = r_A \cdot d_S \cdot G$, and

$U_A = pw_A \cdot r_A \cdot G$,

so $\hat{e}(d_S \cdot R_A, U_A)$

$= \hat{e}(d_S \cdot r_A \cdot d_S \cdot G, pw_A \cdot r_A \cdot G)$

$= \hat{e}(G, G)^{r_A \cdot r_A \cdot pw_A \cdot d_S \cdot d_S}$

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

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

For the reason given above, the scheme proposed by Li contains an error.

4. Our proposed scheme

This section proposes a new efficient password authentication scheme based on elliptic curve cryptography using smart card. We present our proposed scheme in six phases: registration phase, password authentication phase, password change phase, session key distribution phase, user eviction phase and user anonymity phase. Fig. 2 shows the entire protocol structure of our scheme. The proposed scheme is described as follows.

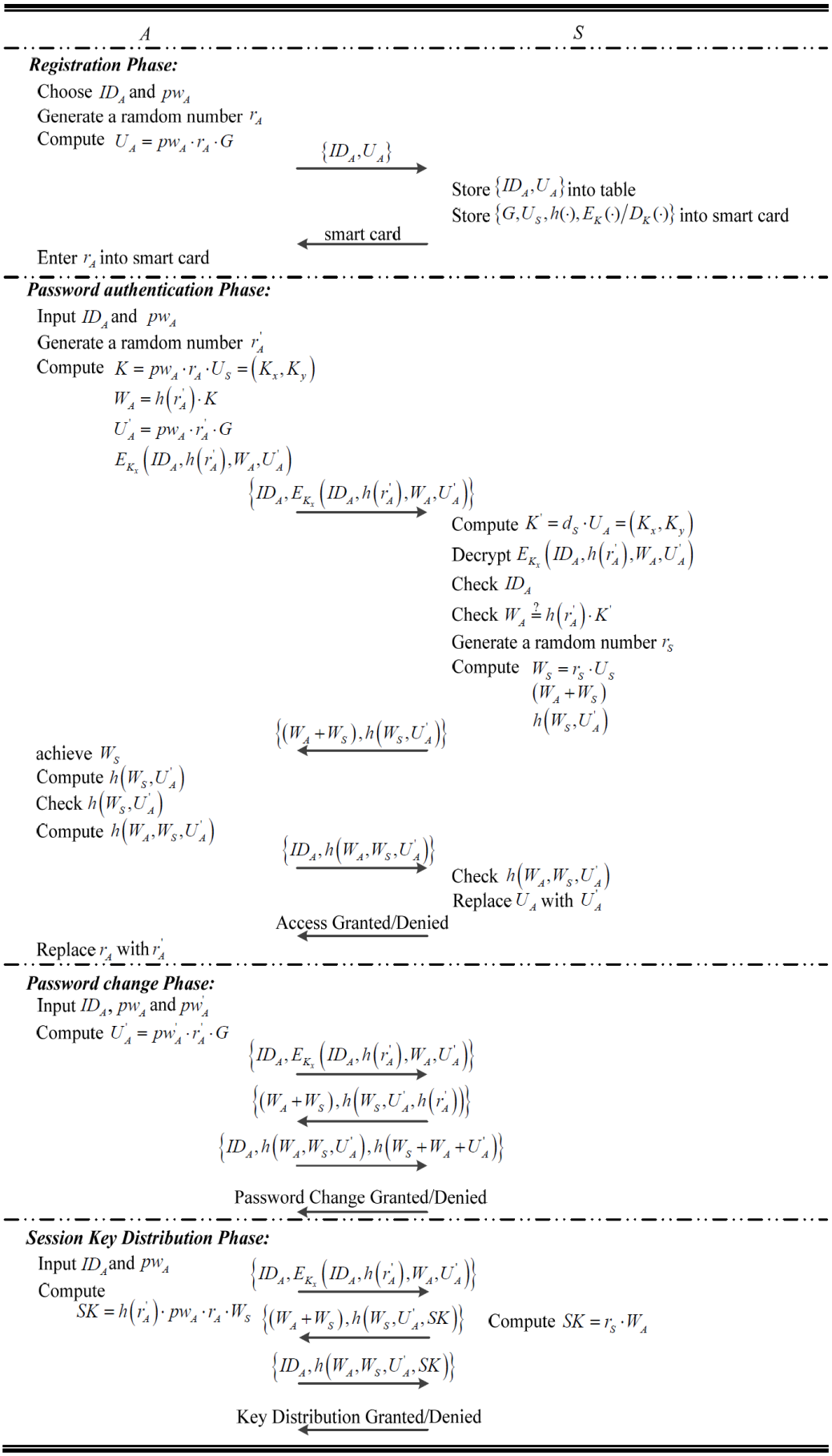

Figure 2. Our proposed scheme

4.1. Registration phase

During the registration phase, a client A requests to be a legal user and the server does the following operations:

R1. $A \rightarrow S: ID_A, U_A$

A registers to the server with his/her identity ID_A and password-verifier $U_A = pw_A \cdot r_A \cdot G$, where r_A is a secret random number.

