Improvements of Juang *et al.*'s Password-Authenticated Key Agreement Scheme Using Smart Cards

Da-Zhi Sun, Jin-Peng Huai, Ji-Zhou Sun, Jian-Xin Li, Jia-Wan Zhang, *Member, IEEE*, and Zhi-Yong Feng

Abstract*—In the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, vol. 55, no. 6, Juang *et al.* proposed a password-authenticated key agreement scheme using smart cards. Although the scheme of Juang *et al.* has many benefits, we find that it suffers from three weaknesses: 1) inability of the password-changing operation; 2) the session-key problem; and 3) inefficiency of the double secret keys. Therefore, we propose an improved scheme to overcome the weaknesses and maintain the benefits of the original scheme. In addition, our improved scheme reduces the storage and computation costs on the smart card compared with the scheme of Juang *et al.* We believe that our improved scheme is more suitable for real-life applications than that of Juang *et al.

***Index Terms*—Authentication, key agreement, network security, password, smart card.**

NOTATION

We first define some notations used throughout this paper.

S	server;
U	user;
ID	U's identifier;
PW	U's human-memorizable password;
K_S	master secret key only kept by S;
K_{SU}	session key shared between S and U;
$h()$, $h_1()$, and $h_2()$	cryptographic hash functions;
$E_K()$	symmetric encryption algorithm using the secret key K;
$\|$	concatenation operator;
$\oplus$	bitwise exclusive-OR operator.

Manuscript received September 22, 2008; revised January 19, 2009. First published March 16, 2009; current version published June 3, 2009. This work was supported in part by the China Postdoctoral Science Foundation Funded Project under Grant 20070420288, in part by the Doctoral Program Foundation of Institutions of Higher Education of China Funded the New Teacher Project under Grant 200800561044, in part by the National Natural Science Foundation of China under Grant 60673196, and in part by the National High Technology Research and Development Plan of China under Grant 2007AA01Z130.

D.-Z. Sun is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China, and also with the School of Computer Science, Beihang University, Beijing 100191, China (e-mail: sundazhi@tju.edu.cn).

J.-P. Huai and J.-X. Li are with the School of Computer Science, Beihang University, Beijing 100191, China (e-mail: huaijp@buaa.edu.cn; lijx@act.buaa.edu.cn).

J.-Z. Sun, J.-W. Zhang, and Z.-Y. Feng are with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: jzsun@tju.edu.cn; jwzhang@tju.edu.cn; zyfeng@tju.edu.cn).

Digital Object Identifier 10.1109/TIE.2009.2016508

I. INTRODUCTION

WITH THE rapid growth of industrial network technology [1]–[6], user authentication takes an important role for achieving the dependable network environments. The goal of user authentication is to provide the communicating entities with some assurance that they know each other's true identities. There is the additional goal that the two entities end up sharing a common key called a session key known only to them. This session key can then be used for some time thereafter to provide privacy, data integrity, or both. There is a vast literature on authenticated key agreement schemes. We refer the reader to [7] for a more extensive historical discussion and to [8] and [9] for formal model approaches to design and analyze authenticated key agreement schemes.

A smart card usually consists of a microprocessor (8 bits for 47 low-cost cards and up to 32 bits in the newest generation of smart cards), memory, and some interface. Usually, the interface is a serial interface, but also universal-serial-bus interfaces or radio-frequency interfaces for contactless smart cards are possible. Smart-card commands are coded in application protocol data units. Some smart cards contain cryptographic coprocessors that can assist the main microprocessor. These coprocessors are usually used to implement standard cryptographic algorithms more efficiently or to perform more complicated arithmetic on large integers.

In order to provide a better support for user authentication, smart cards are used today in thousands of applications. Here, smart cards usually execute cryptographic computations based on secret keys embedded in their nonvolatile memories. Due to this popular usage of smart cards, much attention [10]–[15] has recently been paid regarding the security issues of a smart-card-based authentication system. In the typical scenario, a smart-card-based authentication system always involves two entities, i.e., the server and the user. At first, the server issues the smart card to the user. This smart card is personalized by the user's information. For security enhancement, the user usually selects his human-memorizable password for the smart card. Later on, to log on to the server, the user, with the help of his smart card, runs an authentication session with the server. For the subsequent secret communication, the user and the server need to establish the session key after the authentication session. Clearly, the authentication session is frequently carried out between the user and the server. In addition, the user may want to change his password for the smart card under some

situations. Therefore, the password-changing process is needed to provide it for the user.

According to the aforementioned security requirements, Juang *et al.* [16] proposed a password-authenticated key agreement scheme using smart cards. They broke new ground by pointing out the threat of the smart-card loss. As shown in [17] and [18], implementation attacks can use the side channel information, such as timing measurement, power consumption, and faulty hardware, to extract secret keys from the smart card. Therefore, Juang *et al.* assumed that, once the attacker steals the user's smart card, he could extract all secret keys from the smart card in order to impersonate the user for unauthorized authentication sessions. The major contributions of the scheme of Juang *et al.* are to address the threat of the smart-card loss and the use of the elliptic-curve algorithm for reducing the implementation costs. In fact, most of the previous schemes are insecure under the smart-card-loss assumption. Although the scheme of Juang *et al.* has many benefits, we find that it suffers from three weaknesses: 1) inability of the password-changing operation; 2) the session-key problem; and 3) inefficiency of the double secret keys. That is, it fails to fully meet the security requirements that this type of scheme should achieve. The contributions of this paper therefore include an improved scheme, in which the aforementioned weaknesses are eliminated while some desirable features are added.

