Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 247836, 11 pages
http://dx.doi.org/10.1155/2014/247836

Research Article

Analysis and Enhancement of a Password Authentication and Update Scheme Based on Elliptic Curve Cryptography

Lili Wang

Department of Computer Science, Dezhou University, Dezhou 253023, China

Correspondence should be addressed to Lili Wang; dzxywll@163.com

Received 21 December 2013; Revised 20 March 2014; Accepted 21 March 2014; Published 15 April 2014

Academic Editor: Gongnan Xie

Recently, a password authentication and update scheme has been presented by Islam and Biswas to remove the security weaknesses in Lin and Huang's scheme. Unfortunately, He et al., Wang et al., and Li have found out that Islam and Biswas' improvement was vulnerable to offline password guessing attack, stolen verifier attack, privilege insider attack, and denial of service attack. In this paper, we further analyze Islam and Biswas' scheme and demonstrate that their scheme cannot resist password compromise impersonation attack. In order to remedy the weaknesses mentioned above, we propose an improved anonymous remote authentication scheme using smart card without using bilinear paring computation. In addition, the verifier tables are no longer existent, and the privacy of users could be protected better. Furthermore, our proposal not only inherits the advantages in Islam and Biswas' scheme, but also provides more features, including preserving user anonymity, supporting offline password change, revocation, reregistration with the same identifier, and system update. Finally, we compare our enhancement with related works to illustrate that the improvement is more secure and robust, while maintaining low performance cost.

1. Introduction

With the fast development of communication terminals and networks, users could obtain lots of services distributed over the world, whenever and wherever. Nevertheless, more and more security issues prevent the advanced technologies from moving forward, and more and more people start to concern about the security problems of their information and communication applications. In detail, how to access the remote server securely is concerned by all users as a key issue. Generally speaking, the first line of defense for remote communication systems is authentication, which permits the legal users to obtain their desired services securely, while it rejects the illegal users to access to the servers. After that, to guarantee private communications over the insecure public networks, key agreement provides us the session keys, which are used to encrypt and decrypt the subsequent information transmitted over public channels (e.g., the Internet and radio). In other words, authentication and key agreement plays important roles in guaranteeing the security of the information and communication systems. In this paper, we will focus on the remote authentication and private communication.

Due to the property of easy-to-memory, the password has become the most popular and widely adopted method for authentication, since Lamport's [1] contributions on remote authentication using hash function in 1981. However, the convenient property leads to the weakness of low entropy, which can be the target for adversaries to attack, for example, password guessing (online or offline) attacks [2] and verifier stolen attacks. In addition, password-verifier tables are heavy burdens for servers to store and manage. Furthermore, password-verifier tables are threatened by the attackers, who can compromise these verifier tables and reveal (guess) user's password or masquerade as the legal user. In 2000, Peyravian and Zunic [3] presented one method for protecting and changing passwords in authentication schemes while being transmitted over untrusted networks [4]. Their scheme did not use any symmetric-key or public-key cryptosystems but only employed a collision resistant hash function. In 2002, Hwang and Yeh [5] pointed out that the scheme in [3] was vulnerable to guessing attack, server spoofing, and data eavesdropping attack, and they also proposed two improved schemes to enhance the security of the scheme in [3]. Later on, Lin and Hwang [6] cryptanalyzed the improved schemes in [5] and showed that their improvements were vulnerable to a denial of service attack and did not provide the forward secrecy property in session key distribution. Moreover, Lin and Hwang fixed the schemes in [5] to avoid those problems. Actually for many applications, the authentication schemes, which are based on the password (as the only authentication factor), are insufficient; therefore smart card (as the second authentication factor) based on remote user password authentication schemes [7–9] has been proposed to overcome the vulnerabilities caused by the low-entropy password and verifier tables. In 2011, Hafizul Islam and Biswas [10] designed a password authentication and update scheme based on elliptic curve cryptography as an improvement of Lin and Hwang's [6] scheme, which was demonstrated to be vulnerable to password guessing attack, insider attack, server spoofing attack, and data eavesdropping attack. Unfortunately, He [11] and Wang et al. [12] found out that their improved scheme was not secure as they claimed in [10] and several attacks were demonstrated effectively in [10], for example, offline password guessing attack, stolen verifier attack, privilege insider attack, and denial of service attack. Recently, Li [13] also pointed out that Hafizul Islam and Biswas' [10] scheme was vulnerable to offline password guessing attack, stolen verifier attack, and insider attack. Li presented an advanced smart card-based scheme using bilinear paring computation while providing an anonymous version.

In this paper, we further analyze the scheme in [10] and point out that the scheme is insecure to resist password compromise impersonation (PCI) attack [14–16]. Furthermore, the comments on the existing attacks suggest that we should pay attention to the low-entropy password, avoid using the weak password-verifier table, and take the advantages of the challenge-response mechanism properly, so as to prevent the scheme from being vulnerable to various attacks. In addition, the public key cryptosystem increases the performances cost for users and servers; for example, users should maintain and verify the servers' public keys (certificates) and servers should store users' password verifiers. In order to overcome the shortcomings in [10], we focus on designing an improved password authentication and update scheme. Our improvement is based on the secure one-way function, symmetric encryption/decryption, pseudorandom generator, and elliptic curve cryptosystem without the expensive bilinear paring computation. Finally, our proposal satisfies and achieves the following requirements and goals in the environment of symmetric key cryptosystem.

RG1: Mutual Authentication. Client and server can securely authenticate each other with their own credentials (secret key and verifier table for server, password and smart card for user). In other words, anyone else cannot impersonate any of the legal participants to cheat the intended partners. In detail, the scheme should be secure to resist known common attacks, which can threat the security of mutual authentication, for example, replay attack, reflection attack, parallel session attack, man-in-the-middle attack, known session key attack, forgery attack, and password compromise impersonation attack.

RG2: Session Key Distribution. The legal participants in the scheme should generate a secure session key. In addition, the session key should be only shared between the participants and anyone else could not reveal it. Furthermore, the session key should be generated fresh with key privacy, forward secrecy, and out of key control.

RG3: Password Change. Users can change their passwords securely and freely without interacting with the remote server; that is, users could securely change their passwords offline.

RG4: Revocation and Reregistration. Users can revoke their credentials for some secure concerns and reregister without changing their identifiers in the same server.

RG5: System Update. The master key of the server should be changed termly for security or system update.

RG6: Credentials Leakage Resistant. For users, the password should be protected securely to resist various kinds of guessing attacks launched by insider users, servers, or adversaries. For servers, there are no verifier tables stored in its database to resist verifier-stolen attack or insider server attack.

RG7: Denial of Service Resistance. The server should provide the mechanism to resist the denial of service (DoS) attack caused by exhausted resources (computation, memory, or connection) and malicious password change.

RG8: Preserving User Anonymity. The user's identifier should be protected from being hijacked or theft, because the user's privacy will be concerned in most applications, and any one cannot obtain the user's identifier except the legal participants.

In the rest of the paper, we briefly review Hafizul Islam and Biswas' scheme [10] in Section 2. The analysis and comments on their scheme are presented in Section 3. Furthermore, an improved scheme is proposed in Section 4. In addition, the analysis, comparison, and comments of our proposal are shown in Section 5. The paper si concluded in Section 6. Finally, notations used in this paper are shown in Notations section.

2. Review

In this section, the scheme of Hafizul Islam and Biswas [10] is reviewed in brief. There are four phases in Hafizul Islam and Biswas' [10] scheme, including registration phase, password authentication phase, password change phase, and session key distribution phase. The details of their scheme are described as follows.