R2. $S \rightarrow A: G, U_S, h(\cdot),\ E_K(\cdot)/D_K(\cdot)$

After receiving the message from A, S stores A's identity, password-verifier and a *status-bit* in a write protected file as depicted in Table 1. Afterward, S issues a smart card which contains $\{G, U_S, h(\cdot),\ E_K(\cdot)/D_K(\cdot)\}$ and sends it to A through a secure channel.

R3. A enters r_A into his/her smart card and the smart card contains $\{r_A, G, U_S, h(\cdot), E_K(\cdot)/D_K(\cdot)\}$.

4.2. Password authentication phase

When A wants to access the server, he/she must perform the following steps:

PA1. $A \rightarrow S: ID_A, E_{Kx}(ID_A, h(r_A'), W_A, U_A')$

A inserts the smart card into the card reader, then inputs the identity ID_A and the password pw_A. The smart card generates a new random number r_A', computes $K = pw_A \cdot r_A \cdot U_S = pw_A \cdot r_A \cdot d_S \cdot G = (Kx, Ky)$, $W_A = h(r_A') \cdot K$, $U_A' = pw_A \cdot r_A' \cdot G$ and $E_{Kx}(ID_A, h(r_A'), W_A, U_A')$. Then, A send ID_A and $E_{Kx}(ID_A, h(r_A'), W_A, U_A')$ to S.

PA2. $S \rightarrow A: (W_A + W_S), h(W_S, U_A')$

S achieves the decryption key Kx by computing $K' = d_S \cdot U_A = pw_A \cdot r_A \cdot d_S \cdot G = (Kx, Ky)$ and decrypts $E_{Kx}(ID_A, h(r_A'), W_A, U_A')$ to reveal $(ID_A, h(r_A'), W_A, U_A'$. Then S checks whether the revealed ID_A equals the received ID_A and whether $h(r_A' \cdot K')$ equals W_A. If all of those hold, S generats a random number r_S, computes $W_S = r_S \cdot U_S = r_S \cdot d_S \cdot G$, $(W_A + W_S)$ and $h(W_S, U_A')$. Then, S sends $(W_A + W_S)$ and $h(W_S, U_A')$ to A.

PA3. $A \rightarrow S: ID_A, h(W_A, W_S, U_A')$

A achieves W_S by subtracting W_A, form $(W_A + W_S)$ and computes $h(W_S, U_A')$. Then, A checks whether the hashed result of (W_S, U_A') equals the received $h(W_S, U_A')$. If this holds, A computes $h(W_A, W_S, U_A')$ and sends it with ID_A to S.

PA4. $S \rightarrow A$: Access Granted/Denied

S computes $h(W_A, W_S, U_A')$ by its own W_S and (W_A, U_A') which is received from A in PA1. Then S checks whether the hashed result of (W_A, W_S, U_A') equals the received $h(W_A, W_S, U_A')$. If this holds, S grants A's login request and replaces old password-verifier U_A with new password-verifier $U_A' = pw_A \cdot r_A' \cdot G$, otherwise denies A's login request. Finally, A's smart card replaces r_A with r_A' if all of the conditions are satisfied.

After finishing the password authentication phase, the verifier table is updated and the content of the verifier table is shown in Table 2.

4.3. Password change phase

When A wants to change his/her password pw_A to a new password pw_A', he/she should perform the following steps:

PC1. $A \rightarrow S: ID_A, E_{Kx}(ID_A, h(r_A'), W_A, U_A')$

PC2. $S \rightarrow A: (W_A + W_S), h\big(W_S, U_A', h(r_A')\big)$

PC3. $A \rightarrow S: ID_A, h(W_A, W_S, U_A'), h(W_S + W_A + U_A')$

PC4. $S \rightarrow A$: Password Change Granted/Denied

In this phase, U_A' is equal to $pw_A' \cdot r_A' \cdot G$. In PC4, if the messages $h(W_A, W_S, U_A')$ and $h(W_S + W_A + U_A')$ from A are valid, S updates U_A with U_A'. Moreover, A's smart card replaces r_A with r_A'.