II. Review of Juang *et al.*'s Scheme

For a self-contained discussion, we briefly describe the scheme of Juang *et al.* [16] before demonstrating its weaknesses.

A. Parameter-Generation Phase

In this phase, S chooses an elliptic curve E over a finite field Z_p with a large prime number p, finds a generator point G with order n, selects a random number x as its private key, and computes the public key $P_S = x \times G$. Then, S publishes the parameters (p, E, G, P_S, n).

B. Registration Phase

This phase is invoked whenever U initially registers or reregisters to S, and U can use his smart card after this phase. U and S perform the following steps.

Step 1) U selects a password PW and a random number b and then computes $h(PW\|b)$. U submits the identifier ID and the value $h(PW\|b)$ to S for registration via a secure channel.

Step 2) If ID is a new identifier, then S sets the card identifier $CI=1$ and stores the record $\{ID, CI\}$ in its registration table. If S issues a new card to U that registered before, then S gets the record $\{ID, CI\}$ from its registration table, computes $CI=CI+1$, and updates the record $\{ID,CI\}$ in its registration table. Then, S generates $V=h(ID,K_S,CI)$ and $IM=E_{K_S}(h(PW\|b)\|ID\|CI\|TAG)$, where $TAG=h(ID\|CI\|h(PW\|b))$. S issues a smart card to U that contains the parameters V, IM, ID, and CI.

Step 3) U stores the number b into the smart card.

C. Precomputation Phase

Before the start of the log-in phase, the smart card generates the nonce N_C and then computes and stores two points e and c over the elliptic curve E, where $e = N_C \times G$ and $c = N_C \times P_S$.

D. Log-in Phase

After the precomputation phase, this phase is invoked whenever U wants to log on to S. If U and S complete this phase successfully, they can authenticate each other and use the session key K_{SU} in the subsequent secret communication. The steps of this phase are shown as follows.

Step 1) U inserts his smart card into a card reader and inputs his password PW. The smart card computes $E_V(e)$, where $V = h(ID, K_S, CI)$. The smart card further sends the message $\{IM, E_V(e)\}$ to S.

Step 2) After receiving the message $\{IM, E_V(e)\}$, S decrypts the parameter IM by the master secret key K_S and obtains the value $h(PW\|b)\|ID\|CI\|TAG$, and then, S computes $V = h(ID, K_S, CI)$. Therefore, S can use the key V to decrypt $E_V(e)$ for the point e. Then, S checks if the following are true: 1) the value TAG is equal to $h(ID\|CI\|h(PW\|b))$; 2) the identifier ID is correct; and 3) the card identifier CI is correct. If any of the verifications is false, S terminates this session. If all of the verifications are true, S computes $c = x \times e$ and $M_S = h(c\|N_S\|V)$, where N_S is the nonce chosen by S. S sends the message $\{N_S, M_S\}$ to the smart card.

Step 3) After receiving the message $\{N_S, M_S\}$, the smart card computes and checks if the value M_S is equal to $h(c\|N_S\|V)$. If it is not, the smart card terminates this session. Otherwise, the smart card computes $K_{SU} = h(V, c, N_S)$ and $M_U = h(h(PW\|b)\|V\|c\|N_S)$ and then sends the message $\{M_U\}$ to S.

Step 4) Upon receiving the message $\{M_U\}$, S checks if the value M_U is equal to $h(h(PW\|b)\|V\|c\|N_S)$. If it is not, S sends a wrong password message back to U. U can input the password PW. Then, the smart card computes M_U and sends the message $\{M_U\}$ to S again. If the number of the password verifications exceeds the allowed times, S terminates this session. Otherwise, S accepts the log-in request and computes $K_{SU} = h(V, c, N_S)$.

Note that all messages in this phase are transmitted via an insecure channel.

E. Password-Changing Phase

When U wants to change his password, the session key K_{SU} needs to be established through the log-in phase in advance. This phase requires the following steps.

Step 1) U enters a new password PW^* and a new random number b^*. The smart card computes

$E_{K_{SU}}(ID, h(PW^*\|b^*))$ and sends the message $\{E_{K_{SU}}(ID, h(PW^*\|b^*))\}$ to S.

Step 2) After receiving the message $\{E_{K_{SU}}(ID, h(PW^*\|b^*))\}$ and decrypting $E_{K_{SU}}(ID, h(PW^*\|b^*))$ to obtain the parameters ID and $h(PW^*\|b^*)$, S computes $IM^* = E_{K_S}(h(PW^*\|b^*)\|ID\|CI\|h(ID\|CI\|h(PW^*\|b^*)))$ and then sends the message $\{E_{K_{SU}}(IM^*)\}$ to the smart card.

Step 3) Upon receiving the message $\{E_{K_{SU}}(IM^*)\}$, the smart card decrypts $E_{K_{SU}}(IM^*)$ and replaces the old parameters IM and b with the new parameters IM^* and b^*, respectively.