2.1. Registration Phase. The client A registers to the server S with identity ID_A and password verifier $U_A = \mathrm{pw}_A \cdot G$ and collects the server's public key $U_S = d_S \cdot G$. Then, S stores each legal client's identity ID_A, password-verifier U_A, and a status-bit in a write protected file, where the status-bit indicates the status of the client in the server (logged-in or logged-off).

2.2. Password Authentication Phase

Step A1. A keys ID_A and pw_A into the terminal. A selects a random number $r_A \in [1, q-1]$, computing

$$R_A = r_A \cdot U_S, \qquad W_A = (r_A \cdot \mathrm{pw}_A) \cdot G, \qquad E_{K_x}(\mathrm{ID}_A, R_A, W_A), \tag{1}$$

where the symmetric key K_x is the x-coordinate of $K = \mathrm{pw}_A \cdot U_S = \mathrm{pw}_A \cdot d_S \cdot G = (K_x, K_y)$. Finally, A sends the login request message,

$$\{\mathrm{ID}_A, E_{K_x}(\mathrm{ID}_A, R_A, W_A)\}, \tag{2}$$

to the remote server.

Step A2. S checks the validity of ID_A and computes its corresponding decryption keys K_x by calculating

$$K = d_S \cdot U_A = \mathrm{pw}_A \cdot d_S \cdot G = (K_x, K_y). \tag{3}$$

After decrypting

$$E_{K_x}(\mathrm{ID}_A, R_A, W_A), \tag{4}$$

S compares received ID_A with decrypted ID_A and $\hat{e}(R_A, U_A)$ with $\hat{e}(W_A, U_S)$. If all the conditions are satisfied, S selects a random number r_S and computes

$$W_S = r_S \cdot U_S = r_S \cdot d_S \cdot G. \tag{5}$$

At last, S sends its response message,

$$\{W_A + W_S, H(W_S)\}, \tag{6}$$

to the client.

Step A3. A retrieves W_S by subtracting W_A from $W_A + W_S$. If the hash value of retrieved W_S is equal to received $H(W_S)$, A computes

$$H(W_A, W_S) \tag{7}$$

and sends it to the remote server.

Step A4. S computes

$$H(W_A, W_S), \tag{8}$$

with its own copies of W_A and W_S and compares the results with the received $H(W_A, W_S)$. If they are equal, S accepts the client's login request, otherwise rejects.

2.2.1. Password Change Phase

Step C1. $A \rightarrow S : \{\mathrm{ID}_A, E_{K_x}(\mathrm{ID}_A, R_A, W_A)\}$.

Step C2. $S \rightarrow A : \{W_A + W_S, H(W_S)\}$.

Step C3. $A \rightarrow S : \{\mathrm{ID}_A, H(W_A, W_S), W_A + U'_A, H(W_S, U'_A)\}$.

Step C4. $S \rightarrow A$: password change granted/denied.

If A wants to change the old password pw_A to a new one pw'_A, then A computes the corresponding password verifier $U'_A = \mathrm{pw}'_A \cdot G$ in Step C3. If the authentication token $H(W_A, W_S)$ is authenticated, then S subtracts W_A from $W_A + U'_A$ to extract the new password verifier U'_A. Finally, S replaces U_A with U'_A to finish the password change phase if and only if the hash value of (W_S, U'_A) is equal to received $H(W_S, U'_A)$.

2.3. Session Key Distribution Phase

Step D1. $A \longrightarrow S : \{\mathrm{ID}_A, E_{K_x}(\mathrm{ID}_A, R_A, W_A)\}$.

Step D2. $S \longrightarrow A : \{W_A + W_S, H(W_S)\}$.

Step D3. $A \longrightarrow S : \{\mathrm{ID}_A, H(W_A, W_S)\}$.

Step D4. $S \rightarrow A$: key distribution granted/denied.

In this protocol, two random numbers $r_A, r_S \in [1, q-1]$ are chosen by the client and the server, respectively. A computes the final session key as

$$\mathrm{SK} = (r_A \cdot \mathrm{pw}_A) \cdot W_S = r_A \cdot r_S \cdot \mathrm{pw}_A \cdot d_S \cdot G, \tag{9}$$

and S computes

$$\mathrm{SK} = (r_S \cdot d_S) \cdot W_A = r_A \cdot r_S \cdot \mathrm{pw}_A \cdot d_S \cdot G. \tag{10}$$

3. Analysis

In this section, we demonstrate that Hafizul Islam and Biswas' [10] scheme is vulnerable to password compromise impersonation attack. In addition, the comments on the scheme show the security weaknesses caused by the low-entropy password, weak password-verifier table, and improper challenge-response mechanism.

3.1. Password Compromise Impersonation Attack. The password as the unique secret information of the client plays the key role in the password-based remote authentication schemes. Intuitively, the adversary could impersonate the client, who compromises his/her password, to cheat the remote server as the trivial attack. However, the password compromise impersonation [14–16] as a special attack indicates that the adversary could impersonate the remote server to cheat the client himself/herself using his/her compromised password.

PCI attack is defined as,in the password-based client-server remote authentication (or authenticated key distribution) scheme, the adversary is considered successful in a PCI attack if it can impersonate the uncorrupted remote server S to communicate with the corrupted client A, who compromised his/her password to the adversary. In other words, the goal of the adversary by launching PCI attack is to impersonate the remote server to cheat the client himself/herself without being detected. More detailed introductions about PCI attack could be found in the literatures [14–16].

PCI Attack. Assume that the adversary not only can control the communication between the client and the server, that is, it can eavesdrop, record, intercept, modify, delete, insert messages, or even inject new messages during the protocol execution, but also can obtain the password pw_A of client A. Then PCI attack can be performed as the following steps and referred to as the illustration in Figure 1.

Step 1. The adversary intercepts the login request message,

$$\left\{\mathrm{ID}_A, E_{K_x}\left(\mathrm{ID}_A, R_A, W_A\right)\right\}, \tag{11}$$

sent from A to S, when A initializes a new password authentication session with S in Step A1.

Step 2. The adversary computes

$$K = \mathrm{pw}_A \cdot U_S = \mathrm{pw}_A \cdot d_S \cdot G = \left(K_x, K_y\right) \tag{12}$$

and decrypts

$$E_{K_x}\left(\mathrm{ID}_A, R_A, W_A\right), \tag{13}$$

with K_x to obtain W_A. Then the adversary generates a random number r_S^* and computes

$$W_A + W_S^*, H\left(W_S^*\right), \tag{14}$$

where $W_S^* = r_S^* \cdot U_S$. Finally, the adversary sends the reply,

$$\left\{W_A + W_S^*, H\left(W_S^*\right)\right\}, \tag{15}$$

to A. Note that the verification procedures executed by the adversary could be ignored for simplicity, due to the purpose of impersonating the remote server.

Step 3. After receiving the reply from the adversary, A retrieves W_S^* from $W_A + W_S^*$, verifies the hash value of retrieved W_S^* with received $H(W_S^*)$, and sends

$$H\left(W_A, W_S^*\right) \tag{16}$$

to the adversary.

Step 4. According to the description of the original protocol, the adversary computes $H(W_A, W_S^*)$ with its own copies of W_A and W_S^* and compares the results with the received $H(W_A, W_S^*)$. If they are equal, the adversary accepts the client's login request, otherwise rejects.