After finishing the password change phase, the verifier table is updated and the content of the verifier table is shown in Table 3.

4.4. Session key distribution phase

S1. $A \rightarrow S: ID_A, E_{Kx}(ID_A, h(r_A'), W_A, U_A')$

S2. $S \rightarrow A: (W_A + W_S), h(W_S, U_A', SK)$

S3. $A \rightarrow S: ID_A, h(W_A, W_S, U_A', SK)$

S4. $S \rightarrow A$: Key Distribution Granted/Denied

In this phase, S computes the session key $SK = r_S \cdot W_A$ and A computes the session key $SK = (h(r_A') \cdot r_A \cdot pw_A) \cdot W_S$. Obviously, $SK = r_S \cdot W_A = (h(r_A') \cdot r_A \cdot pw_A) \cdot W_S = h(r_A') \cdot r_A \cdot pw_A \cdot r_S \cdot d_S \cdot G$. in S4, if the message $h(W_A, W_S, U_A', SK)$ from A is valid, S updates U_A with U_A', where $U_A' = pw_A \cdot r_A' \cdot G$. Next, A's smart card replaces r_A with r_A'.

4.5. User eviction phase

If the server wants to evict a client A, S can delete (ID_A, U_A) from its verifier table and A cannot use (ID_A, U_A) to login S since ID_A cannot be found in the verifier table in the password authentication phase.

4.6. Provision of user anonymity

We also extend a user anonymity scheme to provide the user anonymity. In this scheme, the client's true identity and location cannot be traced by any attackers over public networks. Fig. 3 shows our user anonymity scheme.

4.6.1. Registration phase

In this phase, A registers to the server with his/her password-verifier $U_A = pw_A \cdot r_A \cdot G$ and personal cre-

dentials, for example National ID card. S generates a pseudonym IND_A, issues a smart card which contains $\{IND_A, G, U_S, h(\cdot), E_K(\cdot)/D_K(\cdot)\}$ and sends it to A through a secure channel. Afterward, A enters r_A into his/her smart card and the smart card contains $\{r_A, IND_A, G, U_S, h(\cdot), E_K(\cdot)/D_K(\cdot)\}$.

After finishing the registration phase, the content of the verifier table is shown in Table 4.