Note that all messages in this phase are transmitted via an insecure channel.

III. Weaknesses of Juang *et al.*'s Scheme

In this section, we present three weaknesses of the scheme of Juang *et al.*

A. Inability of Password-Changing Operation

Consider that U successfully completes the password-changing operation. It means that the smart card replaces the old parameters IM and b with the new parameters IM^* and b^*, respectively. Obviously, the smart card can use the new parameters V, IM^*, b^*, and PW^* to log on to S. However, the old parameters V, IM, b, and PW are still valid to log on to S, because $IM = E_{K_S}(h(PW\|b)\|ID\|CI\|h(ID\|CI\|h(PW\|b)))$, and S does not directly check the parameters PW and b in the log-in phase. We need to point out that U often executes the password-changing operation when the old parameters V, IM, b, and PW are compromised. In this case, the attacker can impersonate U no matter whether U changes his password. Furthermore, this weakness may potentially destroy a security service called nonrepudiation, which means no denial of a connection with the operation. It is possible that, after the password-changing operation, U still uses the old parameters V, IM, b, and PW to log on to S and then denies this log-in operation by showing his new parameters IM^*, b^*, and PW^*. Usually, nonrepudiation is a necessary security requirement in electronic commerce applications.

Another drawback of the password-changing operation is that it is vulnerable to the so-called denial-of-service attack. If the attacker can determine the password-changing phase, he can block the message $\{E_{K_{SU}}(IM^*)\}$ in Step 2) of the password-changing phase and then randomly send a same-length message $\{IM_f \neq E_{K_{SU}}(IM^*)\}$ to the smart card. Since it is not able to verify whether the message $\{IM_f\}$ is valid, the smart card should replace the old parameters with the error parameters. After that, U cannot use the smart card to log on to S without the reregistration operation.

B. Session-Key Problem

As in the definitions in [19], a key agreement scheme is said to provide the explicit key confirmation if one entity is assured that the second entity has actually computed the session key. The scheme provides the implicit key confirmation if one entity is assured that the second entity can compute the session key. Note that the property of the implicit key confirmation does not necessarily mean that one entity is assured of the second entity actually possessing the session key. In many applications, it is highly desirable for a key agreement scheme to provide the explicit key confirmation. We can see that the scheme of Juang *et al.* merely provides the implicit key confirmation, because both U and S cannot confirm that the other has correctly computed $K_{SU} = h(V, c, N_S)$ after the log-in phase. However, in general, the three-pass key agreement scheme can provide the explicit key confirmation. Hence, the scheme of Juang *et al.* is inefficient due to the three exchanged messages in the log-in phase.

Furthermore, the scheme of Juang *et al.* does not provide forward secrecy [19] for the session key. Forward secrecy requires that, if long-term private keys of one or more entities are compromised, the secrecy of previous session keys established by honest entities can be unaffected. Obviously, if the attacker obtains the master secret key K_S and the private key x for the elliptic-curve algorithm, he can compute any previous session key $K_{SU} = h(V, c, N_S)$ with the help of the corresponding messages $\{IM, E_V(e)\}$ and $\{N_S, M_S\}$ transmitted via the insecure channel.

C. Inefficiency of Double Secret Keys

We can see that the scheme of Juang *et al.* requires S to keep two keys secret, i.e., the master secret key K_S and the private key x for the elliptic-curve algorithm. In common sense, it is possible to only use one secret key for achieving the user authentication and key agreement service. Therefore, two secret keys mean more overheads without the security enhancement for the whole authentication system. Furthermore, we need to point out the drawback of using the elliptic-curve algorithm in the scheme of Juang *et al.* Since S uses the private–public key pair $\{x, P_S = x \times G\}$, this elliptic-curve algorithm is a public key algorithm, which may involve the certificate mechanism, e.g., X.509 [20]. To maintain the certificate framework, the public key infrastructure incurs a nontrivial level of system complexity and implementation costs.

IV. Our Improved Scheme

To overcome the aforementioned weaknesses, we propose an improved scheme, which consists of the parameter-generation phase, the registration phase, the authentication phase, and the password-change phase. If the size of the message is larger than the size of the block of the symmetric encryption algorithm, our improved scheme employs the cipher-block-chaining mode [21] to provide a protection against unauthorized data modification such as deletion or insertion.

A. Parameter-Generation Phase

S chooses an elliptic curve E over a finite field F_p such that the discrete logarithm problem is hard in $E(F_p)$. The set of all the points on E is denoted by $E(F_p)$. S also chooses a point

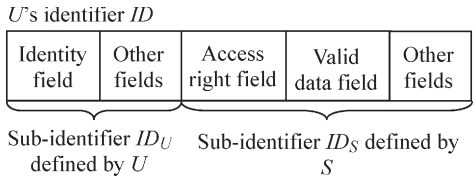

Fig. 1. Verifiable identifier.

$G \in E(F_p)$ such that the subgroup generated by G has a large order n. S publishes the parameters (p, E, G, n).

B. Registration Phase

This phase is invoked whenever U registers or reregisters to S. U and S perform the following steps.

Step 1) U selects the sub-identifier ID_U following the appointed format and then submits it to S for registration via a secure channel.