The password change and session key distribution phases are vulnerable to PCI attack with the same procedures for different targets. First, the adversary could get the new password verifier by retrieving U_A' from $W_A + U_A'$ using the decrypted W_A in $E_{K_x}(\mathrm{ID}_A, R_A, W_A)$ caused by the compromised password pw_A. Then the adversary could further launch offline password guessing attack to obtain the new password pw_A' of the client. Secondly, the adversary can compute and share the session key

$$\mathrm{SK}^* = \mathrm{pw}_A \cdot r_S^* \cdot R_A = r_A \cdot r_S^* \cdot \mathrm{pw}_A \cdot d_S \cdot G, \tag{17}$$

where $\mathrm{SK}^* = \mathrm{pw}_A \cdot r_A \cdot W_S^* = r_A \cdot r_S^* \cdot \mathrm{pw}_A \cdot d_S \cdot G$ is computed by A. Consequently, the adversary could also launch man-in-the-middle attack and modify the communications between A and S arbitrarily.

3.2. Comments. The first and most important weakness in Hafizul Islam and Biswas' [10] scheme is the low-entropy password, which is usually vulnerable to guessing (online or offline) attacks. The reason for guessing attack is that the password is selected in a small space/set, which is called a dictionary D with the size of $|D|$, and therefore the password can be easy-to-remember. However, the small space of the dictionary is a double-edged swords; it provides the convenience for users and could be used by the adversary to guess the correct password through analyzing the security flaws in the algorithms. He [11], Wang et al. [12], and Li [13] have demonstrated that the adversary could launch various offline password guessing attacks, for example, tracing the password in the execution of the scheme to match the redundant information, using the verifier tables to confirm the guessed password, and obtaining the verifier table to guess the client's password by the malicious system manager or the privileged insider. Furthermore, once the password of the client is compromised, the adversary not only can impersonate the client to cheat the remote server, but also can impersonate the remote server to cheat the client himself/herself. Finally, the serious security weaknesses caused by the unique low-entropy factor (password) show that the single factor cannot resist common attacks sufficiently and the second factor (smart card) should be introduced to overcome the security flaws while keeping the improved scheme efficient and practical.

Moreover, the threats on the weak password-verifier tables have shown in [11, 12], for example, offline password guessing attack and privileged insider attacks. The weak password-verifier tables have been the crucial targets for most adversaries, who can take these tables for further attack. Generally speaking, offline password guessing attack is always depending on the verifier tables, which provide the matching information. Moreover, various application servers could take the password-verifier tables carelessly, because the secret key d_S is their crucial information for themselves, but password-verifier tables are not. In addition, the password-verifier tables are the same with the others usually, and the leakage of the password-verifier tables occasionally happens in real applications. Consequently, the weak password-verifier tables should be avoided in the future design.

The challenge-response mechanism should be used for resisting replay attack and contribute to the fresh session key. However, the improper challenge-response mechanism may

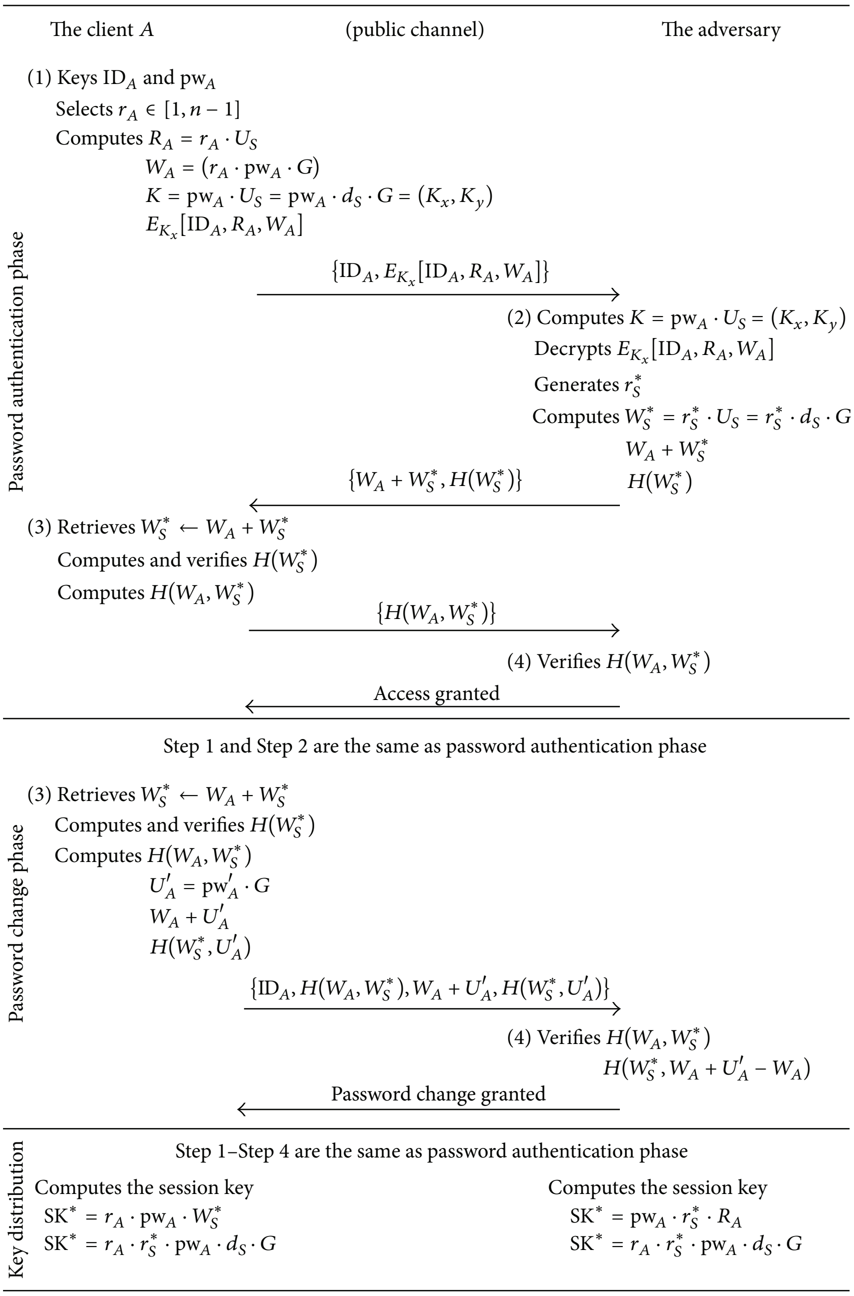

The client A
(public channel)
The adversary
Password authentication phase
(1) Keys ID_A and pw_A
Selects $r_A \in [1, n-1]$
Computes $R_A = r_A \cdot U_S$
$W_A = (r_A \cdot \mathrm{pw}_A \cdot G)$
$K = \mathrm{pw}_A \cdot U_S = \mathrm{pw}_A \cdot d_S \cdot G = (K_x, K_y)$
$E_{K_x}[\mathrm{ID}_A, R_A, W_A]$
$\{\mathrm{ID}_A, E_{K_x}[\mathrm{ID}_A, R_A, W_A]\}$
(2) Computes $K = \mathrm{pw}_A \cdot U_S = (K_x, K_y)$
Decrypts $E_{K_x}[\mathrm{ID}_A, R_A, W_A]$
Generates r_S^*
Computes $W_S^* = r_S^* \cdot U_S = r_S^* \cdot d_S \cdot G$
$W_A + W_S^*$
$H(W_S^*)$
$\{W_A + W_S^*, H(W_S^*)\}$
(3) Retrieves $W_S^* \leftarrow W_A + W_S^*$
Computes and verifies $H(W_S^*)$
Computes $H(W_A, W_S^*)$
$\{H(W_A, W_S^*)\}$
(4) Verifies $H(W_A, W_S^*)$
Access granted
Step 1 and Step 2 are the same as password authentication phase
Password change phase
(3) Retrieves $W_S^* \leftarrow W_A + W_S^*$
Computes and verifies $H(W_S^*)$
Computes $H(W_A, W_S^*)$
$U'_A = \mathrm{pw}'_A \cdot G$
$W_A + U'_A$
$H(W_S^*, U'_A)$
$\{\mathrm{ID}_A, H(W_A, W_S^*), W_A + U'_A, H(W_S^*, U'_A)\}$
(4) Verifies $H(W_A, W_S^*)$
$H(W_S^*, W_A + U'_A - W_A)$
Password change granted
Key distribution
Step 1–Step 4 are the same as password authentication phase
Computes the session key
$\mathrm{SK}^* = r_A \cdot \mathrm{pw}_A \cdot W_S^*$
$\mathrm{SK}^* = r_A \cdot r_S^* \cdot \mathrm{pw}_A \cdot d_S \cdot G$
Computes the session key
$\mathrm{SK}^* = \mathrm{pw}_A \cdot r_S^* \cdot R_A$
$\mathrm{SK}^* = r_A \cdot r_S^* \cdot \mathrm{pw}_A \cdot d_S \cdot G$

