DYNAMIC ID-BASED REMOTE USER MUTUAL AUTHENTICATION SCHEME WITH SMARTCARD USING ELLIPTIC CURVE CRYPTOGRAPHY[1]

SK Hafizul Islam G. P. Biswas*

(*Department of Computer Science and Information Systems*, *Birla Institute of Technology and Science*, *Pilani*, *Rajasthan 333031*, *India*)

*(*Department of Computer Science and Engineering*, *Indian School of Mines*, *Dhanbad, Jharkhand 826004*, *India*)

Abstract In the literature, several dynamic ID-based remote user mutual authentication schemes are implemented using password, smartcard and Elliptic Curve Cryptography (ECC), however, none of them provides resilience against different attacks. Therefore, there is a great need to design an efficient scheme for practical applications. In this paper, we proposed such a scheme in order to provide desired security attributes and computation efficiencies. Compared with other existing techniques, our scheme is more efficient and secured. In addition, our scheme is provably secure in the random oracle model under the hardness assumption of computational Diffie-Hellman problem.

Key words Elliptic Curve Cryptography (ECC); Smartcard; Mutual authentication; Password; Perfect forward secrecy

CLC index TP309.7

DOI 10.1007/s11767-014-4002-0

I. Introduction

Remote user authentication means, a remote server and a user mutually authenticate the legitimacy of each other over an unreliable network. The conventional remote login schemes[1–4] maintained password-verifier table in the remote server for checking the validity of login request made by the user. But the verifier-based remote login scheme suffers from some potential vulnerabilities such as the risk of modifying the password-verifier table by an adversary and then the entire system will be compromised. Consequently, to provide protection from such attack, the relevant cost of the server will be high. In remedy of the above problems, researchers are interested in designing static ID-based remote login schemes[5–11] using smartcard. However, in several applications including digital library, online voting, online money transaction, pay-TV, online shopping, *etc.*, a transaction with static login identity discloses some secret information about the user. An outsider may intercept the user's login message and tries to manipulate with other parameters to forge the user's login identity, known as ID-theft attack[12].

1. Related studies

In 2004, Das *et al.*[13] proposed a dynamic ID-based remote user mutual authentication scheme using password and smartcard. Although Das *et al.*'s scheme used no verifier table and its security is based on one-way hash function, the researchers have identified many vulnerabilities, including impersonation attack[12], privileged-insider attack[14–18], and password guessing attack[12,16]. Furthermore, the scheme is incapable to protect user's anonymity[19] and an adversary can login the server by randomly chosen password[17,20]. In addition, Das *et al.*'s scheme[13] does not provide mutual authentication and session key agreement[12,18]. In 2009, Wang *et al.*[17] independently demonstrated that Das *et al.*'s scheme[13] is password independent and does not provide mutual authentication. An improvement of Das *et al.*'s scheme[13] was proposed by Wang *et al.*[17] and they claimed that the improved

[1] Manuscript received date: January 2, 2014; revised date: June 12, 2014.

Communication author: SK Hafizul Islam, born in 1982, male, Ph.D., Assistant Professor. Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani, Rajasthan 333031, India. Email: hafi786@gmail.com, hafizul.ism@gmail.com, hafizul@pilani.bits-pilani.ac.in.

scheme is secured under all known attacks. Unfortunately, Ahmed *et al.*[21] analyzed that Wang *et al.*'s scheme[17] is vulnerable to different attacks, such as password guessing attack, user masquerade attack, server masquerade attack, and Denial of Service (DoS) attack. Later on, Khan *et al.*[22] also identified that Wang *et al.*[17] has the following security flaws: no provision of user's anonymity, the user cannot choose his password, vulnerability to insider attack, no provision for revocation of lost smartcard, and does not provide session key agreement. In Ref. [22], to remove the above mentioned security flaws, Khan *et al.* proposed an improved scheme. However, Khan *et al.*'s scheme[22] does not provide user's anonymity, session key forward secrecy, known-key secrecy, and also it is vulnerable to password guessing attack and DoS attack[23].

Liao and Wang[24] proposed a dynamic ID-based remote login scheme for multiserver environments using hash function, but Hsiang and Shih[25] showed that Liao and Wang's scheme[24] is vulnerable to server's spoofing attack, privileged-insider attack and masquerade attack and it also failed to provide mutual authentication. In remedy of these flaws, Hsiang and Shih[25] proposed an improved scheme, however, Saho and Chin[26] demonstrated that Liao and Wang's scheme[24] is incapable to provide user's anonymity and vulnerable to server's spoofing attack. Hsiang and Shih[25] proposed an enhancement of Liao and Wang's scheme[24], but later on, Tan[27] proved that Hsiang and Shih's scheme[25] is still vulnerable to off-line password guessing attack, impersonation attack and server's spoofing attack and it cannot protect the extraction of secret data by intercepting the authentication message.

In 2009, Yang and Chang[28] proposed an identity-based remote user mutual authentication scheme for mobile users using elliptic curve cryptography. However, Yang and Chang's scheme[28] suffers from replay attack, clock synchronization problem, known session-specific temporary information attack, inability to protect user's anonymity and does not provide the session key forward secrecy. In addition, Yang and Chang's scheme[28] does not define how to revoke the authentication key with the same login identity, in case if the authentication key is leaked to an adversary by some means[29–31]. Subsequently, two improvements over Yang and Chang's scheme[28] have been proposed by Yoon and Yoo[29], and Chen *et al.*[30]. However, the schemes[29,30] are vulnerable to some known attacks as discussed in Ref. [31].

2. Our contributions

In the literature, several ID-based remote user mutual authentication schemes have been designed based on the one-way hash function. Although the earlier schemes have low computational cost, however, none of them provide sufficient security against cryptographic attacks. We have considered all deficiencies of the previous schemes and proposed an efficient dynamic ID-based remote user mutual authentication and session key agreement scheme using smartcard, password, and Elliptic Curve Cryptosystem (ECC). We have defined an adversarial model based on which the provable security of our scheme is analyzed. It is proven that our scheme is provably secure in the random oracle model with the hardness assumption of the computational Diffie-Hellman problem. Compared to the related schemes, our scheme is secured from known attacks and computationally efficient.

3. Organization of the paper

The rest of the paper is organized as follows. Section II describes the theory of elliptic curve and some computational problems on it. We defined a formal attack model in the Section III. The proposed scheme is described in Section IV. The formal security analysis and efficiency analysis of our scheme is conducted in Section V. Finally, some concluding remarks are given in Section VI.

II. Mathematical Preliminaries

This section discussed the theory of ECC and some mathematical hard problems on it.

1. Elliptic curve cryptosystem

Recently, ECC[32,33] has been accepted as an efficient tool in Public Key Cryptography (PKC) due to the computation, communication, and security strengths. For example, it offers same level of security at reduced key sizes than other PKCs. Below is the brief description of ECC.

Let E/F_p be a set of elliptic curve points over a prime field F_p, defined by the following non-singular elliptic curve:

$$y^2 \bmod p = \left(x^3 + ax + b\right) \bmod p \qquad (1)$$

where $x, y, a, b \in F_p$ and $(4a^3 + 27b^2) \bmod p \neq 0$. A point $P(x, y)$ is an elliptic curve point if it satisfies Eq. (1), and the point $Q(x, -y)$ is called the negative of P, *i.e.* $Q = -P$. Let $P(x_1, y_1)$ and $Q(x_2, y_2)$ $(P \neq Q)$ be two points on the curve (1), the line l (tangent to the curve (1) if $P = Q$) joining the points P and Q intersects the curve (1) at $-R(x_3, -y_3)$ and the reflection of it with respect to x-axis is the point $R(x_3, y_3)$, *i.e.* $P + Q = R$. The points E/F_p together with a point O, called "point at infinity" or "zero point", make the additive elliptic curve cyclic group G_p, *i.e.* $G_p = \{(x, y) : x, y \in F_p \text{ and } (x, y) \in E/F_p\} \cup \{O\}$ of prime order p. The scalar point multiplication on G_p is defined as: $k \cdot P = P + P + \cdots + P(k \text{ times})$. A generator point $P \in G_p$ has order n if n is the smallest positive integer and $nP = O$ [34].

2. Computational problems

This subsection summarizes some existing computational problems on the elliptic curve group. The proposed scheme is based on the following hard problems.

Definition 1 Elliptic Curve Discrete Logarithm Problem (ECDLP) Given a tuple $(P, Q) \in G_p$, it is computationally hard by a polynomial-time bounded algorithm to find an integer $k \in [1, n-1]$ such that $Q = kP$ [28,35,36].

Definition 2 Computational Diffie-Hellman Problem (CDHP) Given a tuple $(P, a \cdot P, b \cdot P) \in G_p$ for any $a, b \in [1, n-1]$, computation of $a \cdot b \cdot P$ is hard by a polynomial-time bounded algorithm[28,35,36].

Definition 3 Elliptic Curve Factorization Problem (ECFP) Given a tuple $(P, Q) \in G_p$, where $Q = aP + bP$ and $a, b \in [1, n-1]$. Computation of aP and bP are hard by a polynomial-time bounded algorithm[28].