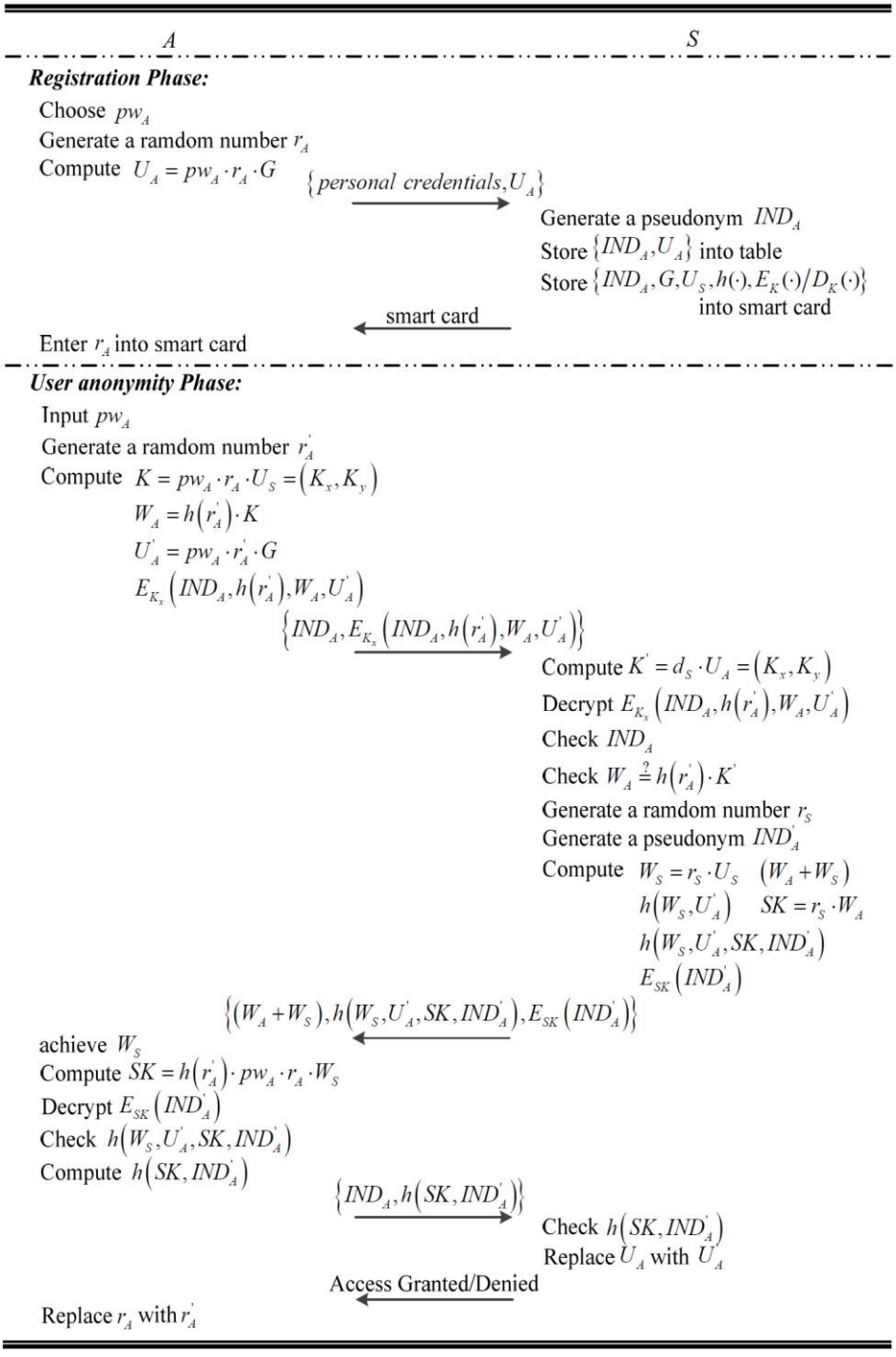

Figure 3. Our user anonymity scheme

4.6.2. User anonymity phase

When a client A wants to access the server anonymously, he/she should perform the following steps:

U1. $A \rightarrow S$: IND_A, $E_{Kx}(IND_A, h(r_A'), W_A, U_A')$

A inserts the smart card into the card reader and inputs the password pw_A. The smart card computes K, W_A, U_A' and $E_{Kx}(IND_A, h(r_A'), W_A, U_A')$. Then, A sends IND_A and $E_{Kx}(IND_A, h(r_A'), W_A, U_A')$ to S.

U2. $S \rightarrow A$: $(W_A + W_S)$, $h(W_S, U_A', SK, IND_A')$, $E_{SK}(IND_A')$

If IND_A, R_A and W_A are valid, S generates a new pseudonym IND_A' and a random number r_S, computes a session key $SK = r_S \cdot W_A$ and sends $(W_A + W_S)$, $h(W_S, U_A', SK, IND_A')$, and $E_{SK}(IND_A')$ to A.

U3. $A \rightarrow S$: IND_A', $h(SK, IND_A')$

A achieves W_S by subtracting W_A form $(W_A + W_S)$ and computes the session key $SK = (h(r_A') \cdot r_A \cdot pw_A \cdot W_S)$ to decrypt $E_{SK}(IND_A')$. Then, A checks whether the hashed result of (W_S, U_A', SK, IND_A') equals the received $h(W_S, U_A', SK, IND_A')$. If this holds, A computes $h(SK, IND_A')$ and sends it with IND_A to S.

U4. $S \rightarrow A$: User Anonymity Granted/Denied

S checks whether the hashed result of (SK, IND_A') equals the received $h(SK, IND_A')$. If this holds, S grants A's anonymously login request and replaces (IND_A, U_A) with (IND_A', U_A'), then A's smart card replaces (r_A, U_A) with (r_A', U_A'), otherwise S denies A 's anonymously login request.

After finishing the user anonymity phase, the content of the verifier table is shown in Table 5.

5. Security proof

In this section, we will prove that our scheme is secure and practical based on the Burrows-Abadi-Needham (BAN) logic [18,19]. In our proposed scheme, the password authentication phase and the password change phase are similar. Compared with the password authentication phase, the session key distribution phase just have one more step (computing the session key). So, the session key distribution phase contains the password authentication. As the result, we only need to prove that the session key distribution phase of our proposed scheme is secure. Then, we can affirm that our proposed scheme is secure. We describe our process of proof in the following steps:

Step 1: We show the goals of the session key distribution phase in our proposed scheme as follows:

Goal 1: $A| \equiv (A \overset{SK}{\leftrightarrow} S)$.