FIGURE 1: PCI attack on Hafizul Islam and Biswas's scheme.

be used by the adversary to launch DoS attack. In addition, the denial of service attack pointed by Wang et al. [12] is caused by the improper challenge-response mechanism, because the adversary could replay all the expired legal login request messages and delegate the resources of the server, for example, computation, memory, and connection. Another reason for the denial of service attack is the expensive cost of the bilinear paring operations. Thus, the improper challenge-response mechanism may cause important security issues or break down the system. Consequently, how to take the maximum advantage of challenge-response mechanism into the scheme is quite helpful for future design.

4. Enhancement

There are two participants in the protocol: the user as the client A and the remote server S. The proposed scheme is composed of five phases, namely, registration phase, authentication with key agreement phase, password change phase, revocation/reregistration phase, and key update phase.

The details of the enhanced scheme are described as follows and illustrated in Figure 2.

4.1. Registration Phase. When the client A wants to register in the remote server S as a legal client to obtain the services, the following steps should be performed.

Step R1. The client A chooses the identity ID_A with the password pw_A, generates a random number b, and sends the registration request,

$$\{\mathrm{ID}_A, H(b \parallel \mathrm{pw}_A)\}, \tag{18}$$

to S over the secure channel.

Step R2. S checks the validity of ID_A after receiving the registration request and computes the client's authentication information

$$\mathrm{AI}_A = H(\mathrm{ID}_A \parallel M_A) \oplus H(\mathrm{ID}_A \parallel H(b \parallel \mathrm{pw}_A)), \tag{19}$$

where $M_A = x \oplus N_A$, x is the secret key of S and N_A is the unique identifier (or random number) generated by S for the smart card. Then the smart card is initialized by the parameters

$$[\mathrm{AI}_A, N_A, G, H(\cdot)], \tag{20}$$

where G is the generator of the elliptic curve cryptosystem. Next, S sends the smart card to A over the secure channel and maintains the client table as

$$[\mathrm{ID}_A, N_A, \mathrm{Status}, \mathrm{Update}], \tag{21}$$

where Status $\in \{0,1\}$ indicates the log-in 1 or log-off 0 status and Update $\in \{0,1\}$ indicates if the client updates the latest authentication information AI_A.

Step R3. The client A initializes the smart card with the parameters b, V_A, where $V_A = H(\mathrm{ID}_A \oplus H(b \oplus \mathrm{pw}_A))$. All the parameters in the smart card are

$$[b, V_A, N_A, \mathrm{AI}_A, G, H(\cdot)], \tag{22}$$

and ID_A with pw_A are kept by the client as his/her own knowledge. Finally, the registration phase is finished and A shares the secret,

$$\mathrm{SAI}_A = H(\mathrm{ID}_A \parallel M_A), \tag{23}$$

with S to authenticate each other and establish the session key.

4.2. Authentication with Key Agreement Phase. When A wants to access the remote server and obtains the desired services, the following operations should be executed.

Step A1. The client A inputs ID_A^* with pw_A^* into his/her smart card. The smart card computes

$$V_A^* = H(\mathrm{ID}_A^* \oplus H(b \oplus \mathrm{pw}_A^*)) \tag{24}$$

and checks

$$V_A^* = ?V_A. \tag{25}$$

If the equation holds, the smart card confirms the legal holder and sends the login request

$$\{\mathrm{Hello}\} \tag{26}$$

to S. Note that once the smart card confirms its legal holder, that is, the equations $\mathrm{ID}_A^* = \mathrm{ID}_A$ and $\mathrm{pw}_A^* = \mathrm{pw}_A$ are true.

Step A2. After receiving the login request, S sends the precomputed challenge,

$$R_S = r_S \cdot G, \tag{27}$$

to A, where r_S is a random number generated by S. Note that the challenge could be seen as a client puzzle [17] and sent by the technology of completely automated public turing test to tell computers and humans apart (CAPTCHA) [18].

Step A3. The client A solves and inputs the challenge R_S^*, and the smart card generates its own challenge

$$R_A = r_A \cdot G, \tag{28}$$

computing

$$\mathrm{TK} = H(r_A \cdot R_S^*) = H(r_A \cdot r_S \cdot G),$$
$$\mathrm{SAI}_A = \mathrm{AI}_A \oplus H(\mathrm{ID}_A \parallel H(b \parallel \mathrm{pw_A})) = H(\mathrm{ID}_A \parallel M_A), \tag{29}$$

where r_A is a random number generated by the smart card. Then the smart card sends the response and its challenge,

$$\{R_S^*, R_A, E_{\mathrm{TK}}[\mathrm{ID}_A, H(R_A \| R_S^* \| \mathrm{SAI}_A)]\}, \tag{30}$$

to S.