III. Formal Attack Model of a Password-based Authentication Scheme

In this section, we defined a formal security model of a password-based authentication scheme[37–39]. In this model, we assume that each participant is either a user $A \in \text{User}$ or a server $S \in \text{Server}$. Also we assume that S holds a private key d_S and each A holds a password PW_A, which is chosen from the small dictionary $\mathcal{D}$. During the registration phase, S stores t into the smartcard and returns the card to A over an out-of-band channel, where t is an (injective) transformation of the password PW_A of A and the secret key d_S of S. In this model, we assume that a probabilistic polynomial time adversary $\mathcal{A}$ and any participant U, where $U \in \text{User}$ or $U \in \text{Server}$ interacts by executing oracle queries, which gives the capability of $\mathcal{A}$ to attack the authentication protocol. We denote U^i be the i instance of a protocol participant U. It is also assume that $\mathcal{A}$ controls the communication channel, *i.e.* $\mathcal{A}$ can intercept, block, inject, remove, or modify, any messages transmitted in the media. In other words, we can say that all the messages between A^i and S^i are transmitted via $\mathcal{A}$. In order to compromise the security of the authentication scheme, $\mathcal{A}$ can ask the following polynomial number of queries:

(1) Execute(A^i, S^j) $\mathcal{A}$ can simulate the passive attack by executing this query. The output of this query consists of the messages that were exchanged during the honest execution of the protocol.

(2) Send(U^i, m) $\mathcal{A}$ can simulate the active attack by executing this query. $\mathcal{A}$ can send a message m to U^i through this query. Upon receiving m to U^i generates some messages according to the protocol description and returns them to $\mathcal{A}$.

(3) Reveal(U^i) This query models the misuse of session keys. This query returns the session key SK of U^i to $\mathcal{A}$ if the session has accepted, otherwise returns a null value.

(4) Corrupt(A, a) To get secret information of U^i, A can issue this query to U^i. This query outputs in the following ways:

(i) If $a = 1$, it outputs the password PW_A of the user A.

(ii) If $a = 2$, it outputs the data stored in the smartcard including t (an injective transformation of PW_A and d_S.

(5) Test(U^i) This query measures the semantic security of the session key. $\mathcal{A}$ can send a single Test query to U^i. Upon receiving this query, U^i flips an unbiased coin b and returns the session key SK of U^i to $\mathcal{A}$ if $b = 1$, or returns a random value with the same bit-length as of session key if

$b=0$.

Definition 4 An instance U^i has accepted if it goes into an accept state after receiving the last expected protocol message.

Definition 5 The session identification (sid) of instance U^i is the concatenation of all messages sent and received by U^i.

Definition 6 Let $A \in \text{User}$ and $S \in \text{Server}$. The instances A^i and S^j are partnered if the following conditions hold: (1) Both A^i and S^j is in the state accepted, *i.e.*, they mutually authenticate each other and they hold the same session key; (2) Both A^i and S^j share the same session id (sid); (3) A^i and S^j 's partner and vice-versa.

Definition 7 An instance U^i is fresh if the following conditions are met: (1) U^i is in the state accepted, *i.e.*, U^i and its partner mutually authenticate each other and hold the same session key, (2) No Reveal queries have been made to U^i or its partner; (3) If $U \in \text{User}$, strictly less than two Corrupt-queries have been made to U^i. Else if $\text{P} \in \text{Server}$, strictly less than two Corrupt-queries have been made to U^i 's partner.

Definition 8 Let $\text{Succ}(\mathcal{A})$ be the event that $\mathcal{A}$ makes a single Test query to some fresh U^i that has terminated, and finally outputs a guess bit b', where $b'=b$ for the bit b that was selected in the Test query. The advantage of $\mathcal{A}$ in violating the semantic security of the Password Authentication Protocol (PAP) is defined as $A\text{dv}_{\mathcal{A}}^{\text{PAP}}(k)=2\Pr\cdot[\text{Succ}(\mathcal{A})]-1$.

Definition 9 The protocol PAP is semantically secure if (1) in the presence of the adversary $\mathcal{A}$, U^i and its partner are in accepted state and hold the same session key; (2) $A\text{dv}_{\mathcal{A}}^{\text{PAP}}(k)$ is negligible.

IV. Proposed Password-based Mutual Authentication Scheme

In this section, we proposed a new and efficient dynamic ID-based remote user mutual authentication scheme using smartcard and password on ECC. Our scheme has two entities, the user A, and the remote server S. The scheme consists of five phases: setup phase, registration phase, mutual authentication with session key agreement phase, password change phase, and lost smartcard revocation phase. Shown in Tab. 1, following notations are used through the paper.

Tab. 1 Notations

Notations	Meanings
A	The user
S	Remote server
ID_A	Identity of the user A
PW_A	Password of the user A
p	k-bit prime number, where k is security parameter
F_p	A prime field
E/F_p	Set of elliptic curve points
P	Base point with the order n
(d_S,V_S)	Secret/public key of the server S
H	Secure one-way hash function, where $H:\{0,1\}^* \to Z_p^*$
kdf	Key derivation function $\text{kdf}:\{0,1\}^* \to \{0,1\}^k$
$\|\|$	Concatenation operation
$\oplus$	Bitwise XOR operator
$+/-$	Elliptic curve point addition/subtraction
$(\cdot)$	Elliptic curve point multiplication

1. Setup phase

Step 1 S selects a k-bit prime number p and a base point P of order n from G_p.

Step 2 S selects a $d_S \in [1,n-1]$ as his private key and computes the public key as $V_S=d_S\cdot P$.

Step 3 S chooses a one-way secure hash function $H:\{0,1\}^* \to Z_p^*$ and a key derivation function $\text{kdf}:\{0,1\}^* \to \{0,1\}^k$.

Step 4 S publishes $\{G_p,P,n,V_S,H,\text{kdf}\}$ as system parameters and keeps d_S secret.

2. Registration phase

Step 1 User A selects his identity ID_A and a password PW_A, and then sends $(\text{ID}_A,\bar{V}_A)$, to S through a secure channel, where $\bar{V}_A=\text{PW}_A\cdot P$.

Step 2 S first checks ID_A which is different from all identities stored in the database or not. If not, S requests A for another one. S obtains the current timestamp T and computes $P_A=h_A\cdot P$, where $h_A=H(\text{ID}_A \,||\, T \,||\, d_S)$.

Step 3 S securely stores ID_A in the database against the user A.

Step 4 S issues a smartcard that contains $(\text{ID}_A, \bar{V}_A, V_S, P_A)$, and sends it to A through a secure channel.

3. Mutual authentication with session key agreement phase

In this phase, A inserts his smartcard to the terminal and keys the identity ID_A and password

PW_A, then the smartcard performs the followings:

Step 1 Compute $V_A = \mathrm{PW}_A \cdot P$ and check the validity of ID_A and the condition $V_A = ?\bar{V}_A$. If either of these is false, the smartcard rejects the login request and asks A for exact identity and password, otherwise proceed to the next step.

Step 2 Select a random number $r_A \in Z_p^*$, compute $R_A = r_A \cdot P$, $M_A = r_A \cdot (V_A + V_S)$ and $C_A = r_A \cdot (P_A + V_S)$.

Step 3 Compute a dynamic identity $\mathrm{DI}_A = \mathrm{ID}_A \oplus H(T_A \,||\, r_A \cdot V_A \,||\, r_A \cdot \mathrm{PW}_A \cdot V_S)$, where T_A is the current timestamp.

Step 4 Send the message $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ to S through an open channel.

After receiving the message $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$, the server S executes the following operations:

Step 5 S first checks the validity of the timestamp using $| T_A^* - T_A | \leq \Delta T_A$. If it is incorrect, S rejects the login request, otherwise proceed to the next steps.

Step 6 S computes $r_A \cdot V_A = M_A - d_S \cdot R_A$ and $r_A \cdot P_A = C_A - d_S \cdot R_A$, and then extracts $\mathrm{DI}_A = \mathrm{ID}_A \oplus H(T_A \,||\, r_A \cdot V_A \,||\, d_S \cdot r_A \cdot V_A)$. If ID_A is not valid, then S rejects the login request, otherwise computes $h_A = H(\mathrm{ID}_A \,||\, T \,||\, d_S)$ (where T is taken from the database).

Step 7 Check $h_A \cdot R_A = ?(r_A \cdot P_A)$. If the result is negative, the login request is rejected, otherwise S authenticates A.

Step 8 S selects a random number $r_S \in Z_p^*$, computes $M_S = r_S \cdot V_S + r_A \cdot V_A$ and $H_S = H(\mathrm{ID}_A \,||\, r_A \cdot V_A \,||\, r_S \cdot V_S \,||\, T_S)$, where T_S is the current timestamp. S then computes the session key $\mathrm{SK} = \mathrm{kdf}$ $(\mathrm{ID}_A ||\, \mathrm{Trans} \,||\, k)$, where $\mathrm{Trans} = (M_A \,||\, R_A \,||\, C_A \,||\, M_S \,||\, H_S \,||\, T_A \,||\, T_S)$ and $K = r_S \cdot d_S \cdot (r_A \cdot V_A) = r_S \cdot r_A \cdot d_S \cdot \mathrm{PW}_A \cdot P$. Then S sends the message (M_S, H_S, T_S) to A through the open channel.