Step 2) If the sub-identifier ID_U is valid, S selects the sub-identifier ID_S and generates the identifier $ID = ID_U \| ID_S$ for U. Then, S generates $V = h(ID \| K_S) \oplus h(PW)$ and $IM = E_{K_S}(ID \| r)$, where PW is the initial password selected by S and r is a random number to provide the identity protection.

Step 3) S issues the password PW and the smart card to U, where the smart card contains the public parameter IM and the private parameter V.

Explanation:

1) In the scheme of Juang *et al.*, U selects and submits the initial password to S. Therefore, it needs the parameter b to prevent the insider attack [10]. However, S determines the initial password for U in our improved scheme. This design not only well adapts to the style of the card issuer but also thwarts the insider attack. After receiving the smart card, U is able to immediately change the initial password using the offline password-changing operation.
2) Our improved scheme requires that U's identifier ID should have the appointed format, which can be directly verified by S. This rule avoids the registration table in the scheme of Juang *et al.* Fig. 1 shows a simple example.

C. Authentication Phase

This phase is invoked whenever U wants to log on to S. After a successful completion of this phase, U and S can authenticate each other and share the session key K_{SU} for the subsequent secret communication. The steps of this phase are shown as follows.

Step 1) U inserts his smart card into a card reader and inputs his password PW. The smart card randomly selects an integer r_C from the interval $[1, n-1]$ and computes $G_C = r_C \times G$ and then sends the message $\{IM, G_C\}$ to S.

Step 2) Upon receiving the message $\{IM, G_C\}$, S decrypts the parameter IM by the master secret key K_S and obtains the value $ID \| r$. Then, S verifies whether the identifier ID is valid. If the verification is false, S terminates this session. If the verification is true, S generates $G_S = r_S \times G$, where r_S is an integer selected at random from the interval $[1, n-1]$, and then computes $K_{SU} = h_1(h(ID \| K_S) \| (r_S \times G_C))$ and $M_S = h_2(K_{SU} \| G_C \| G_S)$. S sends the message $\{M_S, G_S\}$ to the smart card.

Step 3) Upon receiving the message $\{M_S, G_S\}$, the smart card computes $V' = V \oplus h(PW)$ and $K_{SU} = h_1(V' \| (r_C \times G_S))$ and further checks whether the value M_S is equal to $h_2(K_{SU} \| G_C \| G_S)$. If it is not, the smart card terminates this session. Otherwise, the smart card computes $M_U = h_2(K_{SU} \| G_S)$ and then sends the message $\{M_U\}$ to S.

Step 4) Upon receiving the message $\{M_U\}$, S checks whether the value M_U is equal to $h_2(K_{SU} \| G_S)$. If it is correct, U and S successfully authenticate each other and establish the session key K_{SU}. Otherwise, S terminates this session.

Explanation:

1) If U and S correctly execute the aforementioned steps in the authentication phase, they can accept each other's identity and establish the session key. The reason is that $r_S \times G_C = r_S \times r_C \times G = r_C \times G_S$.
2) It needs to point out that our improved scheme employs the elliptic-curve Diffie–Hellman algorithm to achieve the forward security.

D. Password-Change Phase

This phase is invoked whenever U wants to change his password PW with a new one, for example, PW^*.

Step 1) U inserts his smart card into the smart-card reader of a terminal, enters the old password PW, and requests to change password. Next, U enters the new password PW^*.

Step 2) U's smart card computes $V^* = V \oplus h(PW) \oplus h(PW^*)$, which yields $h(ID \| K_S) \oplus h(PW^*)$, and then replaces V with V^*.

Explanation: Unlike the scheme of Juang *et al.*, U can freely change his password without any interaction with S. S can be totally unaware of the change of U's password. Hence, it reduces the possibility of the insider attack.

V. Evaluations of Our Improved Scheme

The characteristics of the scheme of Juang *et al.* have already been demonstrated in [16]. In this section, we focus on the security, functionality, and efficiency of our improved scheme.

A. Security Analysis

To evaluate the security of our improved scheme, we need to assume the capabilities that the attacker may have under the smart-card-based authentication environments. We define two adversarial models as follows.

1) Basic Model. The attacker is allowed to fully control the communication channel between S and any of U. He can

inject, modify, block, and delete messages at will. He can also request any session keys adaptively. However, he is not allowed to compromise the long-term secret keys. All authenticated key agreement schemes including the smart-card-based schemes must thwart this type of attacker. Hence, we call it as the basic model.

2) Special Model. The attacker is allowed to either compromise U's smart card or compromise U's password, but not both. Clearly, the attacker that compromises both U's smart card and U's password can impersonate U, since both of them precisely identify U. It is a trivial case.

Having defined attacker behavior, we can naturally divide the security of the scheme into three cases.

1) General case. The attacker merely has all the capabilities defined as the basic model.
2) Smart-card-loss case. The attacker not only has all the capabilities defined as the basic model but also compromises U's smart card defined as the special model. It means that the attacker knows all parameters stored on U's smart card. Note that we put aside any special security feature that could be supported by the smart card.
3) Password-loss case. The attacker not only has all the capabilities defined as the basic model but also compromises U's password of the smart card defined as the special model.

We can easily see that, if our improved scheme is secure in the aforementioned three cases, our improved scheme is also secure under both the basic and special models. That is, our improved scheme provides the sound security promises. We demonstrate them as follows.