Step A4. After confirming the validity of the response R_S^*, S computes

$$\mathrm{TK} = H(r_S \cdot R_A) = H(r_S \cdot r_A \cdot G) \tag{31}$$

and decrypts

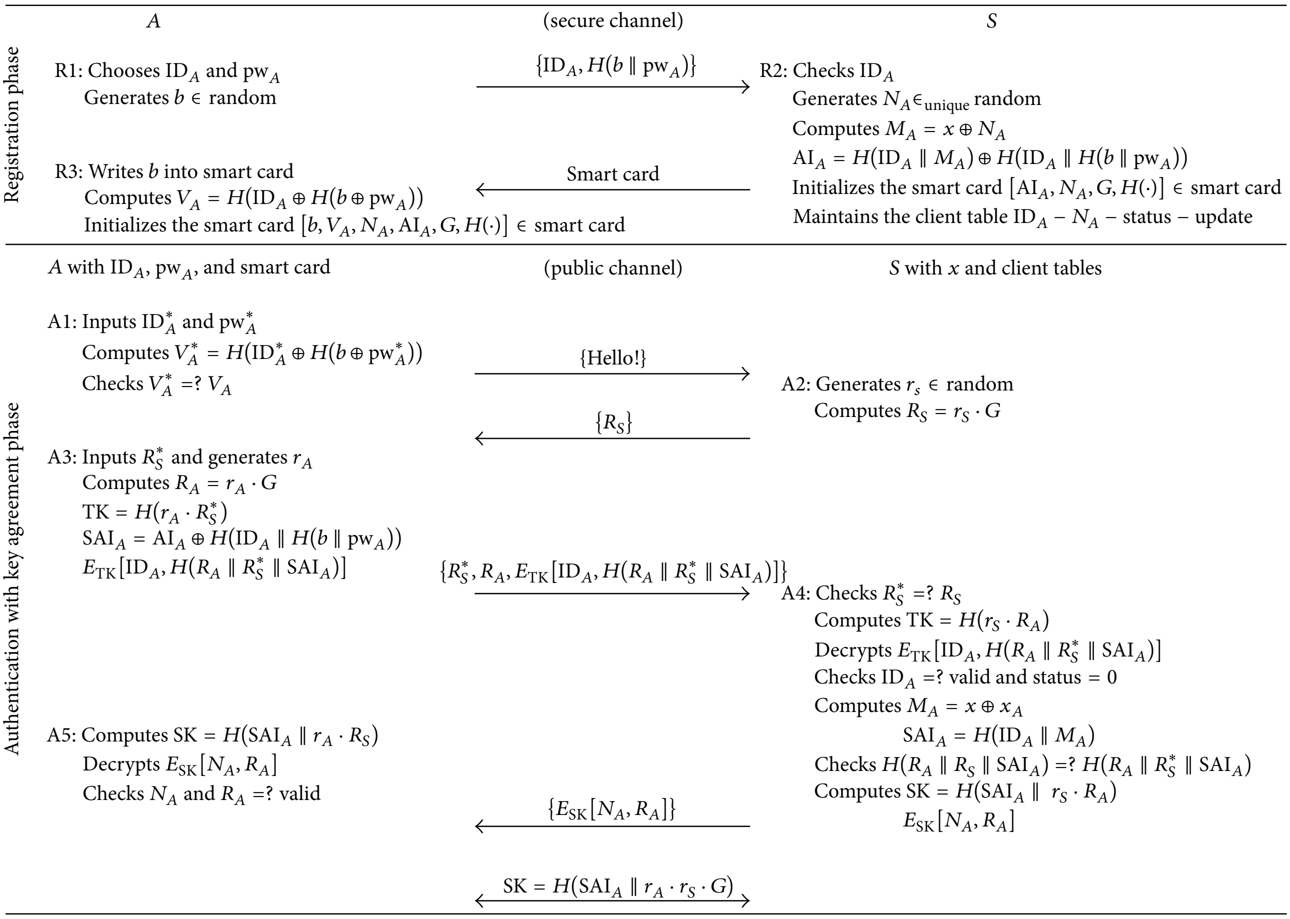
$$E_{\mathrm{TK}}[\mathrm{ID}_A, H(R_A \| R_S^* \| \mathrm{SAI}_A)] \tag{32}$$

to get ID_A. If S finds ID_A in the client tables, then checks the Status of A. If A has logged-in (Status $=$ 1), S terminates the session. Otherwise, S extracts N_A in the client table and computes

$$\mathrm{SAI}_A = H(\mathrm{ID}_A \parallel M_A), \tag{33}$$

where $M_A = x \oplus N_A$. After that, S checks whether the computed value

$$H(R_A \parallel R_S \parallel \mathrm{SAI}_A) \tag{34}$$


Registration phase
A
(secure channel)
S
$\{\mathrm{ID}_A, H(b \parallel \mathrm{pw}_A)\}$
Smart card
Authentication with key agreement phase
A with ID_A, pw_A, and smart card
(public channel)
S with x and client tables
$\{\text{Hello!}\}$
$\{R_S\}$
$\{R_S^*, R_A, E_{\mathrm{TK}}[\mathrm{ID}_A, H(R_A \parallel R_S^* \parallel \mathrm{SAI}_A)]\}$
$\{E_{\mathrm{SK}}[N_A, R_A]\}$
$\mathrm{SK} = H(\mathrm{SAI}_A \parallel r_A \cdot r_S \cdot G)$

R1: Chooses ID_A and pw_A
Generates $b \in$ random

R2: Checks ID_A
Generates $N_A \in_{\text{unique}}$ random
Computes $M_A = x \oplus N_A$
$\mathrm{AI}_A = H(\mathrm{ID}_A \parallel M_A) \oplus H(\mathrm{ID}_A \parallel H(b \parallel \mathrm{pw}_A))$
Initializes the smart card $[\mathrm{AI}_A, N_A, G, H(\cdot)] \in$ smart card
Maintains the client table $\mathrm{ID}_A - N_A -$ status $-$ update

R3: Writes b into smart card
Computes $V_A = H(\mathrm{ID}_A \oplus H(b \oplus \mathrm{pw}_A))$
Initializes the smart card $[b, V_A, N_A, \mathrm{AI}_A, G, H(\cdot)] \in$ smart card

A1: Inputs ID_A^* and pw_A^*
Computes $V_A^* = H(\mathrm{ID}_A^* \oplus H(b \oplus \mathrm{pw}_A^*))$
Checks $V_A^* =?\ V_A$

A2: Generates $r_s \in$ random
Computes $R_S = r_S \cdot G$

A3: Inputs R_S^* and generates r_A
Computes $R_A = r_A \cdot G$
$\mathrm{TK} = H(r_A \cdot R_S^*)$
$\mathrm{SAI}_A = \mathrm{AI}_A \oplus H(\mathrm{ID}_A \parallel H(b \parallel \mathrm{pw}_A))$
$E_{\mathrm{TK}}[\mathrm{ID}_A, H(R_A \parallel R_S^* \parallel \mathrm{SAI}_A)]$

A4: Checks $R_S^* =?\ R_S$
Computes $\mathrm{TK} = H(r_S \cdot R_A)$
Decrypts $E_{\mathrm{TK}}[\mathrm{ID}_A, H(R_A \parallel R_S^* \parallel \mathrm{SAI}_A)]$
Checks $\mathrm{ID}_A =?$ valid and status $= 0$
Computes $M_A = x \oplus x_A$
$\mathrm{SAI}_A = H(\mathrm{ID}_A \parallel M_A)$
Checks $H(R_A \parallel R_S \parallel \mathrm{SAI}_A) =?\ H(R_A \parallel R_S^* \parallel \mathrm{SAI}_A)$
Computes $\mathrm{SK} = H(\mathrm{SAI}_A \parallel\ r_S \cdot R_A)$
$E_{\mathrm{SK}}[N_A, R_A]$

A5: Computes $\mathrm{SK} = H(\mathrm{SAI}_A \parallel r_A \cdot R_S)$
Decrypts $E_{\mathrm{SK}}[N_A, R_A]$
Checks N_A and $R_A =?$ valid

Figure 2: The enhanced scheme.

is equal to decrypted value

$$H\left(R_A \| R_S^* \| \mathrm{SAI}_A\right). \tag{35}$$

If it is, S authenticates A and computes the session key

$$\mathrm{SK} = H\left(\mathrm{SAI}_A \parallel r_S \cdot R_A\right) = H\left(\mathrm{SAI}_A \parallel r_S \cdot r_A \cdot G\right). \tag{36}$$

Then S computes the response

$$E_{\mathrm{SK}}\left[N_A, R_A\right] \tag{37}$$

and sends it to A. In addition, S sets up Status $= 1$ before replying the acceptance.

Step A5. The smart card computes the session key

$$\mathrm{SK} = H\left(\mathrm{SAI}_{\mathrm{A}} \parallel r_A \cdot R_S\right) = H\left(\mathrm{SAI}_{\mathrm{A}} \parallel r_A \cdot r_S \cdot G\right). \tag{38}$$

After receiving the response, the smart card decrypts

$$E_{\mathrm{SK}}\left[N_A, R_A\right] \tag{39}$$

and checks the validity of both N_A and R_A. If they are valid, A authenticates S and establishes the session key SK. Finally, mutual authentication and key agreement phase is finished successfully.