Goal 2: $A| \equiv S| \equiv (A \overset{SK}{\leftrightarrow} S)$

Goal 3: $S| \equiv (A \overset{SK}{\leftrightarrow} S)$

Goal 4: $S| \equiv A| \equiv (A \overset{SK}{\leftrightarrow} S)$

Step 2: We transform the session key distribution phase of our proposed scheme to the idealized form as follows:

Msg 1: $A \rightarrow S$: $\left(W_A, U_A, A \overset{W_A}{\longleftrightarrow} S\right)_K$.

Msg 2: $S \rightarrow A$:
$\left(W_A, W_S, A \overset{W_A}{\longleftrightarrow} S, A \overset{W_S}{\longleftrightarrow} S, A \overset{SK}{\longleftrightarrow} S\right)_{W_A}$

Msg 3: $A \rightarrow S$: $\left(W_A, W_S, A \overset{SK}{\longleftrightarrow} S\right)_{W_A}$.

Step 3: We represent the assumptions about the initial state of the session key distribution phase in our proposed scheme as follows:

Aspt 1: $A| \equiv \#(W_A)$.

Aspt 2: $S| \equiv \#(W_S)$.

Aspt 3: $S| \equiv \#(U_A)$.

Aspt 4: $A| \equiv \#(A \overset{K}{\longleftrightarrow} S)$.

Aspt 5: $S| \equiv \#(A \overset{K}{\longleftrightarrow} S)$.

Aspt 6: $A| \equiv S \Rightarrow (A \overset{W_S}{\longleftrightarrow} S)$.

Aspt 7: $S| \equiv A \Rightarrow (A \overset{W_A}{\longleftrightarrow} S)$.

Aspt 8: $A| \equiv (A \overset{W_A}{\longleftrightarrow} S)$.

Aspt 9: $S| \equiv (A \overset{W_S}{\longleftrightarrow} S)$.

Step 4: We prove the security of the session key distribution phase in our proposed scheme based on the BAN logic as follows:

1) The proof of Goal 1 and Goal 2:

According the Msg 2, we can know:

$S1_{G1,2}$: $A \triangleleft$
$\left(W_A, W_S, A \overset{W_A}{\longleftrightarrow} S, A \overset{W_S}{\longleftrightarrow} S, A \overset{SK}{\longleftrightarrow} S\right)_{W_K}$.

According to Aspt 8, we apply the message-meaning rule to achieve:

$S2_{G1,2}$: $A| \equiv S| \sim$
$\left(W_A, W_S, A \overset{W_A}{\longleftrightarrow} S, A \overset{W_S}{\longleftrightarrow} S, A \overset{SK}{\longleftrightarrow} S\right)$.

According to Aspt 1, we apply the freshness-conjuncatenation rule to achieve:

$S3_{G1,2}$: $A| \equiv \#$
$\left(W_A, W_S, A \overset{W_A}{\longleftrightarrow} S, A \overset{W_S}{\longleftrightarrow} S, A \overset{SK}{\longleftrightarrow} S\right)$.

According to $S2_{G1,2}$ and $S3_{G1,2}$, we apply the nonce-verification rule to achieve:

$S4_{G1,2}$: $A| \equiv S| \equiv$
$\left(W_A, W_S, A \overset{W_A}{\longleftrightarrow} S, A \overset{W_S}{\longleftrightarrow} S, A \overset{SK}{\longleftrightarrow} S\right)$.

According to $S4_{G1,2}$, we apply the BAN logic rule to break conjunctions to achieve:

$S5_{G1,2}$:. $A| \equiv S| \equiv (A \overset{SK}{\longleftrightarrow} S)$ (Goal 2)

According to $S4_{G1,2}$, we apply the BAN logic rule to break conjunctions to achieve:

$S6_{G1,2}$: $A| \equiv S| \equiv (A \overset{W_S}{\longleftrightarrow} S)$.

According to $S6_{G1,2}$ and Aspt 6, we apply the jurisdiction rule to achieve:

$S7_{G1,2}$: $A| \equiv (A \overset{W_S}{\longleftrightarrow} S)$.

According to $SK = (h(r_A') \cdot r_A \cdot pw_A) \cdot W_S$, we could achieve:

$S8_{G1,2}$: $A| \equiv (A \overset{SK}{\longleftrightarrow} S)$. (Goal 1)

2) The proof of Goal 3 and Goal 4:

According the Msg 1, we can know:

$S1_{G3,4}$: $S \lhd \left(W_A, U_A, A \overset{W_A}{\longleftrightarrow} S\right)_K$.