Step 9 On receiving the message (M_S, H_S, T_S), user A checks the validity of the timestamp by $| T_S^* - T_S | \leq \Delta T_S$ and closes the login request if it is incorrect, otherwise computes $r_S \cdot V_S = M_S - r_A \cdot V_A$ and $H_S^* = H(\mathrm{ID}_A \,||\, r_A \cdot V_A \,||\, r_S \cdot V_S \,||\, T_S)$. A also verifies the condition $H_S^* = ?H_S$. If it holds, A authenticates S and computes the session key $\mathrm{SK} = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K)$, where $\mathrm{Trans} = (M_A \,||\, R_A \,||\, C_A \,||\, M_S \,||\, H_S \,||\, T_A \,||\, T_S)$ and $K = r_A \cdot \mathrm{PW}_A \cdot (r_S \cdot V_S) = r_S \cdot r_A \cdot d_S \cdot \mathrm{PW}_A \cdot P$. Otherwise A rejects S 's message.

4. Password change phase

The proposed scheme allows the user A to freely change his password without server's (S) agreement. To change the password, A inserts his smartcard into the card reader and keys (ID_A, PW_A) and the smartcard carries the following operations:

Step 1 Compute $V_A = \mathrm{PW}_A \cdot P$, and check the validity of ID_A and the condition $V_A = ?\bar{V}_A$. If either of them fails, smartcard rejects the password change request and asks A for exact identity and password, otherwise proceed to the next step.

Step 2 A inserts his new password $\mathrm{PW}_{\mathrm{new}}$, and then the smartcard computes $V_{\mathrm{new}} = \mathrm{PW}_{\mathrm{new}} \cdot P$ and updates the memory by replacing $\bar{V}_A$ with V_{new}.

After changing the password, A can get the updated memory of the smart card and can login to S with the new password PW_{new}.

5. Lost smartcard revocation phase

If A lost his smartcard, then he requests S for a new smartcard. In order to obtain a new smart card, both the user A/Smartcard and the server S do the following:

Step 1 A submits $(\mathrm{ID}_A, \bar{V}_A)$ and some personal information to S, where computes $\bar{V}_A = PW_A \cdot P$.

Step 2 Based on the information supplied by A, S checks whether A is valid. If A is valid, S chooses a new timestamp T_{new} and computes $h_A'' = H(\mathrm{ID}_A \,||\, T_{\mathrm{new}} \,||\, d_S)$ and $PW_A'' = h_A'' \cdot P$, and issues a new smartcard which contains $(\mathrm{ID}_A, \bar{V}_A, V_S, P_A'')$.

Step 3 S sends the smartcard to A over an secure channel and then updates the database (ID_A, T) to $(\mathrm{ID}_A, T_{\mathrm{new}})$ for A.

In the proposed scheme, A gets a new smartcard without changing his identity ID_A. If A wants a new smartcard with new password PW_A'', then he sends $(\mathrm{ID}_A, \mathrm{PW}_A'')$ to S in Step 1, where $\bar{V}_A'' = \mathrm{PW}_A'' \cdot P$. Accordingly, S issues a new smartcard for A as described above.

The Fig. 1, further explains the proposed scheme.

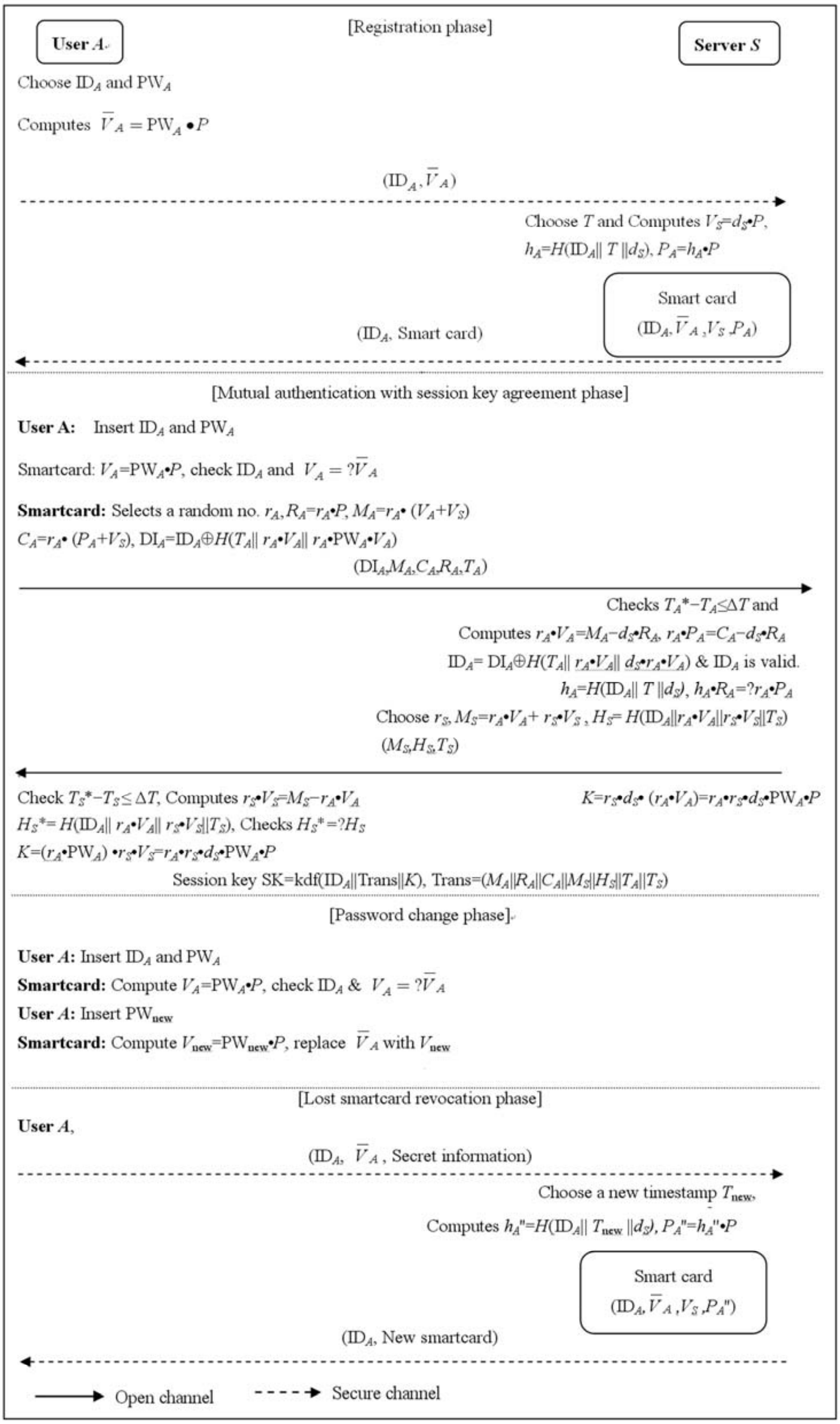
[Registration phase]
User A
Server S
Choose ID_A and PW_A
Computes $\overline{V}_A = \mathrm{PW}_A \bullet P$
$(\mathrm{ID}_A, \overline{V}_A)$
Choose T and Computes $V_S = d_S \bullet P$,
$h_A = H(\mathrm{ID}_A \| T \| d_S)$, $P_A = h_A \bullet P$
Smart card
$(\mathrm{ID}_A, \overline{V}_A, V_S, P_A)$
$(\mathrm{ID}_A$, Smart card$)$
[Mutual authentication with session key agreement phase]
User A: Insert ID_A and PW_A
Smartcard: $V_A = \mathrm{PW}_A \bullet P$, check ID_A and $V_A = ?\overline{V}_A$
Smartcard: Selects a random no. $r_A, R_A = r_A \bullet P, M_A = r_A \bullet (V_A + V_S)$
$C_A = r_A \bullet (P_A + V_S)$, $\mathrm{DI}_A = \mathrm{ID}_A \oplus H(T_A \| r_A \bullet V_A \| r_A \bullet \mathrm{PW}_A \bullet V_A)$
$(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$
Checks $T_A^* - T_A \le \Delta T$ and
Computes $r_A \bullet V_A = M_A - d_S \bullet R_A$, $r_A \bullet P_A = C_A - d_S \bullet R_A$
$\mathrm{ID}_A = \mathrm{DI}_A \oplus H(T_A \| r_A \bullet V_A \| d_S \bullet r_A \bullet V_A)$ & ID_A is valid.
$h_A = H(\mathrm{ID}_A \| T \| d_S)$, $h_A \bullet R_A = ?r_A \bullet P_A$
Choose r_S, $M_S = r_A \bullet V_A + r_S \bullet V_S$, $H_S = H(\mathrm{ID}_A \| r_A \bullet V_A \| r_S \bullet V_S \| T_S)$
(M_S, H_S, T_S)
Check $T_S^* - T_S \le \Delta T$, Computes $r_S \bullet V_S = M_S - r_A \bullet V_A$
$K = r_S \bullet d_S \bullet (r_A \bullet V_A) = r_A \bullet r_S \bullet d_S \bullet \mathrm{PW}_A \bullet P$
$H_S^* = H(\mathrm{ID}_A \| r_A \bullet V_A \| r_S \bullet V_S \| T_S)$, Checks $H_S^* = ?H_S$
$K = (r_A \bullet \mathrm{PW}_A) \bullet r_S \bullet V_S = r_A \bullet r_S \bullet d_S \bullet \mathrm{PW}_A \bullet P$
Session key $\mathrm{SK} = \mathrm{kdf}(\mathrm{ID}_A \| \mathrm{Trans} \| K)$, $\mathrm{Trans} = (M_A \| R_A \| C_A \| M_S \| H_S \| T_A \| T_S)$
[Password change phase]
User A: Insert ID_A and PW_A
Smartcard: Compute $V_A = \mathrm{PW}_A \bullet P$, check ID_A & $V_A = ?\overline{V}_A$
User A: Insert $\mathrm{PW}_{\mathbf{new}}$
Smartcard: Compute $V_{\mathbf{new}} = \mathrm{PW}_{\mathbf{new}} \bullet P$, replace $\overline{V}_A$ with $V_{\mathbf{new}}$
[Lost smartcard revocation phase]
User A,
$(\mathrm{ID}_A, \overline{V}_A$, Secret information$)$
Choose a new timestamp $T_{\mathbf{new}}$,
Computes $h_A'' = H(\mathrm{ID}_A \| T_{\mathbf{new}} \| d_S)$, $P_A'' = h_A'' \bullet P$
Smart card
$(\mathrm{ID}_A, \overline{V}_A, V_S, P_A'')$
$(\mathrm{ID}_A$, New smartcard$)$
Open channel
Secure channel