4.3. Password Change Phase. When the client wants to change the old password pw_A to a new one $\mathrm{pw}_A^{\mathrm{new}}$, the following offline steps should be performed after the smart card confirms its legal holder in Step A1.

Step P1. Once the procedure

$$V_A^* = V_A \tag{40}$$

is successfully verified, A selects the password change option and inputs the new password $\mathrm{pw}_A^{\mathrm{new}}$.

Step P2. The smart card computes

$$\begin{aligned}
V_A^{\mathrm{new}} &= V_A \oplus H\left(\mathrm{ID}_A \oplus H\left(b \oplus \mathrm{pw}_A\right)\right) \oplus H \\
&\quad \times \left(\mathrm{ID}_A \oplus H\left(b \oplus \mathrm{pw}_A^{\mathrm{new}}\right)\right) \\
&= H\left(\mathrm{ID}_A \parallel M_A\right) \oplus H\left(\mathrm{ID}_A \oplus H\left(b \oplus \mathrm{pw}_A^{\mathrm{new}}\right)\right), \\
\mathrm{AI}_A^{\mathrm{new}} &= \mathrm{AI}_A \oplus H\left(\mathrm{ID}_A \parallel H\left(b \parallel \mathrm{pw}_A\right)\right) \oplus H \\
&\quad \times \left(\mathrm{ID}_A \parallel H\left(b \parallel \mathrm{pw}_A^{\mathrm{new}}\right)\right) \\
&= H\left(\mathrm{ID}_A \parallel M_A\right) \oplus H\left(\mathrm{ID}_A \parallel H\left(b \parallel \mathrm{pw}_A^{\mathrm{new}}\right)\right).
\end{aligned} \tag{41}$$

Finally, S replaces V_A, AI_A by $V_A^{\mathrm{new}}, \mathrm{AI}_A^{\mathrm{new}}$, and password change phase is finished.

4.4. Revocation/Reregistration Phase. When A wants to revoke the his/her registration for security concern or reregister without changing his/her identity ID_A, S should delete the random number N_A for revocation or chooses a new random number N_A^{new} and executes the registration phase again for reregistration. After revocation phase, S could not authenticate A or reply the correct response to A without N_A. Similarly, The reregistration phase could make the old smart card expired, because $N_A^{\mathrm{new}} \neq N_A$. Consequently, revocation/reregistration phase is successfully finished.

4.5. System Update Phase. When the remote server requires updating the system or changing its secret key regularly, key update phase should be performed between S and A. S selects new key x^{new} and establishes a new table containing

$$[\mathrm{ID}_A, N_A^{\mathrm{new}}, \mathrm{Status}, \mathrm{Update}], \tag{42}$$

where $N_A^{\mathrm{new}} = x^{\mathrm{new}} \oplus (x \oplus N_A)$. If S updates the secret key, then it initializes all the clients' Update $= 0$ that is, all the clients should update their authentication information

$$\mathrm{SAI}_A^{\mathrm{new}} = H\left(\mathrm{ID}_A \parallel M_A^{\mathrm{new}}\right). \tag{43}$$

Note that the client could update their secret authentication information over a secure channel established by the session key SK. In other words, S must maintain the original secret key and client tables for these specific users, who have not update their authentication information. Upon receiving $\mathrm{SAI}_A^{\mathrm{new}}$, A stores

$$\mathrm{AI}_A^{\mathrm{new}} = \mathrm{SAI}_A^{\mathrm{new}} \oplus H\left(\mathrm{ID}_A \parallel H\left(b \parallel \mathrm{pw}_A\right)\right), \tag{44}$$

replacing AI_A and S deletes the old list in the original tables of A while marking Update $= 1$. Finally, the system update phase is finished successfully.

5. Analysis and Comments

In this section, the security analysis demonstrates that the improved scheme not only remedies the weaknesses mentioned above, but also can resist all known common attacks. Furthermore, the comparisons of the security attribute, performance cost, and functionality illustrate that the improved scheme is more secure, efficient, and practical than the scheme in [10].

5.1. Security Analysis. The security of the scheme is based on the secure cryptographic primitives, including one-way hash function, pseudorandom generator, and symmetric cryptosystem. Furthermore, the assumptions of discrete logarithm problem (DLP) and computational Diffie-Hellman and decisional Diffie-Hellman problems (CDHP and DDHP) on the elliptic curve are hard to be solved under the polynomial time algorithms [19, 20].

5.1.1. Impersonation Attack. The enhanced scheme can resist the following common attacks for the purpose of impersonation, including replay attack, reflection attack, parallel session attack, man-in-the-middle attack, known session key attack, forgery attack, and password compromise impersonation attack.

(1) The technologies of client puzzle R_S and challenge-response mechanism R_A are introduced into resist replay attack, reflection attack, and parallel session attack. r_S and r_A can also contribute to the computation of the fresh session key $\mathrm{SK} = H(\mathrm{SAI}_A \parallel r_A \cdot R_S) = H(\mathrm{SAI}_A \parallel r_S \cdot R_A)$, which can resist known session key attack.

(2) The design of mutual authentication with key agreement can help to resist man-in-the-middle attack in our scheme; that is, the key agreement protocol is authenticated and the adversary could not launch man-in-the-middle attack without authentication. In other words, authenticated Diffie-Hellman mechanism helps to resist man-in-the-middle attack.

(3) Any adversary could not impersonate the legal participants (client or remote server) to share the session key with the intended partner, because the adversary cannot forge the messages $E_{\mathrm{TK}}[\mathrm{ID}_A, H(R_A \| R_S^* \| \mathrm{SAI}_A)]$ or $E_{\mathrm{SK}}[N_A, R_A]$ without knowing the temporary key TK or the session key *SK*. The security of the temporary key TK is based on the assumption of DLP and CDHP. If the adversary could get TK, that is, the adversary can compute $\mathrm{TK} = H(r_A \cdot r_S \cdot G)$ without r_A or r_S, which is infeasible under the assumptions. It is the same for the session key as that the adversary cannot compute SK without solving DLP or CDHP. Furthermore, the secret authentication information SAI_A can also help to resist impersonation attack. SAI_A is important for the adversary to forge the messages for authentication, because $H(R_A \| R_S^* \| \mathrm{SAI}_A)$ and $\mathrm{SK} = H(\mathrm{SAI}_A \parallel r_A \cdot r_S \cdot G)$ are composed of SAI_A. However, $\mathrm{SAI}_A = H(\mathrm{ID}_A \parallel M_A)$ can be computed only by the legal client with the corrected $\mathrm{ID}_A, \mathrm{pw}_A$, and the smart card or by the remote server with x and N_A.

(4) The two-factor authentication with key agreement can resist the password compromise impersonation attack in the enhanced scheme. If the client's password pw_A is compromised, the adversary cannot forge the correct authentication message without knowing ID_A and obtaining the smart card. Furthermore, the secret information cannot be computed by the adversary with pw_A only, because the security of SAI_A depends on $\mathrm{ID}_A, \mathrm{pw}_A, \mathrm{AI}_A$ for user or x, N_A for server.

5.1.2. Password Guessing Attack. In password-based schemes, the adversary can guess the password in a dictionary D, which is defined in a finite space of size $|D|$. The adversary can guess the correct password with the successful probability $1/|D|$. However, the enhanced scheme with two factors can resist such attack due to the first defense of smart card, which can help to protect the information stored in its memory. Furthermore, the anonymity in the enhanced scheme can also resist password guessing attack with higher level, because the adversary must guess ID_A and pw_A at the same time. In other words, the success of the probability about guessing the correct password is Minmum$\{1/|D|^{|D'|}, 1/|D'|^{|D|}\}$, where $|D'|$ is the size of the identity dictionary. In addition, online password guessing attack is out of our consideration, because the technologies of client puzzle and CAPTCHA and additional network equipment (e.g., IDS and firewall) can help the remote server to restrict the limitations of failed login attempts.