Claim 1. Our Improved Scheme is Secure in the General Case: If the attacker does not obtain U's smart card and password PW, we can omit the smart-card factor and treat our improved scheme as an authenticated key agreement scheme. The reason is that both U and S merely use the shared secret key $h(ID\|K_S)$ to authenticate each other and establish the session key. The Bellare–Rogaway model [8] can evaluate the security of the authenticated key agreement scheme. Based on the Bellare–Rogaway model, we prove our improved scheme secure under the following assumptions: 1) The elliptic-curve Diffie–Hellman problem is hard; 2) the hash function $h()$ is the pseudorandom permutation for key derivation; 3) the hash function $h_1()$ can be treated as the random oracle; and 4) the hash function $h_2()$ is the secure message authentication code. The proof needs a lot of background knowledge related to provable security but has no any trick. Therefore, we omit it for simplicity. Note that, once passing this security evaluation, our improved scheme can achieve the goal of user authentication and key agreement with great assurance and certainly can prevent the well-known attacks, such as the replay, parallel-session, reflection, interleaving, and man-in-the-middle attacks.

Claim 2. Our Improved Scheme is Secure in the Smart-Card-Loss Case: If the attacker obtains U's smart card in our improved scheme, he can derive the parameters IM and V from the card. Since the parameter IM is public, the attacker additionally learns the parameter V compared with the general case. We can see that $V = h(ID\|K_S) \oplus h(PW)$. Without U's password PW, the attacker cannot directly use the parameter V to corrupt U's authentication session. The extra threat in the smart-card-loss case arises from the fact that the password space is usually small and much easier to attack than random cryptographic keys [22]. For U, it is very difficult and troublesome to memorize a long or irregular password. Hence, the attacker can guess U's password PW in a relatively small dictionary and then verify the guessed password using available information and devices. This dictionary attack can be further classified as the online and offline dictionary attacks. The online dictionary attack is easier to detect and limit, since the attacker actively tries different passwords against S. The standard ways of preventing such online attack in practice are to either limit the number of failed runs that U is allowed to have before U's password PW is expired or reduce the rate at which U is allowed to make log-in attempts. The offline dictionary attack is very powerful since it can be performed offline; therefore, the attacker does not need to interact with the legitimate entities and can use a lot of computing power. Therefore, we only consider the offline dictionary attack on our improved scheme. The messages $\{IM, G_C\}$, $\{M_S, G_S\}$, and $\{M_U\}$ of a legitimate authentication session and U's parameter V cannot help the attacker to verify the guessed password, because the corresponding value $r_S \times r_C \times G$ is not available. Furthermore, the password-change phase does not assist the offline dictionary attack, because of the absence of any verification step. Based on the aforementioned analysis, we find that the extra parameter V cannot enhance the attacker's capabilities of breaking our improved scheme compared with the general case. Therefore, we claim that our improved scheme is secure in the smart-card-loss case, because this scheme is secure in the general case.

In the smart-card-loss case, the attacker can randomly choose two passwords and then use them and U's smart card to change the password by invoking the password-change phase. If U gets back his smart card, U's succeeding log-in requests will be denied. However, for practical purposes, such a denial-of-service attack in itself is not considered as a security breach. U should accept this trouble just as someone, who loses his key for the door of his house, can burden another new key and lock for the door. After all, U loses his smart card. Moreover, U can reregister to S by invoking the registration phase.

Claim 3. Our Improved Scheme is Secure in the Password-Loss Case: In our improved scheme, the password-loss case means that the attacker learns U's password PW. However, it cannot assist the attacker to break our improved scheme, because all values IM, G_C, M_S, G_S, and M_U in the legitimate authentication session have no relation with U's password PW. Therefore, we claim that our improved scheme is secure in the password-loss case, because this scheme is secure in the general case.

B. Functionality Consideration

In this section, we discuss the security functionalities of our improved scheme, which are not provided in the scheme of Juang *et al.*, and then draw a simple comparison between them. We examine our improved scheme as follows.

1) Usability of the Password-Changing Operation: Consider the password-change phase of our improved scheme. If U wants to change his password PW with a new password PW^*, U's smart card should replace $V = h(ID\|K_S) \oplus h(PW)$ with $V^* = h(ID\|K_S) \oplus h(PW^*)$ after successfully finishing the password-changing operation. If anyone tries to use the parameters V^* and PW to log on to S at this time, S should terminate the authentication session. Our improved scheme does not suffer from the nonrepudiation problem, because U is responsible for his password by himself and need not interact with S during the password-change phase. In fact, S identifies U by the value $h(ID\|K_S)$. Due to the offline password-changing operation, our improved scheme can prevent the denial-of-service problem that existed in the scheme of Juang *et al.*

2) Desirable Key Properties: In our improved scheme, U can be assured that S has actually computed $K_{SU} = h_1(h(ID\|K_S)\|(r_S \times G_C))$, after he successfully completed Step 3) of the authentication phase. The reason is that S needs the correct session key K_{SU} to generate the value M_S, which is equal to $h_2(K_{SU}\|G_C\|G_S)$. For the same reason, S can be assured that U has actually computed $K_{SU} = h_1(V'\|(r_C \times G_S))$, after S verified that the value M_U is equal to $h_2(K_{SU}\|G_S)$ in Step 4) of the authentication phase. Hence, using three exchanged messages in the authentication phase, our improved scheme can provide the explicit key confirmation. Furthermore, our improved scheme achieves forward secrecy for the session key. Even if the long-term secret key $h(ID\|K_S)$ or K_S is compromised, the attacker cannot get the previous session key $K_{SU} = h_1(h(ID\|K_S)\|(r_S \times r_C \times G))$ without the value r_S or r_C.