Fig. 1 Proposed remote user mutual authentication scheme

V. Analysis of the Proposed Scheme

In this section, we analyzed the proposed scheme in terms of security requirements and functional requirements. Our scheme supports traditional password-based remote user mutual authentication, and not only satisfies several security-related requirements, but also fulfills functional requirements.

1. Correctness analysis

Here, we proved the correctness of the proposed scheme, *i.e.* how the user A and the remote server S authenticate each other in the mutual authentication with session key agreement phase.

Theorem 1 In our scheme, both the user A and the remote server S correctly authenticates each other and generates a common session key in each session.

Proof From the Steps 2 and 3 of the mutual authentication with session key agreement phase, A computes $R_A = r_A \cdot P$ and

$$\begin{aligned} M_A &= r_A \cdot (V_A + V_S) \\ &= r_A \cdot (PW_A \cdot P + d_S \cdot P) \\ &= r_A \cdot PW_A \cdot P + r_A \cdot d_S \cdot P \end{aligned}$$

$$\begin{aligned} C_A &= r_A \cdot (P_A + V_S) \\ &= r_A \cdot (h_A \cdot P + d_S \cdot P) \\ &= r_A \cdot h_A \cdot P + r_A \cdot d_S \cdot P \end{aligned}$$

$$\mathrm{DI}_A = \mathrm{ID}_A \oplus H(T_A \,||\, r_A \cdot V_A \,||\, r_A \cdot PW_A \cdot V_S)$$

Then A sends the message $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ to S. From the Step 6 of the mutual authentication with session key agreement phase, S computes

$$\begin{aligned} M_A - d_S \cdot R_A &= r_A \cdot \mathrm{PW}_A \cdot P + r_A \cdot d_S \cdot P - d_S \cdot R_A \\ &= r_A \cdot \mathrm{PW}_A \cdot P + d_S \cdot r_A \cdot P - d_S \cdot r_A \cdot P \\ &= r_A \cdot \mathrm{PW}_A \cdot P \\ &= r_A \cdot V_A \end{aligned}$$

$$\begin{aligned} C_A - d_S \cdot R_A &= r_A \cdot h_A \cdot P + r_A \cdot d_S \cdot P - d_S \cdot R_A \\ &= r_A \cdot h_A \cdot P + r_A \cdot d_S \cdot P - d_S \cdot r_A \cdot P \\ &= r_A \cdot h_A \cdot P + r_A \cdot d_S \cdot P - r_A \cdot d_S \cdot P \\ &= r_A \cdot h_A \cdot P \\ &= r_A \cdot P_A \end{aligned}$$

Since $(r_A \cdot \mathrm{PW}_A) \cdot V_S = r_A \cdot \mathrm{PW}_A \cdot d_S \cdot P = r_A \cdot d_S \cdot \mathrm{PW}_A \cdot P = r_A \cdot d_S \cdot V_A$, we have

$$\begin{aligned} &\mathrm{DI}_A \oplus H(T_A \,||\, r_A \cdot V_A \,||\, r_A \cdot \mathrm{PW}_A \cdot V_S) \\ &\quad = \mathrm{ID}_A \oplus H(T_A \,||\, r_A \cdot V_A \,||\, r_A \cdot \mathrm{PW}_A \cdot V_S) \\ &\qquad \oplus H(T_A \,||\, r_A \cdot V_A \,||\, r_A \cdot \mathrm{PW}_A \cdot V_S) \\ &\quad = \mathrm{ID}_A \end{aligned}$$

According to the Step 7 of the mutual authentication with session key agreement phase, we have $h_A = H(\mathrm{ID}_A \,||\, T \,||\, d_S)$ and

$$\begin{aligned} h_A \cdot R_A &= h_A \cdot r_A \cdot P \\ &= r_A \cdot h_A \cdot P \\ &= r_A \cdot P_A \end{aligned}$$

That is, $h_A \cdot R_A = r_A \cdot P_A$ holds. Thus, S authenticates A. From the Step 8 of the mutual authentication with session key agreement phase, S computes

$$\begin{aligned} M_S &= r_S \cdot V_S + r_A \cdot V_A \\ &= r_S \cdot d_S \cdot P + r_A \cdot \mathrm{PW}_S \cdot P \end{aligned}$$

and $H_S = H(\mathrm{ID}_A \,||\, r_A \cdot V_A \,||\, r_S \cdot V_S \,||\, T_S)$, session key $\mathrm{SK} = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K)$, where $\mathrm{Trans} = (M_A \,||\, R_A \,||\, C_A \,||\, M_S \,||\, H_S \,||\, T_A \,||\, T_S)$ and $K = r_S \cdot d_S \cdot (r_A \cdot V_A) = r_S \cdot r_A \cdot d_S \cdot \mathrm{PW}_A \cdot P$. Now, S sends the message (M_S, H_S, T_S) to A.

From the Step 9 of the mutual authentication with session key agreement phase, A computes

$$\begin{aligned} M_A - r_A \cdot V_A &= r_S \cdot d_S \cdot P + r_A \cdot PW_A \cdot P \\ &\quad - r_A \cdot PW_A \cdot P \\ &= r_S \cdot d_S \cdot P \\ &= r_S \cdot V_S \end{aligned}$$

and $H_S^* = H(\mathrm{ID}_A \,||\, r_A \cdot V_A \,||\, r_S \cdot V_S \,||\, T_S)$. Thus, $H_S^* = H_S$ holds. A authenticates S and computes the session key as $\mathrm{SK} = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K)$, where $\mathrm{Trans} = (M_A \,||\, R_A \,||\, C_A \,||\, M_S \,||\, H_S \,||\, T_S)$ and $K = r_A \cdot PW_A \cdot (r_s \cdot V_S) = r_S \cdot r_A \cdot d_S \cdot PW_A \cdot P$. Therefore, A and S mutually authenticate each other and hold the same session key SK. Q.E.D.

2. Formal security analysis

In this subsection, we will discuss the formal security validation of the proposed protocol in the attack model defined in Section III. In a password-based authentication scheme, secure mutual authentication and session key agreement are two important aspects. We can say, the mutual authentication is achieved between the user and remote server if they generate the same session key after the secure mutual authentication and any outsiders cannot learn anything about the session key. The proposed scheme also benefits from the use of smartcard to protect the secret information for the authentication.

Theorem 2 Let $\mathcal{D}$ be the small password dictionary and PAP is the proposed password authentication protocol. Assume that $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{PAP}}(K)$ be the success probability of a polynomial time bounded adversary $\mathcal{A}$, who executes q_s times Send queries, q_e times Execute queries, q_r times Reveal queries, q_H times Hash queries, and q_k times kdf queries. Then the probability that $\mathcal{A}$ breaks the proposed password authentication scheme is

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{PAP}}(k) \le \frac{q_H^2 + \left(q_s + q_e\right)^2 + q_s^2 + q_r^2 + q_k^2}{2^{k-1}} + 2\cdot\left(q_k \cdot \mathrm{Adv}_{\mathcal{A}}^{\mathrm{CDH}}(k) + \frac{q_s}{\mathcal{D}}\right)$$

where $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CDH}}(k)$ is the success probability of $\mathcal{A}$ of solving the Computational Diffie-Hellman (CDH) problem within polynomial time bound.