According to Aspt 5, we apply the message-meaning rule to achieve:

$S2_{G3,4}$: $S| \equiv A| \sim \left(W_A, U_A, A \overset{W_A}{\longleftrightarrow} S\right)$.

Table 6. Notations

$P\| \equiv X$	P believes X
$\#(X)$	X is fresh
$P \Rightarrow X$	P has jurisdiction over X
$P \lhd X$	P sees X
$P\| \sim X$	P once said X
(X, Y)	X or Y is one part of (X, Y)
$(X)_Y$	X is hash with the key K
$P \overset{K}{\longleftrightarrow} X$	P and Q use the shared key K to communicate
SK	The session key used in the current session
$\frac{P\| \equiv P \overset{K}{\longleftrightarrow} Q, P \lhd \{X\}_K}{P\| \equiv Q\| \sim X}$	The message-meaning rule
$\frac{P\| \equiv \#(X)}{P\| \equiv \#(X, Y)}$	The freshness-conjuncatenation rule
$\frac{P\| \equiv \#(X), P\| \equiv Q\| \sim X}{P\| \equiv Q\| \equiv X}$	The nonce-verfication rule
$\frac{P\| \equiv Q \Rightarrow X, P\| \equiv Q\| \equiv X)}{P\| \equiv X}$	The jurisdiction rule

According to Aspt 3, we apply the freshness-conjuncatenation rule to achieve:

$S3_{G3,4}$: $S| \equiv \#\left(W_A, U_A, A \overset{W_A}{\longleftrightarrow} S\right)$.

According to $S2_{G3,4}$ and $S3_{G3,4}$, we apply the nonce-verification rule to achieve:

$S4_{G3,4}$: $S| \equiv A| \equiv \left(W_A, U_A, A \overset{W_A}{\longleftrightarrow} S\right)$.

According to $S4_{G3,4}$, we apply the BAN logic rule to break conjunctions to achieve:

$S5_{G3,4}$: $S| \equiv A| \equiv \left(A \overset{W_A}{\longleftrightarrow} S\right)$.

According to Aspt 7, we apply the jurisdiction rule to achieve:

$S6_{G3,4}$: $S| \equiv \left(A \overset{W_A}{\longleftrightarrow} S\right)$.

According the Msg 3, we can know:

$S7_{G3,4}$: $S \lhd \left(W_A, W_S, A \overset{SK}{\longleftrightarrow} S\right)_{W_A}$.

According to $S6_{G3,4}$ and $S7_{G3,4}$, we apply the message-meaning rule to achieve:

$S8_{G3,4}$: $S| \equiv A| \sim \left(W_A, W_S, A \overset{SK}{\longleftrightarrow} S\right)$.

According to Aspt 2, we apply the freshness-conjuncatenation rule to achieve:

$S9_{G3,4}$: $S| \equiv \#\left(W_A, W_S, A \overset{SK}{\longleftrightarrow} S\right)$.

According to $S8_{G3,4}$ and $S9_{G3,4}$, we apply the nonce-verification rule to achieve:

$S10_{G3,4}$: $S| \equiv A| \equiv \left(W_A, W_S, A \overset{SK}{\longleftrightarrow} S\right)$.

According to $S10_{G3,4}$, we apply the BAN logic rule to break conjunctions to achieve:

$S11_{G3,4}$: $S| \equiv A| \equiv \left(A \overset{SK}{\longleftrightarrow} S\right)$ (Goal 4)

According to $S6_{G3,4}$ and $SK = r_S \cdot W_A$, we could achieve:

$S11_{G3,4}$: $S| \equiv \left(A \overset{SK}{\longleftrightarrow} S\right)$ (Goal 3)

According the proof above, we can know that both A and S believe that the session key is shared between A and S.

6. Security analysis and comparisons

In this section, we will demonstrate that our proposed scheme is more efficient and secure than Li's scheme. Our proposed scheme is secure against various attacks and can achieve all security requirements. The cyber criminal might perform various attacks as follows.

6.1. Resistance to smart card loss attack

A client A's smart card contains $\{r_A, G, U_S, h(\cdot), E_K(\cdot)/D_K(\cdot)\}$. If the cyber criminal steals the smart card, he/she could only get the secret data r_A from it, the other data in the smart card are public to all clients. However, he/she doesn't know the identity ID_A or the password pw_A of A. As a result, he/she cannot use the secret data r_A to impersonate the client A to login in the server for some resources or guess the password of A utilizing the password-guessing attack. Therefore, our proposed scheme can withstand the smart card loss attack.