3) Small Verification Table: The scheme of Juang *et al.* does not require the password table but needs to maintain a verification table to store the parameters ID and CI for each U. The verification table must prevent the illegal modification. Otherwise, the scheme of Juang *et al.* at least suffers from the denial-of-service attack. Our improved scheme needs the verification table only to store the revoked identifiers ID. Since few identifiers ID usually can be compromised, the verification table in our improved scheme is smaller than that in the scheme of Juang *et al.*, particularly when there are a large number of users in the authentication system.

4) More Identity Protection: As in the claim of Juang *et al.*, the identifier ID is included in the parameter IM, which is sent to S and is protected by the symmetric encryption algorithm. Only S can decrypt the parameter IM by using the master secret key K_S and get the identifier ID. The identifier ID is never explicitly transmitted via the insecure channel. Therefore, both schemes can provide the user's identity protection. Consider the smart-card-loss case. Since the smart card contains the parameters V, IM, ID, CI, and b in the scheme of Juang *et al.*, the attacker can obtain the identifier ID. However, the attacker cannot get the identifier ID in our improved scheme, because he cannot derive the identifier ID from the parameters V and IM without the master secret key K_S.

Table I summarizes the security functionalities that are believed to be provided by the scheme of Juang *et al.* and our improved scheme. The names of the functionalities have been abbreviated to save space: NPT denotes no password table, PU denotes password usability, IKA denotes implicit key authentication, EKA denotes explicit key authentication, FS denotes forward secrecy, VT denotes verification table, and IDP denotes identity protection.

TABLE I
FUNCTIONALITY: SCHEME OF JUANG *et al.* VERSUS OUR IMPROVED SCHEME

Security Functionality \ Scheme		Scheme of Juang et al.	Our Improved Scheme
Password	NPT	Yes	Yes
	PU	Register	Register / Change
Session key	IKA	Yes	Yes
	EKA	No	Yes
	FS	No	Yes
Others	VT	Practical	Very practical
	IDP	Half	Full

TABLE II
STORAGE COST: SCHEME OF JUANG *et al.* VERSUS OUR IMPROVED SCHEME

Storage Cost \ Scheme	Scheme of Juang et al.	Our Improved Scheme
Smart Card	512 bits	256 bits
Server	291 bits	128 bits

C. Efficiency Comparison

Herein, we only compare our improved scheme with the scheme of Juang *et al.*, because both schemes employ similar cryptographic tools. We follow all parameter assumptions in the evaluation of Juang *et al.* For the scheme of Juang *et al.*, we do not take account of the part *TAG* in the parameter *IM*.

Assume that the block size of the symmetric encryption algorithm is 128 bits, and the output size of the cryptographic hash function is 128 bits. Assume that the modulus in the elliptic-curve algorithm is of 163 bits. This means that it needs $163 \times 2 = 326$ bits to store a point in the elliptic curve.

In the storage cost concern, our improved scheme requires the smart card to store the parameters V and IM instead of the parameters V, IM, ID, CI, and b in the scheme of Juang *et al.* We can further estimate that the parameters V, IM, ID, CI, and b in the scheme of Juang *et al.* need $128 + 256 + 32 + 32 + 64 = 512$ bits of storage space, where both the identifier ID and the card identifier CI can be 32 bits, the random number b can be 64 bits, and the parameter IM must require two blocks. Correspondingly, the parameters V and IM in our improved scheme need $128 + 128 = 256$ bits of storage space, where the identifier ID can be 64 bits, the random number r can be 64 bits, and the parameter IM requires one block. For the storage costs in S, we focus on the secret parameters, because S must expend more resources to protect them. S needs a 128-bit storage space for the secret parameter K_S in our improved scheme. In the scheme of Juang *et al.*, S needs about $163 + 128 = 291$ bits of storage space for the secret parameters x and K_S. We list the storage costs of the scheme of Juang *et al.* and our improved scheme in Table II.

TABLE III
COMMUNICATION COST: SCHEME OF JUANG *et al.* VERSUS OUR IMPROVED SCHEME

Communication Cost \ Scheme	Scheme of Juang et al.	Our Improved Scheme
Session Run	966 bits	1036 bits
Password Operation	512 bits	---

TABLE IV
COMPUTATION COST: SCHEME OF JUANG *et al.* VERSUS OUR IMPROVED SCHEME

Computation Cost \ Scheme		Scheme of Juang et al.	Our Improved Scheme
Smart Card	Registration Operation	1 H	---
	Session Run	2 M +4 H+1 E	2 M +4 H
	Password Operation	1 H +2 E	2 H
Server	Registration Operation	1 H+1 E	2 H+1 E
	Session Run	1 M+4 H+2 E	2 M +4 H +1 E
	Password Operation	3 E	---