Proof Let $\mathcal{A}$ tries to breach the proposed password authentication protocol, then we can construct an algorithm $\mathcal{C}$ that will solve the CDH problem with the help of $\mathcal{A}$. Based on the Ref. [37], we define the following games Game i ($i = 0,1,\cdots,5$), where $\mathcal{A}$ has no advantage. Each game addresses a different security aspect. For each game Game i, we define $S_i (i = 0,1,\cdots,5)$ as the event that $\mathcal{A}$ wins the authenticated key agreement-security in the Game i. Let F be an event that may occur during the $\mathcal{A}$'s execution such that F is detectable by $\mathcal{C}$, F is independent of S_i, Game i and Game $i+1$ are identical unless F occurs, then we have

$$\left|\Pr\left[S_{i+1}\right] - \Pr\left[S_i\right]\right| \le \Pr[F] \tag{2}$$

Game 0 This game corresponds to the real attack, in the random oracle model. By definition, we have

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{PAP}}(k) = \left|2\Pr\left[S_0\right] - 1\right| \tag{3}$$

Game 1 In this game, the random oracle H is simulated by maintaining a hash list L_H^{list}. This game is perfectly indistinguishable from the real execution of the protocol since the oracles including Execute, Reveal, Send, Corrupt, and Test are also simulated as done in the real attack (Figs. 2–7). Thus, we have

$$\Pr\left[S_1\right] = \Pr\left[S_0\right] \tag{4}$$

$\mathcal{C}$ maintains an initial-empty hash list L_H^{list} for the hash function H, which includes the tuples of the form (q,r), where $r = H(q)$. On a receiving a hash query for the input q, $\mathcal{C}$ searches the list L_H^{list} and returns the old value r if the same query asked earlier, otherwise chooses a number $r \in_R Z_p^*$ such that there is no tuple of the form $(\cdot,r)$ in L_H^{list} and returns it to $\mathcal{A}$. Then $\mathcal{C}$ inserts the tuple (q,r) to L_H^{list}.

Fig. 2 Simulation of H oracle

$\mathcal{C}$ maintains an initial-empty list $L_{\mathrm{kdf}}^{\mathrm{list}}$ for kdf, which includes the tuples of the form (q',r'), where $r' = \mathrm{kdf}(q')$. On a receiving a kdf query for the input q', $\mathcal{C}$ searches the list $L_{\mathrm{kdf}}^{\mathrm{list}}$ and returns the old value r' if the same query asked earlier, otherwise chooses a number $r' \in_R Z_p^*$ such that there is no tuple of the form $(\cdot,r')$ in $L_{\mathrm{kdf}}^{\mathrm{list}}$ and returns it to $\mathcal{A}$. Then $\mathcal{C}$ inserts the tuple (q',r') to $L_{\mathrm{kdf}}^{\mathrm{list}}$.

Fig. 3 Simulation of kdf oracle

Game 2 This game is identical with Game 1 except that the simulation is terminated if a collision occurs in the simulation of the transcripts $(\mathrm{ID}_A, M_A, C_A, R_A, T_A)$ and (M_S, H_S, T_S). Based on the birthday attack, probability of collisions of the simulation of H oracle is at most $q_H^2 / 2^k$. Similarly, the probability of collisions in the transcripts simulation is at most $(q_s + q_e)^2 / 2^k$. Because $H(T_A \,||\, x\cdot V_A \,||\, x\cdot PW_A \cdot V_S)$ and $H(\mathrm{ID}_A \,||\, x\cdot V_A \,||\, y\cdot V_A \,||\, T_S)$ was chosen randomly from a uniform distribution. Thus, we have

$$\left|\Pr\left[S_2\right] - \Pr\left[S_1\right]\right| \le \frac{q_H^2}{2^k} + \frac{\left(q_s + q_e\right)^2}{2^{\mathrm{k}}} \tag{5}$$

On a query Send(A^i,start), assume that A^i is in correct state, we proceed as follows:

Choose a number $x \in_R Z_p^*$, compute $R_A = x \cdot P$, $M_A = x \cdot (V_A + V_S)$, $C_A = r_A \cdot (P_A + V_S)$ and $\mathrm{DI}_A = \mathrm{ID}_A \oplus H(T_A \,||\, r_A \cdot V_A \,||\, r_A \cdot PW_A \cdot V_S)$.

This query returns $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ as answer.

On a query Send(S^i,$(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$), assume that S^i is in correct state, we proceed as follows:

Check $|T_A^* - T_A| \le \Delta T_A$. Compute $x \cdot V_A = M_A - d_S \cdot R_A$. $x \cdot P_A = C_A - d_S \cdot R_A$, $\mathrm{ID}_A = \mathrm{DI}_A \oplus H(T_A \,||\, x \cdot V_A \,||\, d_S \cdot x \cdot V_A)$. If either of $|T_A^* - T_A| \le \Delta T_A$ or ID_A is invalid, the server instance terminates without accepting. Also check $h_A \cdot R_A = ?(x \cdot P_A)$, if it does not hold, server instance terminates without accepting. Otherwise, choose a $y \in_R Z_p^*$, compute $M_S = y \cdot V_S + x \cdot V_A$, $H_S = H(\mathrm{ID}_A \,||\, x \cdot V_A \,||\, y \cdot V_S \,||\, T_S)$ and the session key $\mathrm{SK}_S = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K_S)$, where $\mathrm{Trans} = (M_A \,||\, R_A \,||\, C_A \,||\, M_S \,||\, H_S \,||\, T_A \,||\, T_S)$ and $K_S = y \cdot d_S \cdot (x \cdot V_A)$. This query returns (M_S, H_S, T_S) as answer.

On a Send(A^i,(M_S, H_S, T_S)), assume that A^i is in correct state, we proceed as follows:

Check the validity of $|T_S^* - T_S| \le \Delta T_S$ and the user instance terminates without accepting if it is invalid, otherwise compute $y \cdot V_S = M_S - x \cdot V_A$, $H_S^* = H(\mathrm{ID}_A \,||\, x \cdot V_A \,||\, y \cdot V_S \,||\, T_S)$. User instance also verifies whether $H_S^* = ?\,H_S$ holds. If it is, compute the session key $\mathrm{SK}_A = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K_A)$, where $\mathrm{Trans} = (M_A \,||\, R_A \,||\, C_A \,||\, M_S \,||\, H_S \,||\, T_A \,||\, T_S)$ and $K_A = x \cdot \mathrm{PW}_A \cdot (y \cdot V_S)$. Finally, user instance accepts and terminates.

Fig. 4 Simulation of Send query

On a query Execute(A^i,S^j), we simulate the Send query as follows:

$(\mathrm{DI}_A, M_A, C_A, R_A, T_A) \leftarrow$ Send(A^i,start)

$(M_S, H_S, T_S) \leftarrow$ Send(S^j,$(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$)

This query returns $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ and (M_S, H_S, T_S).

Fig. 5 Simulation of Execute query

On a query Reveal(P^i), we proceed as follows:

If the instance P has accepted, this query returns the session key SK.

Fig. 6 Simulation of Reveal query

On a Test(P^i) query, we proceed as follows:

Get SK from the Reveal(P^i) query and flip an unbiased coin b. If $b = 1$, return SK, otherwise return a random value.

Fig. 7 Simulation of Test query

Game 3 In this game, the simulations are aborted however, $\mathcal{A}$ may have been guessed the correct authentication values $H(T_A \,||\, x \cdot V_A \,||\, x \cdot PW_A \cdot V_S)$ and $H(\mathrm{ID}_A \,||\, x \cdot V_A \,||\, y \cdot V_A \,||\, T_S)$. Thus, Game 2 and **Game 3** are indistinguishable unless the server instance (or user instance) rejects a valid authentication value. Thus, we have

$$\left|\Pr[S_3] - \Pr[S_2]\right| \le \frac{q_s^2}{2^k} + \frac{q_r^2}{2^k} \tag{6}$$

Game 4 In this game, the session key is computed using the oracle kdf, so that the session key SK is completely independent from kdf, K_A and K_S. In Execute queries, one gets $\mathrm{SK}_S = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K_S)$ or $\mathrm{SK}_A = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K_A)$. Therefore, Game 3 and Game 4 are indistinguishable unless $\mathcal{A}$ queries kdf on $(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K_S)$ or $(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K_A)$, on the common value $(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K)$, (where $K = \mathrm{CDH}(x \cdot V_A, y \cdot V_S) = x \cdot y \cdot \mathrm{PW}_A \cdot d_S \cdot P$, since $V_A = \mathrm{PW}_A \cdot P$ and $V_S = d_S \cdot P$). In addition, whatever the guess bit b involved in Test query, the answer is random, and independent for all the sessions. Hence, $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{CDH}}(k) \ge (1/q_k)\,|\Pr[S_4] - \Pr[S_3]|$, *i.e.*, we have

$$|\Pr[S_4] - \Pr[S_3]| \le q_k \cdot \mathrm{Adv}_{\mathcal{A}}^{\mathrm{CDH}}(k) \tag{7}$$

Game 5 This game is identical with Game 4 except that in Test query this game is terminated if $\mathcal{A}$ asks a kdf query with $(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K)$. $\mathcal{A}$ can get the session key SK by kdf query with probability at most $q_k^2/2^k$. Thus we have

$$\left|\Pr[S_5] - \Pr[S_4]\right| \le \frac{q_k^2}{2^k} \tag{8}$$

If $\mathcal{A}$ does not make any kdf query with the correct input, it will not have any advantage in distinguishing the real session key from a random one and thus $|\Pr[S_5]| = 1/2$. In addition, if the Corrupt(A,2) query has been made, it implies that the password-corrupt query Corrupt(A,1) has not been made. The probability of $\mathcal{A}$ launching the off-line password guessing attack is $q_s / |\mathcal{D}|$. Adding the Eqs. (3)~(8), we have

$$\mathrm{Adv}_{\mathcal{A}}^{\mathrm{PAP}}(k) \le \frac{q_H^2 + (q_s + q_e)^2 + q_s^2 + q_r^2 + q_k^2}{2^{k-1}} + 2 \cdot \left(q_k \cdot \mathrm{Adv}_{\mathcal{A}}^{\mathrm{CDH}}(k) + \frac{q_s}{\mathcal{D}}\right)$$

Q.E.D.