5.1.3. Secrecy of the Session Key. The secrecy of the fresh session key $\mathrm{SK} = H(\mathrm{SAI}_A \parallel r_A \cdot r_S \cdot G)$ includes key privacy, forward secrecy, and key control. First, the challenge-response mechanism R_A and R_S can help to contribute the fresh of the session key and make the generation of the session key out of control. Secondly, the secure authentication information SAI_A, which can be computed by A and S, decides that any one cannot break the key privacy without knowing SAI_A. Furthermore, under the assumptions of DLP, CDHP, and DDHP, the forward secrecy of the session key can be protected even if the long term keys SAI_A or x is compromised. Finally, the authenticated Diffie-Hellman key exchange enhances the security of the scheme, because the compromise of the temporary random number cannot threat the security of the final session key $\mathrm{SK} = H(\mathrm{SAI}_A \parallel r_A \cdot r_S \cdot G)$ without knowing SAI_A.

5.1.4. Credentials Leakage Resistant. The credentials mentioned in the enhancement are ID_A, pw_A, SAI_A, x, the smart card, and client tables. Credentials leakage means the adversary could get some of the credentials. In detail, the anonymous login request protects ID_A from leakage and meanwhile protects pw_A from guessing attack. Specifically, if the adversary could forge a server by phishing user's identity ID_A, user anonymity cannot be preserved as usual. An additional mechanism should be provided to avoid this attack, while the other credentials are still protected as normal. Furthermore, secure one-way hash function helps to avoid the compromise of SAI_A from $H(R_A \| R_S^* \| \mathrm{SAI}_A)$, $H(\mathrm{SAI}_A \parallel r_A \cdot r_S \cdot G)$ and protect x from being extracted in $H(\mathrm{ID}_A \parallel M_A)$ by the insider clients.

5.1.5. Denial of Service Resistance. The technologies of client puzzle and CAPTCHA are introduced to protect the system from being DoS attacks. In addition, the other network equipment (e.g., IDS and firewall) can be used in the system to avoid such attacks.

5.2. Comparisons and Comments. The comparisons and comments with related works [6, 10, 13] on security and functionality are shown to illustrate that our enhancement is more secure and robust. The comparisons of security features in Table 1 show that our enhancement satisfies more security features, including password guessing attack resistance (PGAR), verifier table attack resistance (VTAR), password compromise impersonation attack resistance (PCIAR), forward secrecy resistance (FSR), denial of service attack resistance (DoSAR), and known temporary information attack resistance (KTIAR). Moreover, the comparisons of functionalities in Table 2 show that our enhancement provides more functionalities mentioned in Section 1 to support user friendly property and system flexibility. In addition, our enhancement can be implemented in the environments of symmetric cryptosystem; that is, it is more practical without public key infrastructure (PKI). Finally, our enhancement of two-factor authentication with key agreement scheme using smart card is suitable for mobile wireless communication system while keeping low efficiency on elliptic curve cryptosystem without expensive computations, for example, modular exponentiation or bilinear pairings.

Table 1: Comparisons of security features.

	PGAR	VTAR	PCIAR	FSR	DoSAR	KTIAR
[6]	No	No	Yes	No	No	No
[10]	No	No	No	Yes	No	Yes
[13]	No	Yes	Yes	Yes	No	Yes
Ours	Yes	Yes	Yes	Yes	Yes	Yes

For computational comparison, we only consider the latest schemes, for example, [10, 13], and our proposal. Table 3 shows the computation cost in the login and authentication phase, which is the main procedure of the scheme. It illustrates that our proposal costs 3 (4) more hash function and one more symmetric decryption (encryption) operation for user (server), but we save more time cost operations, such as point-multiplication operation on elliptic curve, point-multiplication operation on finite field, addition operation, and bilinear paring computation on elliptic curve.

6. Conclusion

In this paper, the scheme of Hafizul Islam and Biswas is cryptanalyzed and improved. Password compromise impersonation attack is demonstrated and some security weaknesses are discussed about their scheme. Furthermore, an enhanced scheme in symmetric key environment is presented to overcome the existing weaknesses and provide more functionalities. In detail, the technologies of client puzzle and CAPTCHA are introduced to resist the common known attacks with proper challenge-response mechanism. The public key infrastructure is replaced by the second factor (smart card) to enhance the security and robustness of the scheme. In addition, the enhanced scheme can also be used in global mobility networks to provide secure authentication and private communication. Finally, the analysis and comments show that our improved scheme is more secure, practical, efficient, and suitable for smart card while providing more user friendly property and system flexibility.

Table 2: Comparisons of functionalities.

	RG1	RG2	RG3	RG4	RG5	RG6	RG7	RG8	PKI
[6]	No	Yes	No	No	No	Yes	No	No	Yes
[10]	Yes	Yes	No	No	No	No	Yes	No	Yes
[13]	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes
Ours	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No

Table 3: Comparisons of computation cost.

Proposal	Participant	Computation cost
[10]	User	$3T_{\mathrm{ME}} + 1T_{\mathrm{MF}} + 1T_{\mathrm{ENC}} + 2T_H + 1T_{\mathrm{AE}}$
	Server	$2T_{\mathrm{ME}} + 1T_{\mathrm{MF}} + 1T_{\mathrm{DEC}} + 1T_H + 2T_P$
[13]	User	$4T_{\mathrm{ME}} + 2T_{\mathrm{MF}} + 1T_{\mathrm{ENC}} + 2T_H + 1T_{\mathrm{AE}}$
	Server	$2T_{\mathrm{ME}} + 1T_{\mathrm{MF}} + 1T_{\mathrm{DEC}} + 2T_H + 2T_P$
Ours	User	$2T_{\mathrm{ME}} + 0T_{\mathrm{MF}} + 1T_{\mathrm{ENC}} + 6T_H + 0T_{\mathrm{AE}} + 1T_{\mathrm{DEC}}$
	Server	$2T_{\mathrm{ME}} + 0T_{\mathrm{MF}} + 1T_{\mathrm{DEC}} + 5T_H + 0T_P + 1T_{\mathrm{ENC}}$

T_{ME}: point-multiplication operation on elliptic curve.
T_{MF}: point-multiplication operation on finite field.
T_{ENC}: symmetric encryption operation.
T_{DEC}: symmetric decryption operation.
T_H: hash operation.
T_{AE}: addition operation on elliptic curve.
T_P: bilinear paring operation on elliptic curve.

Notations

A:	The client
S:	The server
ID_A:	Identity of the client A
pw_A:	Secret password of the client A
G:	Base point of the elliptic curve group
d_S:	Secret key of the server S
U_S:	Public key of the server S
U_A:	Password verifier of the client A
q:	A large prime number
$E_q(a, b)$:	Nonsingular elliptic curve over a finite field
$H(\cdot)$:	Collision-resistant one-way secure hash function
E_K/D_K:	Symmetric encryption/decryption algorithm with key K
$\hat{e}$:	Bilinear pairings mapping
SK:	Symmetric session key.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

References

[1] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[2] C. C. Lee, C. H. Liu, and M. S. Hwang, "Guessing attacks on strong-password authentication protocol," *International Journal of Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[3] M. Peyravian and N. Zunic, "Methods for protecting password transmission," *Computers & Security*, vol. 19, no. 5, pp. 466–469, 2000.