Consider the communication costs. In a normal session run, our improved scheme needs exchange data IM, G_C, M_S, G_S, and M_U, while that of Juang *et al.* need exchange data IM, $E_V(e)$, N_S, M_S, and M_U. Let the nonce N_S be 128 bits. The communication cost of a normal session run is $128 + 326 + 128 + 326 + 128 = 1036$ bits in our improved scheme and $256 + 326 + 128 + 128 + 128 = 966$ bits in that of Juang *et al.* For the password-changing operation, our improved scheme need not exchange any data, while that of Juang *et al.* needs exchange data $E_{K_{SU}}(ID, h(PW^*\|b^*))$ and $E_{K_{SU}}(IM^*)$. Similarly, we can calculate the communication cost of the password-changing operation for the scheme of Juang *et al.* In Table III, we show the communication costs of the scheme of Juang *et al.* and our improved scheme.

In Table IV, we tabulate the computation costs of the registration operation, the session run, and the password-changing operation for both the scheme of Juang *et al.* and our improved scheme. The names of the computation operations have been abbreviated to save space: H denotes the cryptographic hash computation, E denotes the symmetric encryption or decryption computation, and M denotes the scalar multiplication computation over the elliptic curve. Consider the computation cost of the smart card. We can see that our improved scheme is a little more efficient than the scheme of Juang *et al.* Due to the limited hardware resources, the smart card is always unable to provide powerful computation capability. Hence, it is a desirable feature. However, our improved scheme requires S to perform an extra scalar multiplication computation compared with the scheme of Juang *et al.* Although the computation cost of S is dominated by the scalar multiplication computation, we need to point out that it is unimportant, because S always has powerful computation capability in most application environments.

VI. CHALLENGES ON OUR IMPROVED SCHEME

It is a difficult task to design the password-authenticated key agreement scheme using smart cards, because the designers face the difficult task of reconciling security, functionality, and efficiency requirements and sometimes must make design decisions that appear well motivated but have unintended consequences. We outline two challenges on our improved scheme.

1) The first challenge is from the security of the smart card in our improved scheme. The security of the smart card is mainly dominated by the attack ability. To our best knowledge, currently, no technique can estimate the attack ability for the smart card in a general way. Therefore, the definition of the smart-card model is a tough job due to dramatic different potential attacks, security requirements, and environments, which are always determined by many unknown factors. For designing our improved scheme, we assume that, if the attacker obtains the smart card, he can know all parameters stored on the smart card. However, new attacks on the smart card may compromise our improved scheme in some ways.
2) The second challenge is due to symmetric cryptographic techniques in our improved scheme. We avoid the asymmetric cryptographic technique to achieve efficient implementations in practice. However, it leads to the fact that our improved scheme cannot be efficiently extended to the multiserver scenario due to the key management and trust problem. In fact, how to adapt our improved scheme to the multiserver scenario is a meaningful topic that deserves further research.

VII. CONCLUSION

Recently, Juang *et al.* proposed a password-authenticated key agreement scheme using smart cards. In this paper, we have shown the weaknesses of the scheme of Juang *et al.* and further proposed an improved scheme. Our improved scheme not only preserves the benefits of the scheme of Juang *et al.* but also fixes its weaknesses. In addition, our improved scheme further reduces the storage and computation costs on the smart card compared with that of Juang *et al.* Therefore, we believe that our improved scheme is more suitable for real-life applications than that of Juang *et al.*

ACKNOWLEDGMENT

The authors would like to thank the editors and the reviewers for their useful suggestions and comments.

REFERENCES

[1] C. L. Hwang, L. J. Chang, and Y. S. Yu, "Network-based fuzzy decentralized sliding-mode control for car-like mobile robots," *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 574–585, Feb. 2007.
[2] G. P. Liu, Y. Xia, J. Chen, D. Rees, and W. Hu, "Networked predictive control of systems with random network delays in both forward and feedback channels," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1282–1297, Jun. 2007.
[3] C. Lazar and S. Carari, "A remote-control engineering laboratory," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2368–2375, Jun. 2008.
[4] D. Dondi, A. Bertacchini, D. Brunelli, L. Larcher, and L. Benini, "Modeling and optimization of a solar energy harvester system for self-powered wireless sensor networks," *IEEE Trans. Ind. Electron.*, vol. 55, no. 7, pp. 2759–2766, Jul. 2008.
[5] F. Gil-Castiñeira, F. J. González-Castaño, and L. Franck, "Extending vehicular CAN fieldbuses with delay-tolerant networks," *IEEE Trans. Ind. Electron.*, vol. 55, no. 9, pp. 3307–3314, Sep. 2008.