3. Further security analysis

The proposed scheme mutually authenticates user and the remote server and generates a secure session key between them in each session and satisfies the other security notions such as server's spoofing attack, replay attack, parallel session attack, impersonation attack, known session-specific temporary information attack, *etc*. In addition, our scheme also achieves the properties of session key agreement protocol as indicated by Blake-Wilson[40]. We discussed all of the known security properties against our scheme.

Proposition 1 The proposed scheme can prevent server's spoofing attack.

Proof In this attack, an adversary can impersonate the remote server by fabricating the login messages and thereby to cheating a legal user. In order to impersonate S, the adversary $\mathcal{A}$ does as follows:

(a) $\mathcal{A}$ intercepts the login message $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ of previous session, where

$$\begin{aligned} M_A &= r_A \cdot (V_A + V_S) \\ &= r_A \cdot (V_A + d_S \cdot P) \\ &= r_A \cdot V_A + r_A \cdot d_S \cdot P \\ &= r_A \cdot V_A + d_S \cdot r_A \cdot P \\ &= r_A \cdot V_A + d_S \cdot R_A \end{aligned}$$

$$\begin{aligned} M_S &= r_S \cdot V_S + r_A \cdot V_A \\ &= r_S \cdot V_S + r_A \cdot \mathrm{PW}_S \cdot P \\ &= r_S \cdot V_S + \mathrm{PW}_S \cdot r_A \cdot P \\ &= r_S \cdot V_S + \mathrm{PW}_S \cdot R_A \end{aligned}$$

and $H_S = H(\mathrm{ID}_A \,||\, r_A \cdot V_A \,||\, r_S \cdot V_S \,||\, T_S)$.

(b) $\mathcal{A}$ tries to generate a forged response message of the form (M_S, H_S, T_S), where $M_S = r_S \cdot V_S + r_A \cdot V_A$ and $H_S = H(\mathrm{ID}_A \,||\, r_A \cdot V_A \,||\, r_S \cdot V_S \,||\, T_S)$.

(c) As $\mathcal{A}$ has no knowledge about PW_A and d_S, he cannot compute $r_A \cdot V_A$ and $r_S \cdot V_S$ from M_A and M_S due to the difficulties of ECFP. Thus, $\mathcal{A}$ cannot generate the forged message (M_S, H_S, T_S).

We can conclude that the server's spoofing attack is infeasible in our scheme. Q.E.D.

Proposition 2 The proposed scheme can prevent the replay attack.

Proof The replay attack means an adversary $\mathcal{A}$ tries to masquerade either A or S or both through the reuse of information obtained from a previous run protocol. For this purpose, $\mathcal{A}$ does as follows:

(a) Suppose that $\mathcal{A}$ intercepted the login message $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ of previous session and send it to S in the current session.

(b) As the timestamp validity condition $|T_A^* - T_A| \leq \Delta T_A$ is not true, S rejects the message $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$.

(c) By the same reason, $\mathcal{A}$ cannot impersonate S just by replaying the old message (M_S, H_S, T_S) to A.

Thus, our scheme withstands the replay attack. Q.E.D.

Proposition 3 The proposed scheme can prevent the parallel session attack.

Proof In this attack, an adversary can masquerade the user by generating a valid login message out of some previous session eavesdropped messages transmitted between the user and the remote server. The parallel session attack to be successful, the adversary $\mathcal{A}$ executes the following steps:

(a) Suppose that $\mathcal{A}$ captures A's login message $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$, which is sent to S, where $\mathrm{DI}_A = \mathrm{ID}_A \oplus H(T_A \,||\, r_A \cdot V_A \,||\, r_A \cdot PW_A \cdot V_S)$, $M_A = r_A \cdot V_A + V_S$ and $C_A = r_A \cdot (P_A + V_S)$.

(b) $\mathcal{A}$ captures S's response message (M_S, H_S, T_S), which is sent to A in response of the message $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$, where $M_S = r_S \cdot V_S + r_A \cdot V_A$ and $H_S = H(\mathrm{ID}_A \,||\, r_A \cdot V_A \,||\, r_S \cdot V_S \,||\, T_S)$.

(c) The timestamps T_A and T_S are incorporated in the messages $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ and (M_S, H_S, T_S), $\mathcal{A}$ cannot create fabricated login message $(\mathrm{DI}_A^{'}, M_A^{'}, C_A^{'}, R_A^{'}, T_A^{'})$ without PW_A and d_S as discussed in Proposition 1.

Therefore, the proposed scheme resists the parallel session attack. Q.E.D.

Proposition 4 The scheme can prevent the impersonation attack.

Proof The impersonation attack means, an adversary $\mathcal{A}$, who does not know the secrets of $\mathcal{A}$ and S, can try to fabricate the messages exchanged between S and $\mathcal{A}$ to impersonate them. For this purpose $\mathcal{A}$ does as follows:

(a) Suppose that $\mathcal{A}$ intercepts the messages $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ and (M_S, H_S, T_S).

(b) From Pproposition 3, we show that $\mathcal{A}$ cannot create a forged login message for the fresh timestamps $T_A^{'}$ and $T_S^{'}$ without PW_A and d_S. Hence $\mathcal{A}$ cannot impersonate A.

(c) In other way, $\mathcal{A}$ can theft A's smartcard and extract $(\overline{V_A}, P_A, V_S)$, where $V_A = \overline{V_A} = \mathrm{PW}_A \cdot P$. Now $\mathcal{A}$ choose a random number $r \in_R Z_p^*$, compute $R_A = r \cdot P$, $M_A = r \cdot (V_A + V_S)$ and $C_A = r \cdot (P_A + V_S)$. However, he cannot compute $\mathrm{DI}_A = \mathrm{ID}_A \oplus H(T_A \,||\, r \cdot V_A \,||\, r \cdot \mathrm{PW}_A \cdot V_S)$ because $PW_A \cdot V_S$ cannot be computed from the pair $(\overline{V_A}, V_S)$ due to CDHP and hence $\mathcal{A}$ cannot generate a forged login message $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ to impersonate S.

Thus, our scheme protects the impersonation attack. Q.E.D.

Proposition 5 The proposed scheme can withstand the Denial of Service (DoS) attack and stolen-verifier attack.

Proof The server closes a login session if the number of login attempts of an account with an incorrect password exceeds a limit value. Even so, such a user's account is still workable and later login requests will pass as long as correct password is provided.

(a) In our scheme, no password-verifier table is stored at server's database and thus, DoS attack is infeasible. In other words, $\mathcal{A}$ may theft A's smartcard, but cannot change the password stored on it without supplying the correct password.

(b) If $\mathcal{A}$ thefts A's smartcard and extract the secret information $(\overline{V_A}, P_A, V_S)$, where $\overline{V_A} = \mathrm{PW}_A \cdot P$, however he cannot extract PW_A from $\overline{V_A}$ due to ECDLP and accordingly stolen-verifier attack is not possible.

The proposed scheme resists DoS attack and stolen-verifier attack. Q.E.D.

Proposition 6 Proposed scheme can provide the perfect forward security.

Proof Perfect forward secrecy means, if the password PW_A and the private key d_S are known to an adversary $\mathcal{A}$, but all the previous session keys should be secured from the adversary. Suppose that PW_A and d_S are known to $\mathcal{A}$, he can execute the following steps in order to break the perfect forward security:

(a) $\mathcal{A}$ intercepts the message $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$, where $M_A = r \cdot (V_A + V_S)$ and $C_A = r \cdot (P_A + V_S)$.

(b) $\mathcal{A}$ computes $r_A \cdot V_A = M_A - d_S \cdot R_A$ and $r_S \cdot V_S = C_A - PW_A \cdot R_A$.

(c) However, $\mathcal{A}$ cannot compute the session key $\mathrm{SK} = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K)$ since $K = r_S \cdot d_S \cdot \mathrm{PW}_A \cdot P$ is still unknown to him.

(d) $\mathcal{A}$ may try to compute K from the pair $(r_A \cdot V_A, r_S \cdot V_S)$ directly, but it is also computationally infeasible of CDHP.

Thus, the perfect forward secrecy is achieved in our scheme. Q.E.D.

Proposition 7 The proposed scheme can resist the known session-specific temporary information attack.

Proof This attack[35,41,42] states that if the session short-term secrets are leaked accidentally to an adversary, however the past or future session keys should be secured from this disclosure. In our scheme, both A and S compute the session key SK.

(a) Suppose that $\mathcal{A}$ knows r_A and r_S.

(b) $\mathcal{A}$ can compute $r_A \cdot V_A = M_A - r_A \cdot V_S$ and $r_S \cdot V_S$, where V_S is the public key of S.