6.2. Resistance to offline password guessing attack

In our scheme, if an adversary eavesdrops the messages exchanged between the client A and the server S, he/she cannot guess the password pw_A from those messages. Because the adversary dose not know

A's random number r_A, he/she cannot guess the password pw_A from the encryption key $(Kx, Ky) = K = pw_A \cdot r_A \cdot U_S = pw_A \cdot r_A \cdot d_S \cdot G$, the sum of $(W_A + W_S)$ and those hash values.

6.3. Resistance to insider attack

Even if the adversary steals the identity and password-verifier from the server's verifier table, he/she cannot impersonate a legal client A to access the server. The adversary has no idea to derive the password pw_A and the random number r_A from the password-verifier $U_A = pw_A \cdot r_A \cdot G$. So, he/she is unable to compute a valid encrypted message $E_{Kx}(ID_A, h(r_A'), W_A, U_A')$.

6.4. Resistance to impersonation attack

An adversary may attempt to forge a valid login message to masquerade the legal client and login in the server. However, it is impossible for the adversary to compute a valid U_A' without the knowledge of the client's password. Moreover, the adversary is unable to forge W_A and a valid login request $E_{Kx}(ID_A, h(r_A'), W_A, U_A')$, because the symmetric secret key Kx is known only to the client and the server.

6.5. Resistance to denial of service attack

In general, a denial-of-service (DoS) attack can disrupt the availability of the authentication between the clients and the server. If the server undertakes too many computationally expensive cryptographic operations, the resources of the server would be exhausted immediately. However, our proposed scheme no longer needs bilinear pairing operations and reduces the burden on the server. Therefore, our proposed scheme is secure against DoS attacks.

6.6. Mutual authentication

Mutual authentication is an essential requirement in many systems. In step PA3 of our scheme, the client sends $E_{Kx}(ID_A, h(r_A'), W_A, U_A')$ to the server. The server verifies the user by checking whether $h(r_A') \cdot K'$ equals W_A, then sends the message $\{(W_A + W_S), h(W_S, U_A')\}$ to the client. After receiving these messages from the server, only the legal client can obtain W_S to authenticate the server by checking whether the hashed result of (W_S, U_A') equals the received $h(W_S, U_A')$ and sends $h(W_A, W_S, U_A')$ to the server. The server must check whether the hashed result of (W_A, W_S, U_A') equals the received $h(W_A, W_S, U_A')$. After the authentication steps, the client and the server replaced old r_A and U_A with new r_A' and U_A'. Obviously, our scheme can achieve mutual authentication.

6.7. Freely choosing identity, freely choosing and updating the password

In the registration phase of our proposed scheme, a client can freely choose his/her identity ID_A and password pw_A. The server only stores identity ID_A and password-verifier $U_A = pw_A \cdot r_A \cdot G$. Nobody except the client knows the pw_A. After the registration phase, the client can update his/her password through the password change phase.

6.8. Provide perfect forward secrecy

Perfect forward secrecy means that the attacker cannot extract the past session keys, even if he/she succeeds to obtain a subset of session keys in some way. In our proposed scheme, the session key $SK = r_S \cdot W_A = (h(r_A') \cdot r_A \cdot pw_A) \cdot W_S = h(r_A') \cdot r_A \cdot pw_A \cdot r_S \cdot d_S \cdot G$ is established with the secure random number r_A, r_A' and $r_S (r_A, r_A'$ are chosen by the client and r_S is chosen by the server). Obviously, the session keys are independent. Even if the attacker obtains some session keys, he/she cannot achieve any other session keys. Therefore, our proposed scheme can provide perfect forward secrecy.

Table 7. Comparison of our scheme and that of Li

Properties	Our scheme	Li's scheme
Registration phase	1P	1P
Password authentication phase	5H+2S+6P+2A	4H+2E+2S+7P+2A
Password change phase	7H+2S+6P+6A	6H+2E+2S+7P+6A
Session key distribution phase	5H+2S+8P+2A	4H+2E+2S+9P+2A
User anonymity phase	5H+4S+8P+2A	4H+2E+4S+9P+2A

6.9. Performance analysis

A comparison of our scheme and that of Li is summarized in Table 7. We define the notation H as a one-way hash function, E as a bilinear pairing, S as the operation of symmetric encryption/decryption, P as the operation of point multiplication and A as the operation of elliptic curve point addition/subtraction. We know that one-way hash function is more efficient than the operation of point multiplication. The scheme proposed by Li used bilinear pairing which is abandoned in our scheme to authenticate a client. Based on the description above, our scheme is more simple and efficient than Li's scheme.