[6] P. Mariño, F. Poza, M. A. Domínguez, and S. Otero, "Electronics in automotive engineering: A top–down approach for implementing industrial fieldbus technologies in city buses and coaches," *IEEE Trans. Ind. Electron.*, vol. 56, no. 2, pp. 589–600, Feb. 2009.
[7] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1997.
[8] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Proc. 13th Advances Cryptology–CRYPTO*, D. R. Stinson, Ed, 1994, LNCS, vol. 773, pp. 232–249.
[9] M. Bellare and P. Rogaway, "Provably secure session key distribution—The three party case," in *Proc. 27th Annu. ACM Symp. Theory Comput.*, 1995, pp. 57–66.
[10] W. C. Ku and S. M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 204–207, Feb. 2004.
[11] D. Z. Sun and Z. F. Cao, "Improvement of Lee-Kim-Yoo's remote user authentication scheme using smart cards," in *Proc. 2nd Int. Conf. FSKD*, L. Wang and Y. Jin, Eds, 2005, LNAI, vol. 3614, pp. 596–599.
[12] D. Z. Sun, J. D. Zhong, and Y. Sun, "Weakness and improvement on Wang-Li-Tie's user-friendly remote authentication scheme," *Appl. Math. Comput.*, vol. 170, no. 2, pp. 1185–1193, Nov. 2005.
[13] T. F. Cheng, J. S. Lee, and C. C. Chang, "Security enhancement of an IC-card-based remote login mechanism," *Comput. Netw.*, vol. 51, no. 9, pp. 2280–2287, Jun. 2007.
[14] J. Y. Liu, A. M. Zhou, and M. X. Gao, "A new mutual authentication scheme based on nonce and smart cards," *Comput. Commun.*, vol. 31, no. 10, pp. 2205–2209, Jun. 2008.
[15] D. Z. Sun, J. P. Huai, J. Z. Sun, J. W. Zhang, and Z. Y. Feng, "A new design of wearable token system for mobile device security," *IEEE Trans. Consum. Electron.*, vol. 54, no. 4, pp. 1784–1789, Nov. 2008.
[16] W. S. Juang, S. T. Chen, and H. T. Liaw, "Robust and efficient password-authenticated key agreement using smart cards," *IEEE Trans. Ind. Electron.*, vol. 55, no. 6, pp. 2551–2556, Jun. 2008.
[17] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Side channel cryptanalysis of product ciphers," in *Proc. 5th ESORICS*, J. J. Quisquater, Y. Deswarte, C. Meadows, and D. Gollmann, Eds, 1998, LNCS, vol. 1485, pp. 97–110.
[18] D. Z. Sun, J. P. Huai, J. Z. Sun, and Z. F. Cao, "An efficient modular exponentiation algorithm against simple power analysis attacks," *IEEE Trans. Consum. Electron.*, vol. 53, no. 4, pp. 1718–1723, Nov. 2007.
[19] S. Blake-Wilson and A. Menezes, "Authenticated Diffie–Hellman key agreement protocols," in *Proc. 5th Annu. Int. Workshop SAC*, S. Tavares and H. Meijer, Eds, 1999, LNCS, vol. 1556, pp. 339–361.
[20] ITU-T, *Recommendation X.509-the Directory: Public-Key and Attribute Certificate Frameworks, ITU-T Study Group 17*, Aug. 2005. (equivalent to ISO/IEC 9594-8).
[21] NIST, *Recommendation for Block Cipher Modes of Operation, NIST Special Publication 800-38A 2001 Edition*, Dec. 2001, Washington DC: U.S. Dept. Commerce/NIST.
[22] S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 3, pp. 230–268, Aug. 1999.

Da-Zhi Sun received the B.E. degree in electronic engineering from Nanchang Institute of Aeronautical Technology, Jiangxi, China, in 1999, the M.E. degree in control science and engineering from Nanchang University, Jiangxi, in 2002, and the Ph.D. degree in computer science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2006.

He is currently a Lecturer with the School of Computer Science and Technology, Tianjin University, Tianjin, China. He is also with the School of Computer Science, Beihang University, Beijing, China. His current research interests include applied number theory, applied cryptography, and information security.

Jin-Peng Huai was born in Harbin, China, in 1962. He received the Ph.D. degree in computer science and technology from Beihang University, Beijing, China.

He is currently a Professor with the School of Computer Science, Beihang University. He has published more than 90 papers on the Internet, web service technology, and network security. His research interests are in the areas of computer software and information security.

Ji-Zhou Sun received the Ph.D. degree from the Department of Computing (Informatics), University of Sussex, Sussex, U.K.

He is currently a Professor with the School of Computer Science and Technology, Tianjin University, Tianjin, China. His research interests include virtual reality, network and parallel computing, and image processing.

Jian-Xin Li received the Ph.D. degree from the School of Computer Science, Beihang University, Beijing, China, in 2008.

He is an Assistant Professor and a Research Staff Member with the School of Computer Science, Beihang University. He has authored over 20 papers in the *International Journal of Peer-to-Peer Networking and Applications*, Symposium on Reliable Distributed Systems, Symposium on High Assurance Systems Engineering, eScience, etc. His research interests include information security, trust management, and distributed systems.

Jia-Wan Zhang (M'06) received the Ph.D. degree from the School of Computer Science and Technnology, Tianjin University, Tianjin, China.

He is currently an Associate Professor with the School of Computer Science, Tianjin University. His current research interests include information visualization, scientific visualization, and image processing.

Dr. Zhang is a member of the Association for Computing Machinery.

Zhi-Yong Feng received the Ph.D. degree from the School of Computer Science and Technnology, Tianjin University, Tianjin, China.

He is a Professor with the School of Computer Science and Technology, Tianjin University. He has published over 80 papers. His research interests include Web service, electronic commerce security, and knowledge engineering.