(c) Then $\mathcal{A}$ can compute (V_A, V_S) from the pair $(r_A \cdot V_A, r_S \cdot V_S)$ since (r_A, r_S) is known to him.

(d) However, from the pair $(V_A, V_S) = (\mathrm{PW}_A \cdot P, d_S \cdot P)$, $\mathcal{A}$ cannot compute $d_S \cdot \mathrm{PW}_A \cdot P$ and thus $K = r_S \cdot d_S \cdot \mathrm{PW}_A \cdot P$ as well due to CDHP. Hence the session key $\mathrm{SK} = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K)$ is still unknown to $\mathcal{A}$.

Thus, the known session-specific temporary information attack is hard in our scheme. Q.E.D.

Proposition 8 The proposed scheme can resist the key-compromise impersonation attack.

Proof This attack states that the adversary who knows the password of the user should not be able to impersonate the remote server.

(a) Assume that PW_A is known to $\mathcal{A}$.

(b) $\mathcal{A}$ computes $\mathrm{PW}_A \cdot (d_S \cdot P)$ using PW_A and S's public key $V_S = d_S \cdot P$.

(c) In order to compute SK, $\mathcal{A}$ must generates K. However, $\mathrm{SK} = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K)$ can be computed either (r_A, r_S) or S's private key d_S is known to $\mathcal{A}$.

(d) However, $\mathcal{A}$ cannot extract (r_A, r_S) from (M_A, M_S) or d_S from V_S due to ECDLP.

Thus, our scheme resists the key-compromise impersonation attack. Q.E.D.

Proposition 9 The proposed scheme can resist the privileged-insider attack.

Proof It is difficult to remember a long password, if it is not frequently used. The user A may register into many servers with the same password and login identity. If the password is revealed to a privilege-insider $\mathcal{A}$ of S, then he may impersonate the legal user and can access other servers where the user is registered as a valid client.

(a) In the registration phase, A registers to S with $\overline{V_A} = \mathrm{PW}_A \cdot P$. Therefore, A's password PW_A is unknown to the insider $\mathcal{A}$ of S.

(b) Due to difficulties of ECDLP, insider $\mathcal{A}$ cannot figure out the password PW_A from $\overline{V_A}$.

Thus, the risk of impersonation due to privilege-insider attack is not possible. Q.E.D.

Proposition 10 The proposed scheme can protect unknown key-share attack and reflection attack.

Proof As discussed in the Ref. [43], the session key will provides the freshness and data origin authentication, and free from reflection attack and unknown key-share attack if the protocol transcript is incorporated in the key derivation function kdf. Thus, we have used user identity and transmitted messages $\mathrm{Trans} = (M_A \,||\, R_A \,||\, C_A \,||\, M_S \,||\, H_S \,||\, T_A \,||\, T_S)$ and $K = r_S \cdot d_S \cdot \mathrm{PW}_A \cdot P$ for the session key $\mathrm{SK} = \mathrm{kdf}(\mathrm{ID}_A \,||\, \mathrm{Trans} \,||\, K)$ generation. Accordingly, the session key of our scheme resists the unknown key-share attack and reflection attack.

Q.E.D.

Now a comparative study is given against different security attributes of the proposed scheme with other schemes[12, 13, 17, 22, 24, 28, 29, 30] in Tab. 2.

Tab. 2 Security comparisons of the proposed scheme with other existing schemes

Scheme	Attacks								
	Server spoofing	Password guessing	DoS	Impersonation	Reflection	Insider	Known-key	Random password	Known session-specific temporary information
Ref. [12]	Yes	Yes	Yes	No	No	Yes	Yes	No	No
Ref. [13]	No	No	No	No	Yes	No	Yes	No	No
Ref. [17]	No	Yes	Yes	No	Yes	No	Yes	Yes	No
Ref. [22]	Yes	No	No	Yes	Yes	Yes	No	Yes	No
Ref. [24]	No	No	No	No	Yes	No	Yes	Yes	No
Ref. [28]	Yes	NA*	Yes	No	Yes	Yes	Yes	NA*	No
Ref. [29]	Yes	NA*	Yes	Yes	Yes	Yes	Yes	NA*	No
Ref. [30]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Proposed	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

No: do not protect the attack. Yes: protect the attack.

*These schemes are not password based scheme.

4. Efficiency analysis

In this section, efficiency analysis of the proposed scheme is discussed. The following fundamental requirements are required for an efficient smartcard based remote user mutual authentication scheme.

(1) User's anonymity

User's anonymity is one of the security aspects of an efficient remote login system. The proposed scheme preserves user's anonymity in all aspects, since in the Step 2 of mutual authentication phase, instead the original identity ID_A, a dynamic identity DI_A is sent and it is masked by $H(T_A \,||\, r_A \cdot V_A \,||\, r_A \cdot \mathrm{PW}_A \cdot V_S)$. Note that the adversary cannot extract ID_A from DI_A since r_A, d_S, $H(T_A \,||\, r_A \cdot V_A \,||\, r_A \cdot \mathrm{PW}_A \cdot V_S)$ and PW_A are unknown to him. Therefore, user's anonymity is preserved.

(2) Mutual authentication

Mutual authentication helps to withstand server's spoofing attack where an attacker pretends to be the server to manipulate sensitive data of the legal users. In our scheme, S first authenticates A after validating $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ and then

A authenticates S by analyzing (M_S, H_S, T_S). However, the attacker cannot cheat A and S without PW_A and d_S. Thus, a secure mutual authentication is proposed in our scheme.

(3) Session key agreement

In simple remote login system, mutual authentication may accomplish the necessary security requirements, but in some applications (*i.e.* e-voting, online money transaction, pay-TV, *etc.*) where confidential data exchange between the user and the remote server, a session key agreement is also necessary. In our scheme, A sends $(\mathrm{DI}_A, M_A, C_A, R_A, T_A)$ to S, after validating it S generates the session key SK and sends (M_S, H_S, T_S) to A. A checks the legality of S based upon the correctness of (M_S, H_S, T_S) and then computes the same session key SK. Thus, the proposed scheme supports secure session key generation.

(4) User can choose password freely

In some remote login scheme, the password is chosen by the remote server not by the user, which is not a case in real-life applications (*e.g.*, email subscription, online banking, e-voting, *etc.*). Furthermore, a privileged-insider of the remote server may theft the user password from the server's database and can impersonate the legal user. In our scheme, users can choose their password freely without any agreement from the remote server and no password verification table is stored in the server database. So the privileged-insider attack could not happen in the proposed scheme.

(5) Secure password change

In remote login scheme, for security purpose user can change his password periodically. In our scheme, user is free to change his password and update the information stored in the smartcard without server's agreement. Since server's participation is not required in password change phase, so the new password cannot be revealed to a privileged-insider.

(6) No verification table

An attacker may steal the password-verifier from the server's database and may impersonate a legal user to login to the remote server using stolen-verifier. The proposed scheme is free from the stolen verifier attack. Since, there is no such verification table stored on the server, by which an adversary can make a fabricated login request to impersonate a legal user to login to the remote server, or can impersonate the server to cheat the legal user.

(7) Minimum number of server's secret

In a remote login scheme, user and server achieve mutual authentication, session key agreement and other security requirements by using their secrets. The remote server has to pay more maintenance cost if the number of the secret key of the server is increased. In the schemes[12,13,17,22,24], the remote server keeps two secrets that puts a burden on the system. In common practice, it is better to achieve the desired security by using only one secret key. In our scheme, server keeps only one secret.

We presented a comparative study of different functional requirements of some existing schemes[12,13,17,22,24,28–30] with the proposed scheme and summarized the results in Tab. 3.

(8) Computation cost

An efficient mutual authentication scheme should have lower computation in order to speed the execution of the protocol. To evaluate the computational efficiency, we compared our scheme with the ECC-based schemes[28–30]. The Table 4 illustrated the computation cost analysis among existing schemes. For convenience, we considered the registration phase and mutual authentication with session key agreement phase for efficiency analysis. Anyone can assumed that our scheme required more computational costs than the existing schemes[28–30] in order to acquired the better security. However, the schemes[28–30] used a hash function, called map-to-point hash function and its computational cost is more than an elliptic curve point multiplication, but the simple cryptographic hash function is used in our scheme. Thus, the overall computation cost of the proposed scheme is lower than the schemes[28–30]. In addition, the schemes[28–30] are static ID-based remote user authentication, while the proposed scheme is a dynamic ID-based authentication. Furthermore, the schemes[28–30] are not secured, whereas our scheme is free from all known attacks.

Tab. 3 Security comparisons of the proposed scheme and other schemes

Schemes	Mutual authentication	Session key agreement	Perfect forward secrecy	User's anonymity	Revocation of lost smartcard	Secure password change	No verification table	Password chosen by user	No. of server's secret
Ref. [12]	Yes	No	NA*	Yes	No	No	Yes	Yes	Two
Ref. [13]	No	No	NA*	No	No	No	Yes	No	Two
Ref. [17]	Yes	No	NA*	No	No	Yes	Yes	No	Two
Ref. [22]	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Two
Ref. [24]	No	Yes	No	Yes	No	Yes	Yes	Yes	Two
Ref. [28]	Yes	Yes	No	Yes	NA**	Yes	Yes	NA***	One
Ref. [29]	Yes	Yes	No	No	NA*	Yes	Yes	NA***	One
Ref. [30]	Yes	Yes	No	No	No	Yes	Yes	Yes	One
Proposed	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	One

*These schemes do not support session key agreement phase.

**These schemes do not support revocation of lost smartcard.

***These schemes are not password based scheme.

Tab. 4 Computation cost comparisons of the proposed scheme with other schemes

Schemes	Phases		Total cost
	Registration	Authentication	
Ref. [28]	PM + H	8PM +5PA + 8H	9PM + 5PA + 9H
Ref. [29]	PM + H	7PM + 4PA + 12H	8PM + 4PA + 13H
Ref. [30]	PM +4 H	8PM + 4PA + 11H	9PM + 4PA + 15H
Proposed	2PM + H	7PM + 5PA + 5H	9PM+ 5PA + 6H

Note: PM: elliptic curve scalar multiplication.
PA: elliptic curve point addition/subtraction.
H: Hash operation.

VI. Conclusion

In this paper, we proposed a dynamic ID-based remote user mutual authentication scheme using password and smartcard on elliptic curve cryptosystem. The primary merit of our scheme is that it supports mutual authentication with secure session key agreement phase and lost smartcard revocation phase. In addition, in our scheme, an user can choose and change his password freely and no verification table is needed for authentication. Compared with other schemes, proposed scheme achieves more functionally and protects all relevant attacks. The proposed scheme is thus more efficient, secure and flexible than other existing schemes.

References

[1] L. Lamport. Password authentication with insecure communication. *Communications of the ACM*, **24** (1981)11, 770–772.

[2] C. L. Lin and T. Hwang. A password authentication scheme with secure password updating. *Computers and Security*, **22**(2003)1, 68–72.

[3] E. J. Yoon, E. K. Ruy, and K. Y. Roo. A secure user authentication scheme using hash functions. *ACM Operating Systems Review*, **38**(2004)2, 62–68.

[4] M. Peyravian and C. Jeffries. Secure remote user access over insecure networks. *Computer Communications*, **29**(2006)5, 660–667.

[5] M. S. Hwang and L. H. Li. A new remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronic*, **46**(2000)1, 28–30.

[6] M. Kumar. New remote user authentication scheme with smart cards. *IEEE Transactions on Consumer Electronics*, **50**(2004)2, 597–600.

[7] R. Lu and Z. Cao. Efficient remote user authentication scheme using smart card. *Computer Networks*, **49** (2005)4, 535–540.

[8] H. T. Liaw, J. F. Lin, and W. C. Wu. An efficient and complete remote user authentication scheme using smart cards. *Mathematical and Computer Modelling*, **44**(2006)1–2, 223–228.

[9] I. E. Liao, C. C. Lee, and M. S. Hwang. A password authentication scheme over insecure networks. *Journal of Computer and System Sciences*, **72**(2006)4, 727–740.

[10] M. L. Das, A. Saxena, V. P. Gulati, and D. B. Phatak.

A novel remote client authentication protocol using bilinear pairings. *Computer and Security*, **25**(2006)3, 184–189.

[11] T. Goriparthi, M. L. Das, and A. Saxena. An improved bilinear pairing based remote user authentication scheme. *Computer Standards and Interfaces*, **31**(2009), 181–185.

[12] I. E. Liao, C. C. Lee, and M. S. Hwang. Security enhancement for a dynamic ID-based remote user authentication scheme. Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP'05), 2005, Seoul, Korea, 22–26.

[13] M. L. Das, A. Saxena, and V. P. Gulati. A dynamic ID-based remote user authentication scheme. *IEEE Transactions on Consumer Electronics*, **50**(2004)2, 629–631.

[14] W. C. Ku, and S. T. Chang. Impersonation attack on a dynamic ID-based remote user authentication scheme using smart cards. *IEICE Transactions on Communications*, **E88**(2005)B5, 2165–2167.

[15] X. Zhang, Q. Feng, and M. Li. A modified dynamic ID-based remote user authentication scheme. Proceedings of the International Conference on Communications, Circuits and Systems, Vol. 3, 2006, Guilin, China, 1602–1604.

[16] Y-C. Lee, G-K. Chang, W-C. Kuo, and J-L. Chu. Improvement of the dynamic ID-based remote user authentication scheme. Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, China, 2008, 3283–3287.

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

[18] Y-F. Chang and H-C. Chang. Security of dynamic ID-based remote user authentication scheme. Proceedings of the Fifth International Joint Conference on INC, IMC, IDC, 2009, Seoul, Korea, 2108–2110.

[19] H. Y. Chien, and C. H. Chen. A remote authentication scheme preserving user anonymity. Proceedings of the 19th International Conference on Advanced Information Networking and Applications, Vol. 2, 2005, Taiwan, China, 245–248.

[20] A. K. Awasthi. Comments on a secure dynamic ID-based remote user authentication scheme. *Transaction on Ccryptology*, **1**(2004)2, 15–16.

[21] M. A. Ahmed, D. R. Lakshmi, and S. A. Sattar. Cryptanalysis of a more efficient and secure dynamic ID-based remote user authentication scheme. *International Journal of Network Security & Its Applications*, **1**(2009)3, 32–37.

[22] M. K. Khan, S. K. Kim, and K. Alghathbar. Cryptanalysis and security enhancement of a'more efficient & secure dynamic ID-based remote user authentication scheme. *Computer Communications*, **34**(2009)3, 305–309.

[23] D. He, J. Chen, and J. Hu. Weaknesses of a dynamic ID-based remote user authentication scheme. Cryptology ePrint Archive: Report 2010/314, 2010.

[24] Y. P. Liao and S. S. Wang. A secure dynamic ID-based remote user authentication scheme for multi-server environment. *Computer Standards & Interfaces*, **31** (2009)1, 24–29.

[25] H. C. Hsiang, and W. K. Shih. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standards and Interfaces*, **31**(2009)6, 1118–1123.

[26] M. H. Shao and Y. C. Chin. A novel approach to dynamic ID-based remote user authentication scheme for multi-server environment. Proceedings of the 4th International Conference on Network and System Security, 2010, Melbourne, Austrelia, 548–553.

[27] Z. Tan. Cryptanalysis of two ID-based password authentication schemes for multi-server environments. *International Journal of Digital Content Technology and its Applications*, **5**(2011)1, 87–94.

[28] J. H. Yang and C. C. Chang. An ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. *Computes and Security*, **28**(2009)3–4, 138–143.

[29] E-J. Yoon and K-Y. Yoo. Robust ID-based remote mutual authentication with key agreement scheme for mobile devices on ECC. Proceedings of the International Conference on Computational Science and Engineering, 2009, Vancouver, Canada, 633–640.

[30] T-H. Chen, Y-C. Chen, and W-K. Shih. An advanced ECC ID-based remote mutual authentication scheme for mobile devices. Proceedings of the 7th International Conference on Ubiquitous, Autonomic and Trusted Computing, 2010, Xi'an, Shaanxi, China, 116–120.

[31] S. H. Islam and G. P. Biswas. A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem. *Journal of Systems and Software*, **84**(2011)11, 1892–1898.

[32] V. S. Miller. Use of elliptic curves in cryptography. in: Proceedings of the Proceeding on Advances in Cryptology (Crypto'85), Springer-Verlag, LNCS, Vol. 218, New York, USA, 1985, 417–426.

[33] N. Koblitz. Elliptic curve cryptosystem. *Mathematics of Computation*, **48**(1987)177, 203–209.

[34] D. Hankerson, A. Menezes, and S. Vanstone. Guide to elliptic curve cryptography. Springer-Verlag, New York, USA, 2004.

[35] S. H. Islam and G. P. Biswas. A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks. *Annals of Telecommunications*, **67**(2012)11–12, 547–558.

[36] S. H. Islam and G. P. Biswas. Provably secure and pairing-free certificateless digital signature scheme using elliptic curve cryptography. *International Journal of Computer Mathematics*, **90**(2013)11, 2244–2258.

[37] V. Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archieve, Report 2004/332, 2004.

[38] M. Bellare, and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. Proceedings of the 1st ACM Conference on Computer and Communications Security, 1993, Fairfax, VA, USA, 62–73.

[39] J. Xu, W-T. Zhu and D-G. Feng. An improved smartcard based password authentication scheme with provable security. *Computer Standard and Interfaces*, **32**(2009)4, 723–728.

[40] S. Blake-Wilson, D. Johnson, and A. Menezes. Key agreement protocols and their security analysis. Proceedings of the 6th IMA International Conference on Cryptography and Coding, Springer-Verlag, LNCS, Vol. 1335, 1987, 30–45.

[41] R. Canetti and H. Krawczyk. Analysis of key exchange protocols and their use for building secure channels. Proceedings of the Advances in Cryptology (Eurocrypt'01), Springer-Verlag, LNCS, Vol. 2045, 2001, 453–474,

[42] Z. Cheng, M. Nistazakis, R. Comley, and L. Vasiu. On the indistinguishability-based security model of key agreement protocols-simple cases. Cryptology ePrint Archieve, Report 2005/129, 2005.

[43] S. Wang, Z. Cao, K-K. R. Choo, and L. Wang. An improved identity-based key agreement protocol and its security proof. *Information Sciences*, **179**(2009)3, 307–318.