7. Conclusions

In this paper, we discuss the security flaws of the password authentication and user anonymity scheme proposed by Li. Due to the similarity of password authentication phase and password change phase, when a legal client A wants to change his/her password, a cyber criminal can impersonate A to access the server. We also find that the verification $\hat{e}(d_S \cdot$

$R_A, U_A) = \hat{e}(W_A, U_S)$ in Li's scheme is not correct. Moreover, the scheme proposed by Li used bilinear pairing to authenticate a client. Then we proposed an efficient password authentication scheme using smart card based on elliptic curve cryptography. Comparing with Li's scheme, our scheme has lower computation costs, and is more secure than Li's scheme.

References

[1] **Tzung-Her Chen, Wei-Bin Lee.** A new method for using hash functions to solve remote user authentication. *Computers & Electrical Engineering*, 2008, Vol. 34, No. 1, 53-62.

[2] **H. Y. Chien, J. K. Jan, Y. M. Tseng.** An efficient and practical solution to remote authentication: smart card. *Computers & Security*, 2002, Vol. 21, No. 4, 372–375.

[3] **C. L. Hsu.** Security of Chien et al.'s remote user authentication scheme using smart cards. *Computer Standards & Interfaces*, 2004, Vol. 26, No. 3, 167–169.

[4] **C. T. Li.** Secure smart card based password authentication scheme with user anonymity. *Information Technology and Control*, 2011, Vol. 40, No. 2, 157–162.

[5] **R. Song.** Advanced smart card based password authentication protocol. *Computer Standards & Interfaces*, 2010, Vol. 32, No. (5–6), 321–325.

[6] **R. C. Wang, W. S. Juang, C. L. Lei.** Robust authentication and key agreement scheme preserving the privacy of secret key. *Computer Communications*, 2011, Vol. 34, No. 3, 274–280.

[7] **H. Wang, Y. Zhang, H. Xiong, B. Qin.** Cryptanalysis and improvements of an anonymous multi-receiver identity-based encryption scheme. *IET Information Security*, 2012, Vol. 6, No. 1, 20–27.

[8] **L. Lamport.** Password authentication with insecure communication. *Communications of ACM*, 1981, Vol. 24, No. 11, 770-772.

[9] **Eun-Jun Yoon, Kee-Young Yoo.** A robust and flexible biometrics remote user authentication scheme. *International Journal of Innovative Computing Information and Control*, 2012, Vol. 8, No. 5, 3173-3188.

[10] **Min Xie, Libin Wang.** One-round identity-based key exchange with Perfect Forward Security. *Information Processing Letters*, 2012, Vol. 112, No. 14-15, 587-591.

[11] **Eun-Jun Yoon, Sung-Bae Choi, Kee-Young Yoo.** A secure and efficiency ID-based authenticated key agreement scheme based on elliptic curve cryptosystem foe mobile devices. *International Journal of Innovative Computing Information and Control*, 2012, Vol. 8, No. 4, 2637-2653.

[12] **M. L. Das, A. Saxena, V. P. Gulati.** A dynamic ID-based remote user authentication scheme. *IEEE Transactions on Consumer Electronics*, 2004, Vol. 50, No. 2, 629-631.

[13] **A. K. Das.** Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards. *IET Information Security*, 2011, Vol. 5, No. 3, 145–151.

[14] **Horng-Twu Liaw, Jiann-Fu Lin, Wei-Chen Wu.** An efficient and complete remote user authentication scheme using smart cards. *Mathematical and Computer Modelling*, 2006, Vol. 44, No. 1-2, 223-228.

[15] **S. H. Islam, G. P. Biswas.** Design of improved password authentication and update scheme based on elliptic curve cryptography. *Mathematical and Computer Modelling*, 2013, Vol. 57, No. 11-12, 2703-2717.

[16] **C. L. Lin, T. Hwang.** A password authentication scheme with secure password updating. *Computers & Security*, 2003, Vol. 22, No. 1, 68-72.

[17] **Chun-Ta Li.** A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card. *IET Information Security*, 2013, Vol. 7, No. 1, 3-10.

[18] **M. Burrows, M. Abadi, R. Needham.** A logic of authentication. *ACM Transactions on Computer Systems*, 1990, Vol. 8, No. 1, 18-36.

[19] **M. K. Khan, D. He.** A new dynamic identity-based authentication protocol for multi-server environment using elliptic curve cryptography. *Security and Communication Networks*, 2012, Vol. 5, No. 11, 1260-1266.

Received March 2014.