[4] I.-E. Liao, C.-C. Lee, and M.-S. Hwang, "A password authentication scheme over insecure networks," *Journal of Computer and System Sciences*, vol. 72, no. 4, pp. 727–740, 2006.

[5] J.-J. Hwang and T.-C. Yeh, "Improvement on Peyravian-Zunic's password authentication schemes," *IEICE Transactions on Communications*, vol. 85, no. 4, pp. 823–825, 2002.

[6] C.-L. Lin and T. Hwang, "A password authentication scheme with secure password updating," *Computers & Security*, vol. 22, no. 1, pp. 68–72, 2003.

[7] L. Yang, J.-F. Ma, and Q. Jiang, "Mutual authentication scheme with smart cards and password under trusted computing," *International Journal of Network Security*, vol. 14, no. 3, pp. 156–163, 2012.

[8] M. Kumar, M. K. Gupta, and S. Kumari, "An improved efficient remote password authentication scheme with smart card over insecure networks," *International Journal of Network Security*, vol. 13, no. 3, pp. 167–177, 2011.

[9] R. Ramasamy and A. P. Muniyandi, "An efficient password authentication scheme for smart card," *International Journal of Network Security*, vol. 14, no. 3, pp. 180–186, 2012.

[10] S. Hafizul Islam and G. P. Biswas, "Design of improved password authentication and update scheme based on elliptic curve cryptography," *Mathematical and Computer Modelling*, vol. 57, no. 11-12, pp. 2703–2717, 2013.

[11] D. He, "Comments on a password authentication and update scheme based on elliptic curve cryptography," Cryptology EPrint Archive Report 2011/411, 2011, https://eprint.iacr.org/2011/411.pdf.

[12] D. Wang, C. G. Ma, L. Shi, and Y. H. Wang, "On the security of an improved password authentication scheme based on ECC," in *Information Computing and Applications*, vol. 7473 of *Lecture Notes in Computer Science*, pp. 181–188, 2012.

[13] C. T. Li, "A new password authentication and user anonymity scheme based on elliptic curve cryptography and smart card," *IET Information Security*, vol. 7, no. 1, pp. 3–10, 2013.

[14] D.-G. Feng and J. Xu, "A new client-to-client password-authenticated key agreement protocol," in *Coding and Cryptology*, vol. 5557 of *Lecture Notes in Computer Science*, pp. 63–76, Springer, Berlin, Germany, 2009.

[15] W. Jin and J. Xu, "An efficient and provably secure cross-realm client-toclient password-authenticated key agreement protocol with smart cards," in *Cryptology and Network Security*, vol. 5888 of *Lecture Notes in Computer Science*, pp. 299–314, 2009.

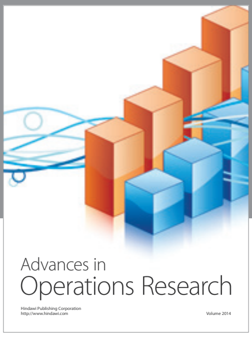
[16] M. C. Gorantla, C. Boyd, J. M. G. Nieto, and M. Manulis, "Modeling key compromise impersonation attacks on group key exchange protocols," *ACM Transactions on Information and System Security*, vol. 14, no. 4, article 28, 2011.

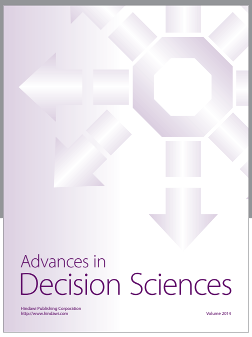
[17] B. Waters, A. Juels, J. A. Halderman, and E. W. Felten, "New client puzzle outsourcing techniques for DoS resistance," in *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS '04)*, pp. 246–256, October 2004.

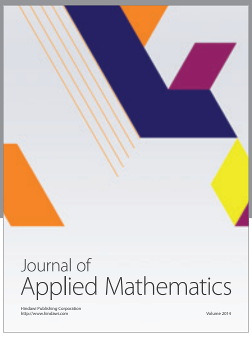
[18] L. Von Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: using hard AI problems for security," in *Advances in Cryptology—EUROCRYPT 2003*, vol. 2656 of *Lecture Notes in Computer Science*, pp. 294–311, Springer, Berlin, Germany, 2003.

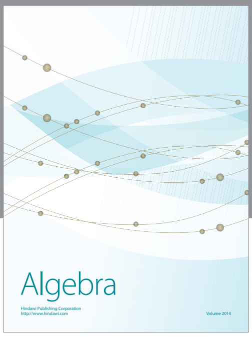
[19] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer, New York, NY, USA, 2004.

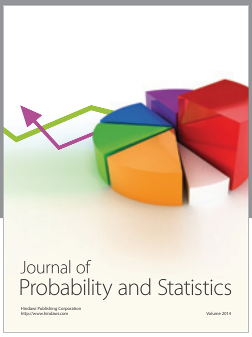
[20] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.


Advances in
Operations Research

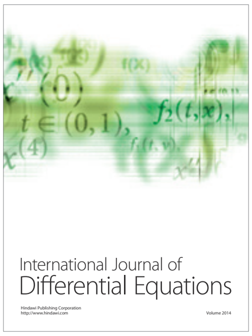

Advances in
Decision Sciences


Journal of
Applied Mathematics

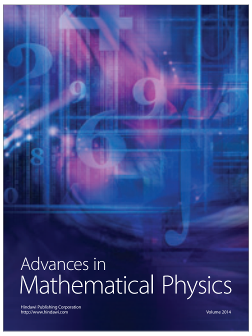

Algebra

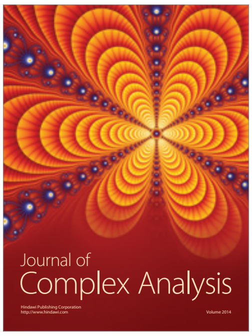

Journal of
Probability and Statistics

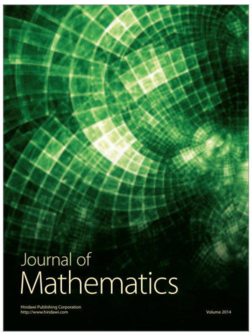
The Scientific
World Journal

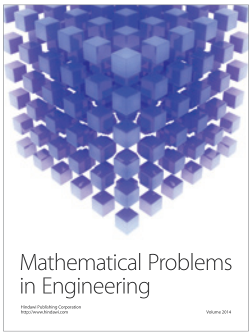

International Journal of
Differential Equations

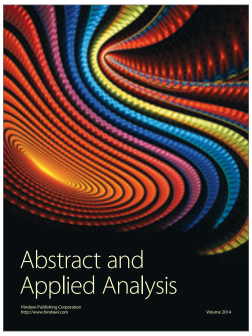
International Journal of
Combinatorics

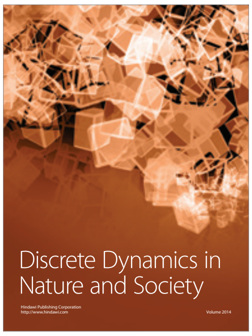

Advances in
Mathematical Physics

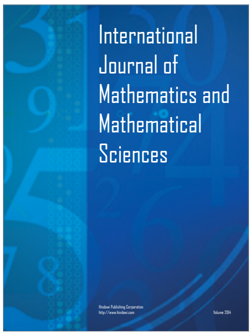

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

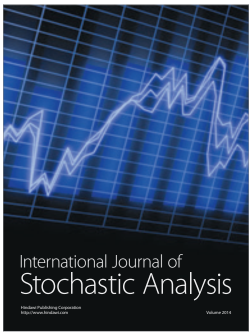

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization