Hindawi Publishing Corporation
Mobile Information Systems
Volume 2015, Article ID 934716, 7 pages
http://dx.doi.org/10.1155/2015/934716

Research Article

A Multipurpose Key Agreement Scheme in Ubiquitous Computing Environments

Chin-Chen Chang,[1] Iuon-Chang Lin,[2] and Chia-Chi Wu[3]

[1] *Department of Information Engineering and Computer Science, Feng Chia University, Taichung 40724, Taiwan*
[2] *Department of Management Information Systems, National Chung Hsing University and Department of Photonics and Communication Engineering, Asia University, Taichung 413, Taiwan*
[3] *Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan*

Correspondence should be addressed to Iuon-Chang Lin; iclin@nchu.edu.tw

Received 13 March 2015; Revised 23 October 2015; Accepted 29 October 2015

Academic Editor: Hua Lu

Due to the rapid advancement of cryptographic techniques, the smart card has recently become a popular device because it is capable of storing and computing essential information with such properties as tamper resistance. However, many service providers must satisfy the user's desire to be able to access services anytime and anywhere with the smart card computing devices. Therefore, multipurpose smart cards have become very popular identification tokens. In 2011, Wang et al. proposed an authentication and key agreement scheme for smart card use. Even so, two drawbacks still exist; that is, (1) the security requirement of mutual authentication has not been satisfied and (2) the authentication scheme cannot be used for multipurpose smart cards. In this paper, we propose an efficient and secure multipurpose, authenticated, key agreement scheme in which the user is required to register only once and can be authenticated without any registration center. Furthermore, the proposed scheme can be used in ubiquitous environments because of its low computation and communication overhead.

1. Introduction

Currently, the uses of smart cards include shopping, taking buses or subway, paying bills, parking cars, and passing through guarded gates. When the smart card is embedded in a mobile phone, many commercial transactions can be performed in ubiquitous computing environments. Therefore, multipurpose smart cards are very popular identification tokens, and service providers must satisfy the user's desire to be able to access services anytime and anywhere with the smart card computing devices. However, in the ubiquitous computing environment, the communication channels are insecure and may suffer from eavesdropping, interception, and impersonation attacks [1]. Hence, we must simultaneously consider both service and security requirements to protect the rights and the privacy of users and providers [2]. These ubiquitous computing devices usually are small with limited computation and communication capabilities. Therefore, it is a difficult challenge to deploy comprehensive security mechanisms in the ubiquitous computing environment.

Although the smart card can be used to authenticate a user's identity and perform electronic transactions, we must still consider the risk of accidental loss of the cards. Therefore, establishing a password is the most popular method for protecting the user.

In general, people choose words that are easy to remember or word strings with special meanings as passwords, but just using a password for authentication can easily make the user vulnerable to security breaches. Hence, the smart card is applied to improve the authentication security. As a result, most e-commercial transactions use both the smart cards and the passwords to ensure authentication and maintain security. Over the past two decades, many schemes have been proposed to achieve both user authentication and confidentiality of messages based on smart cards. In 1981, Lamport [3] proposed the well-known remote user authentication scheme with password tables. In 1993, to provide better security, Chang and Hwang proposed a novel multiserver authentication scheme [4] without password tables. Afterwards, many related research essays [5–11] have been proposed to improve the security and performance of authentication.

In 2004, Das et al. proposed a dynamic ID-based remote user authentication scheme [6] using smart cards. However, it had a serious security flaw; that is, if a malicious attacker gets the smart card, he or she can freely choose passwords to be authenticated by the server. In 2009, Wang et al. proposed an improved scheme [12] to enhance Das et al.'s scheme, but Khan et al. [13] found that Wang et al.'s scheme is infeasible because it cannot provide a secure communication channel between users and servers. Thus, Khan et al. proposed an enhanced scheme [13] to overcome these weaknesses. However, Khan et al.'s scheme cannot be applied in multipurpose and ubiquitous environments.

In 2011, Wang et al. proposed an improved scheme [14] to solve the problems associated with losing a smart card and the known-key attacks, which are vulnerabilities that exist in Wang et al.'s scheme [15] in 2007. They claimed that their scheme can achieve the following criteria [14]:

(C1) *No verification table*: no verification or password table is stored at the server's end.

(C2) *Freely chosen password*: users can arbitrarily choose and change their passwords.

(C3) *The server administrator being not able to derive the user's passwords*: even the administrator will not obtain privilege to derive the user's passwords.

(C4) *No one being able to impersonate a valid user*: the authentication scheme must completely resist impersonation attacks.

(C5) *No clock synchronization or time-delay problems*: it can get higher performance and better reduce synchronization cost than others.

(C6) *Mutual authentication*: the scheme should resist reply, password-guessing, known-key, and stolen-verifier attacks.

(C7) *Session key agreement*: the server and the user must negotiate a session key for protecting subsequent communications.

(C8) *Low computation and communication cost*: due to the constrained power and the limited memory of the smart card, high computation operations should be reduced to achieve bandwidth demands.

(C9) *The user's ability to revoke the smart card rather than the user's identity*: even if the user losses her or his smart card, her or his identity can be unchanged.

(C10) *The smart card loss protection*: the scheme can protect the lost smart card from impersonation or guessing attacks.

(C11) *The smart card's possibility to be used in a multipurpose environment*: the smart card can be used to log in to many servers that provide a variety of services.

After a thorough analysis of Wang et al.'s scheme [14], we found some security issues; that is, (1) a malicious attacker can easily impersonate the legitimate server to deceive the user, but the user cannot be conscious of this attack. So, the fooled user may submit his privacy information to an attacker and (2) the scheme cannot achieve the multipurpose smart card requirement because it only has single-server authentication. In this paper, we propose a novel approach for solving these problems and improving the security strength. Furthermore, our scheme can be applied to the multipurpose, smart card environment; that is, the smart card can be authenticated by multipurpose servers. In addition, our scheme ensures computation efficiency, so it can be easily implemented in ubiquitous computing environments.

The rest of this paper is organized as follows. In Section 2, we review Wang et al.'s user authentication scheme and demonstrate the security drawback. Then, in Section 3, we present our scheme, that is, the multipurpose, smart card authenticated key agreement scheme, followed by the security and efficiency analyses shown in Section 4. Finally, concluding remarks are presented in Section 5.

2. Review of Wang et al.'s Scheme

In this section, we briefly review Wang et al.'s authentication and key agreement scheme [14] and demonstrate that their scheme cannot satisfy mutual authentication (C6) against the impersonation attack. Notations used throughout this paper are described in Section 2.1. The details and the drawbacks of Wang et al.'s scheme are demonstrated in Sections 2.2 and 2.3, respectively.

2.1. Notations

U: the set of users, $U = \{U_1, U_2, \ldots, U_n\}$,

S: the set of registered servers, $S = \{S_1, S_2, \ldots, S_n\}$,

RC: the registration center,

x: the server's master key, the length of which is sufficient to resist the brute force attack,

UID: the identity of the user,

CID: the identity of the smart card,

SID: the identity of the server,

PW: the password of the user,

$h()$: a secure one-way hash function [16, 17] with an l-bit output,

N: a nonce value,

n, p: two large primes,

E_p: an elliptic curve equation over p of the server,

G: a generator point of E_p with a large order n,

a, b: two integer elements,

Q_i: a large prime generated by U_i, where $Q_i > 2^l$,

R, W: two random numbers,

SK: the session key,

$\oplus$: the exclusive-or operation done for two-bit strings,

$\parallel$: the string concatenation operator,

$E_k(m)$: the ciphertext of m, which is the product of m encrypted using the key k in the secure symmetric cryptosystem [18, 19],

$D_k(c)$: the plaintext of c, which is the product of c decrypted using the key k in the secure symmetric cryptosystem [18, 19],

CRL: the smart card revocation list.

2.2. Review of Wang et al.'s Scheme. In this subsection, we briefly review and discuss Wang et al.'s scheme [14]. There are two participants involved, that is, the user and the server. Let UID, CID, and SID be the unique identification of the user, server, and smart card, respectively.

Wang et al.'s scheme comprises several phases, that is, registration phase, authentication phase, password changing phase, revoking smart card phase, user eviction phase, and user anonymity phase, but we only discuss the first two phases. The other phases of their scheme basically conform to the above-mentioned security requirements.

Before the scheme starts, it must set some system parameters, which must satisfy the elliptic curve cryptosystem requirements [20], for example, $p > 2^{160}$, $4a^3 + 27b^2$, and mod $p \neq 0$. We assume that all system parameters conform to the security requirements.

Registration Phase. In this phase, all messages are delivered in a secure channel, since the smart card cannot be transmitted in the network. When a new user U wants to access a server's services, he/she must first submit his/her identity (UID) to the server for registration. If the server accepts the application, it then takes the following steps.

Step 1. The server computes a parameter $B = h(x \parallel \text{UID} \parallel \text{CID}) \times G$.

Step 2. The server stores (UID, B, G, E_p) in the smart card and issues it to U.

Step 3. The server maintains the (UID, CID) table.

Step 4. After receiving the smart card, U inputs her or his password (PW) into the smart card. The smart card computes $B' = B \oplus h(\text{PW})$. Then it replaces B with B' in the smart card. As a result, the smart card stores (UID, B', G, E_p):

$$\begin{array}{lcl}
\text{The user} & & \text{The sever} \\
T_1 = R \times G, & \xrightarrow{\text{UID},T_1,T_2} & T_2' = T_1 \times h(x \parallel \text{ID} \parallel \text{CID}), \\
B = B' \oplus h(\text{PW}), & & \text{checks } T_2' \text{ with } T_2. \\
T_2 = h(R \times B). & & K = h(W \times T_1),\ V_1 = h(T_2 \parallel K), \\
 & \xleftarrow{T_3,V_1} & T_3 = W \times G. \\
K' = h(R \times T_3), & & \\
V_1' = h(R \times B \parallel K'). & & \\
\text{checks } V_1' \text{ with } V_1, & & \\
V_2 = h(R \times B \parallel K' + 1). & & \text{checks } h(T_2' \parallel K + 1) \text{ with } V_2. \\
\text{session key } K'. & \xrightarrow{V_2} & \text{session key } K.
\end{array} \tag{1}$$

Authentication Phase. We illustrate this phase in (1) and explain the details as follows. When U wants to log in to the server, he/she inserts the smart card into the card reader and inputs his/her password PW into the device. The user U performs the following steps.

Step 1. U computes $T_1 = R \times G$, $B = B' \oplus h(\text{PW})$, and $T_2 = R \times B$.

Step 2. U delivers (UID, T_1, T_2) to the server.

The server receives the above message and then executes the steps as follows.

Step 3. The server computes $T_2' = T_1 \times h(x \parallel \text{UID} \parallel \text{CID})$.

Step 4. The server checks T_2' with T_2. If they are equal, then the user's identity can be sure. Otherwise, S terminates this procedure.

Step 5. The server calculates $K = h(W \times T_1)$, $V_1 = h(T_2 \parallel K)$, and $T_3 = W \times G$.

Step 6. The server returns (T_3, V_1) to the user.

After receiving (T_3, V_1), U enforces the steps to validate the server's identity and generate a session key as follows.

Step 7. U computes $K' = (R \times T_3)$ and $V_1' = h(R \times B \parallel K')$.

Step 8. U checks V_1' with V_1. If they are equal, then the server's identity is valid. Otherwise, U terminates this procedure.

Step 9. U computes $V_2 = h(R \times B \parallel K' + 1)$ and transfers V_2 to the server.

Step 10. If V_2 passes the validation with $h(T_2' \parallel K' + 1)$, then the server and U can obtain a session key K. Otherwise, the server will give up on this authentication.

2.3. Drawbacks of the Reviewed Scheme. After analyzing the above protocol, we can easily derive the session key $K = K' = h(R \times W \times G)$ to keep data secrecy in further communications. However, we find that it still has two drawbacks. First, their scheme cannot be applied in the smart card multipurpose requirements because it is only designed for a single-server authentication environment. In addition, it has a security flaw. The malicious attacker can impersonate a legal server to cheat the user. Hence, it cannot satisfy the mutual authentication requirement. We show how the attacker can impersonate a legitimate server in the authentication phase as follows.

Assume that a malicious attacker Mary can intercept all transmitted messages between the user and the server. Then, she counterfeits a legal server to perform authentication with the user. First, the user sends (UID, T_1, T_2) to Mary. Then, Mary randomly chooses a number W_m to compute $K_m = h(W_m \times T_1)$, $V_{1m} = h(T_2 \parallel K_m)$, and $T_{3m} = W_m \times G$ as in Step 8. After that, Mary delivers (T_{3m}, V_{1m}) to U. As a result, V_{1m} can pass the validation in Step 11 because $K_m = h(W_m \times T_1) = h(W_m \times R \times G) = h(R \times T_{3m}) = K'$ and $V_{1m} = h(T_2 \parallel K_m) = h(R \times B \parallel K') = V_1'$. Finally, Mary drops the returned V_2 of Step 12, so she can securely communicate with U using the session key K_m. Therefore, Wang et al.'s scheme cannot achieve mutual authentication.

3. The Proposed Scheme

In this section, we first list the superiorities of our scheme over Wang et al.'s scheme in Section 3.1. Then, the details of our novel scheme are presented in Section 3.2.

3.1. Superiorities of Our Scheme

3.1.1. Mutual Authentication. Our protocol ensures mutual authentication between U and S without a password table.

3.1.2. Multipurpose Smart Cards. The smart card can satisfy the multipurpose requirement. The smart card can be used to access multiple servers on the user's demand.

3.1.3. Efficiency and Practicability. The user can dynamically choose or remove services, as he or she chooses. The user's changing of her or his demands will not affect any service server. In addition, the transmission rounds and computation load are simplified in the authentication phase. Therefore, our scheme can be easily implemented for ubiquitous environments.

3.2. Our Proposed Scheme. In our scheme, the user can use the smart card to dynamically access many kinds of services. Therefore, the registration center RC is a necessary participant to manage adding or removing the services of the users.

The proposed scheme consists of five phases, that is, (1) the initialization phase, (2) the registration phase, (3) the authentication phase, (4) the demands-changing phase, and (5) the card-revoking phase. Note that x is the RC's secret key in our scheme. The details are shown as follows.

Initialization Phase

Step 1. If the server S_j wants to join this service group, it must submit its identity SID_j and its secret prime number P_j to RC for registration.

Step 2. RC stores (SID_j, P_j) and sends $w_j = h(x, \mathrm{SID}_j)$ to S_j through a secure channel.

Registration Phase

Step 1. U_i arbitrarily chooses a large prime $Q_i > 2^l$ and sends (UID_i, Q_i) to RC for registration and asks a set S_d of services, where $S_d \subseteq S$.

Step 2. RC performs the following processes:

(2.1) RC computes all $h(\mathrm{UID}_i \parallel P_d)$'s, where $d \in S_d$.

(2.2) RC expands the length of each $h(\mathrm{UID}_i \parallel P_d)$ to be $l+1$ by setting the most significant bit to be 1.

(2.3) RC calculates $A_i = Q_i \prod_{d \in S_d} h(\mathrm{UID}_i \parallel P_d)$ and $B_i = h(x \parallel A_i \parallel \mathrm{UID}_i \parallel \mathrm{CID})$.

Step 3. RC stores $(\mathrm{UID}_i, A_i, B_i)$ in the smart card. Then, RC issues this smart card to U_i.

Step 4. After receiving the smart card, U_i inputs her or his password PW_i into the smart card. The smart card computes $A_i' = A_i \oplus h(\mathrm{PW}_i)$. Then it replaces A_i with A_i' in the smart card. As a result, the smart card stores $(\mathrm{UID}_i, A_i', B_i)$.

Authentication Phase. We illustrate this phase in (2) and explain the details as follows. When U_i wants to log in to S_j, where $S_j \in S_d$, he/she inserts the smart card into the card reader and inputs his/her password PW_i into the device. The user U_i performs the steps as follows.

Step 1. The smart card computes $A_i = A_i' \oplus h(\mathrm{PW}_i)$ and generates a random key K, where $2^{l-1} < K < 2^l$.

Step 2. The smart card calculates $T_1 = A_i + K$ and $T_2 = E_K(\mathrm{UID}_i \parallel \mathrm{SID}_j \parallel \mathrm{CID} \parallel N_1)$, where N_1 is a nonce value.

Step 3. U_i delivers $(\mathrm{UID}_i, T_1, T_2)$ to the server S_j.

S_j receives the above message and then executes the steps as follows.

Step 4. S_j computes $K' = T_1 \bmod h(\mathrm{UID}_i \parallel P_{S_j})$ and $D_{K'}(T_2)$ to obtain UID'_i, SID'_j, CID', and N'_1.

Step 5. S_j checks the UID'_i, SID'_j, and CID'. If UID'_i and SID'_j pass the validation and CID' does not belong to CRL, then the user's identity can be sure. Otherwise, S_j terminates this procedure.

Step 6. S_j calculates $T_3 = E_{K'}(\mathrm{SID}_j \parallel N'_1 + 1)$.

Step 7. S_j returns (SID_j, T_3) to U_i.

After receiving (SID_j, T_3), U_i executes the steps to validate the server's identity as follows.

Step 8. U_i computes $D_K(T_3)$ to obtain SID'_j and $N'_1 + 1$.

Step 9. U_i checks SID'_j and $N'_1 + 1$ with the received SID_j and $N_1 + 1$. If they are valid, then the server's identity can be sure, and the session key $\mathrm{SK} = K$. Otherwise, U_i terminates this procedure:

$$\begin{array}{lcl}
\text{The user} & & \text{The sever} \\
A_i = A'_i \oplus h(\mathrm{PW}_i), & & \\
T_1 = A_i + K, & \xrightarrow{\mathrm{UID}_i, T_1, T_2} & K = T_1 \bmod h\left(\mathrm{UID}_i \parallel P_{S_j}\right), \\
T_2 = E_K\left(\mathrm{UID}_i \parallel \mathrm{SID}_j \parallel \mathrm{CID} \parallel N_1\right). & & D_K(T_2) = \left(\mathrm{UID}'_i \parallel \mathrm{SID}'_j \parallel \mathrm{CID}' \parallel N'_1\right), \\
 & & \text{checks } \mathrm{SID}'_j, \mathrm{UID}'_i \text{ and } \mathrm{CID}', \\
 & \xleftarrow{\mathrm{SID}_j, T_3} & T_3 = E_K\left(\mathrm{SID}_j \parallel N'_1 + 1\right), \\
D_K(T_3) = \left(\mathrm{SID}' \parallel N'_1 + 1\right) & & \mathrm{SK} = K. \\
\text{checks } \mathrm{SID}' \text{ and } N'_1 + 1, & & \\
\mathrm{SK} = K. & &
\end{array} \tag{2}$$

Demands-Changing Phase. When the user U_i changes her mind, she wants to increase or remove some services. She must perform the registration phase again. She chooses a new services combination set $U_n \subseteq U$. Then RC and U_i perform Steps 2 through 4. Afterwards, RC gets a new set $(\mathrm{UID}_i, A_i, B_i)$, and the smart card stores a new $(\mathrm{UID}_i, A'_i, B_i)$. Other participants will not be affected by these changes.

Card-Revoking Phase. When the user U_i loses his smart card, he must apply to RC for a new one. RC will record the lost card's CID into CRL and publish the CRL to all registered servers. Then, RC will perform the same steps in the registration phase to issue a new smart card to the user.

4. Security and Efficiency Analyses

In this section, we discuss several significant attacks and analyze the efficiency of our scheme. The security analyses are shown in Section 4.1. Then, we demonstrate that the proposed scheme can achieve the computation and communication efficiency listed in Section 4.2.

4.1. Security Analyses

4.1.1. Choosing the Session Key. Because the session key K is a modular, it must be less than all $h(\mathrm{UID}_i \parallel P_{S_j})$'s of A_i. Otherwise, the server will not derive the correct session key K. However, for security reasons, we expect the K value to be as large as possible. To achieve these two requirements, the session key K must satisfy $2^{l-1} < K < \min\{Q_i, H(\mathrm{UID}_i \parallel P_d) \mid d \in S_d\} \leq 2^l$. Otherwise, there is a possibility that an incorrect number K will be derived in the server. To ensure that the above equation holds, we expand the length of each $h(\mathrm{UID}_i \parallel P_d)$ to be $l + 1$ and set the most significant bit as 1. Meanwhile, the system must check whether $2^{l-1} - 1 \leq K \leq 2^l - 1$. Therefore, the availability of our scheme can be sure.

4.1.2. Session Key Security. If an attacker collects many T_1's and tries to derive the next session K, it will be impossible. Due to the process of generating the session key in the authentication phase, each session key is independent and different.

4.1.3. The Server's Secrecy Protection. Although the user knows $A_i = Q_i \prod_{d \in S_d} h(\mathrm{UID}_i \parallel P_d)$, the server's secrecy P_d can still be protected. The user cannot compute any P_d, since $h()$ is a secure one-way hash function [16, 17]. In addition, each $h(\mathrm{UID}_i \parallel P_d)$ may not be a prime, so it can resist the collusion attacks of several legitimate subscribers. The malicious user will get nothing to calculate A_i/A_z because both A_i and A_z are two products of many respective different factors. It is hard to find any common divisor among them, since $h(\mathrm{UID}_i \parallel P_j)$ and $h(\mathrm{UID}_z \parallel P_j)$ are different.

4.1.4. Impersonating Attacks. No adversary can impersonate the eligible user in our scheme. When the adversary tries to impersonate the eligible user, he/she uses the fake message $(\mathrm{UID}_i, T_1', T_2')$ to log in to the server and will get stuck in the authentication process. Since he/she does not know A' and B' of U_i, he/she cannot compute T_1' and T_2'.

On the other hand, if the attacker impersonates the service server, the user will detect that someone is trying to impersonate the server in Step 9 of authentication phase. This is because the adversary cannot compute K without the true P_{S_j}. As a result, he/she cannot respond with the correct messages $N_1' + 1$ and T_3 to the user.

Even if a legal subscriber z wants to impersonate a legal subscriber i, it is still very difficult because the user z cannot derive the $h(\mathrm{UID}_i \parallel P_{S_j})$ from T_1. Hence, no one can impersonate the eligible user or the service server in our scheme.

4.1.5. Reply Attacks. Both S_j and U_i must check nonce N_1; meanwhile, they are protected by the secure key K encryption, since the attacker cannot change it arbitrarily. This way, we can eliminate the possibility of a replay attack.

4.1.6. Password-Guessing Attacks. If a malicious attacker tries to guess the password of a lost smart card, he will fail. The password is stored neither in the smart card nor on the server's disk. His incorrect guesses of the password will be rejected in Step 5 of the authentication phase, since the incorrect A_i is used.

4.1.7. Known-Key Attacks. Each session key is different from all others, since the session key K is randomly generated during each iteration. Hence, our scheme can achieve forward secrecy and backward secrecy.

4.1.8. Smart Card Loss Attacks. If any user loses his smart card, he can apply for a new one and revoke the lost smart card in the card-revoking phase. If an attacker deploys a lost smart card to log in to the server, it will fail because the server will check CID in Step 5 of authentication phase. Therefore, our scheme can satisfy (C9) and (C10) of the aforementioned criteria.

4.2. Efficiency Analyses

Property 1 (the scheme needs no password and encrypted key table). Since the server and the user can compute K in the authentication phase without the help of the encrypted key table or the password table, the challenge-response interactive authentication can be ensured.

Property 2 (the scheme provides mutual authentication without RC's support). As shown in our scheme, when the new server and the user join this system, RC does not need to transmit any message to each user and the server. Since the smart card and S_j compute the session key, RC is not involved. On the other hand, RC only takes charge of the registration of new users or new servers. Hence, our proposed scheme can reduce RC's overhead.

Property 3 (the scheme provides higher security and computation efficiency). Wang et al.'s scheme is based on the difficulty of solving the elliptic curve discrete logarithm problem with a 160-bit key; the security is quite solid, for now. However, our scheme deploys a symmetric cryptosystem and key length with at least 128 bits. According to Table 1 [14], our scheme will provide higher security than ECC-160 bits and provide greater computation efficiency because it can estimate an account of a symmetric key encryption (DES or AES functions) 1000 times faster than the asymmetric key encryption (ECC) speed, according to Schneier's book [20]. Therefore, our scheme fits for low computation devices and ubiquitous environments.

Table 1: Equivalent key sizes in bits [14].

Symmetric	ECC	RSA	Year to attack in MIPS	Security lifetime
80	160	1024	10^{12}	Until 2010
112	224	2048	10^{24}	Until 2030
128	256	3072	10^{28}	Beyond 2031

Property 4 (the scheme provides both communication and round efficiencies). It is assumed that both the output size of the secure one-way hashing function [16] and the block size of the secure symmetric cryptosystems are 160 bits. We list the comparisons of communication cost between our scheme and the related schemes in Table 2. Obviously, our scheme's communication efficiency is better than Wang et al.'s scheme [14]. Moreover, both of Wang et al.'s schemes [14, 15] are insecure. In addition, our scheme only needs two-round interactions to complete authentication and key agreement negotiation. That is the smallest number of rounds in any of the related schemes.

Property 5 (the scheme is practical). In Table 3, comparisons of the criteria between our scheme and the related schemes are shown.

According to Table 3, our scheme proposes a solution to enhance the security drawback of Wang et al.'s scheme, and it also satisfies the multipurpose smart card requirement. Moreover, the numbers of different kinds of computation operations required by our scheme are smaller than those required by Wang et al.'s scheme [14], so the computation load of our scheme is lighter than the others. In addition, among aforementioned schemes, ours is the only one that can be used in the distributed authentication architecture. It is obvious that our proposed scheme is superior to both of Wang et al.'s schemes [14, 15] in terms of both round efficiency and computation efficiency.

5. Conclusions

In this paper, we have proposed a multipurpose key agreement scheme using smart cards. The proposed scheme enhances Wang et al.'s scheme. Moreover, it provides better functionality and efficiency. According to the analyses in

Table 2: Comparisons of communication cost.

	Our scheme	Wang et al.'s scheme [14]	Wang et al.'s scheme [15]
Communication cost	161 + 160 × 3 = 641 bits	160 × 2 + 224 × 2 + 64 = 832 bits	64 + 160 × 2 + 32 = 476 bits

Table 3: Criteria comparisons between our scheme and the related schemes.

	Our scheme	Wang et al.'s scheme [14]	Wang et al.'s scheme [15]
(C1)	Yes	Yes	Yes
(C2)	Yes	Yes	Yes
(C3)	Yes	Yes	Yes
(C4)	Yes	Yes	No
(C5)	Yes	Yes	No
(C6)	Yes	No	No
(C7)	Yes	Yes	Yes
(C8)	Yes	Yes	Yes
(C9)	Yes	Yes	No
(C10)	Yes	Yes	No
(C11)	Yes	No	No

the above section, our scheme can be practically used in ubiquitous computing environments.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. Ye, S. Dasiopoulou, G. Stevenson et al., "Semantic web technologies in pervasive computing: a survey and research roadmap," *Pervasive and Mobile Computing*, vol. 23, pp. 1–25, 2015.

[2] N. Wang, N. Zhang, and T. Aaron Gulliver, "Cooperative key agreement for wireless networking: key rates and practical protocol design," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 2, pp. 272–284, 2014.

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

[4] C.-C. Chang and S.-J. Hwang, "Using smart cards to authenticate remote passwords," *Computers and Mathematics with Applications*, vol. 26, no. 7, pp. 19–27, 1993.

[5] H.-Y. Chien, J.-K. Jan, and Y.-M. Tseng, "An efficient and practical solution to remote authentication: smart card," *Computers and Security*, vol. 21, no. 4, pp. 372–375, 2002.

[6] M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 629–631, 2004.

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

[8] M.-S. Hwang and L.-H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28–30, 2000.

[9] I.-C. Lin, H.-H. Ou, and M.-S. Hwang, "Efficient access control and key management schemes for mobile agents," *Computer Standards and Interfaces*, vol. 26, no. 5, pp. 423–433, 2004.

[10] H.-M. Sun, B.-Z. He, C.-M. Chen, T.-Y. Wu, C.-H. Lin, and H. Wang, "A provable authenticated group key agreement protocol for mobile environment," *Information Sciences*, vol. 321, pp. 224–237, 2015.

[11] N. Wang, X. Song, J. Cheng, and V. C. M. Leung, "Enhancing the security of free-space optical communications with secret sharing and key agreement," *Journal of Optical Communications and Networking*, vol. 6, no. 12, pp. 1072–1081, 2014.

[12] Y.-Y. Wang, J.-Y. Liu, F.-X. Xiao, and J. Dan, "A more efficient and secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 32, no. 4, pp. 583–585, 2009.

[13] M. K. Khan, S.-K. Kim, and K. Alghathbar, "Cryptanalysis and security enhancement of a more efficient & secure dynamic ID-based remote user authentication scheme," *Computer Communications*, vol. 34, no. 3, pp. 305–309, 2011.

[14] R.-C. Wang, W.-S. Juang, and C.-L. Lei, "Robust authentication and key agreement scheme preserving the privacy of secret key," *Computer Communications*, vol. 34, no. 3, pp. 274–280, 2011.

[15] R. C. Wang, W. S. Juang, and C. L. Lei, "A simple and efficient key exchange scheme against the smart card loss problem," in *Emerging Directions in Embedded and Ubiquitous Computing*, vol. 4809 of *Lecture Notes in Computer Science*, pp. 728–744, 2007.

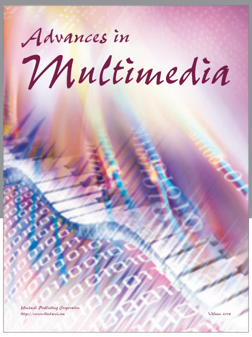
[16] NIST, "Secure hash standard," NIST FIPS PUB 180-1, National Institute of Standards and Technology, 1995, http://www.itl.nist.gov/fipspubs/fip180-1.htm.

[17] R. Rivest, "The MD5 message-digest algorithm," RFC 1321, Internet Activities Board, Internet Privacy Task Force, 1992.

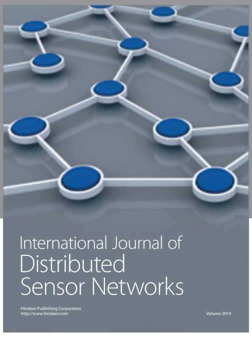
[18] National Bureau of Standards, *NBA FIPS PUB 46-1, Data Encryption Standard*, US Department of Commerce, National Bureau of Standards, 1988.

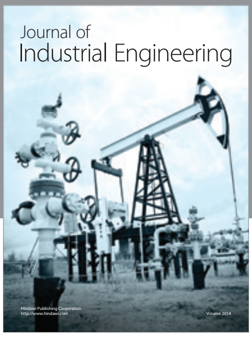
[19] NIST FIPS PUB, "Advanced Data Encryption Standard, National Institute of Standards and Technology," 2001, http://www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

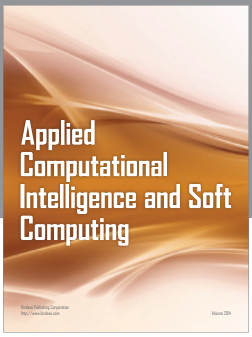
[20] B. Schneier, *Applied Cryptography*, Wiley, New York, NY, USA, 2nd edition, 1996.

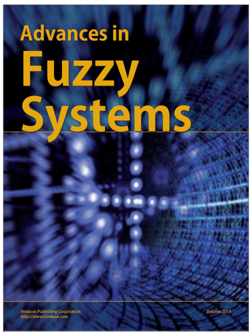

Advances in
Multimedia

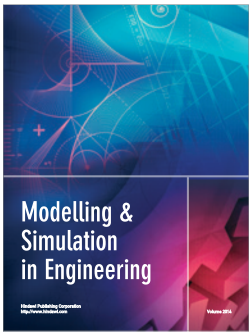
The Scientific World Journal

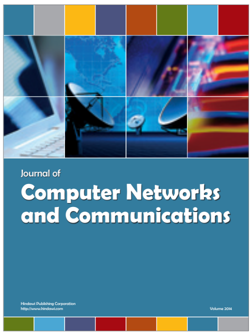

International Journal of
Distributed Sensor Networks

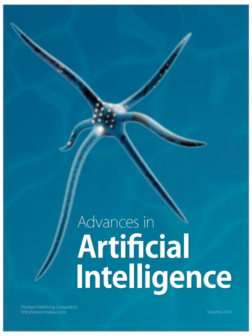

Journal of
Industrial Engineering

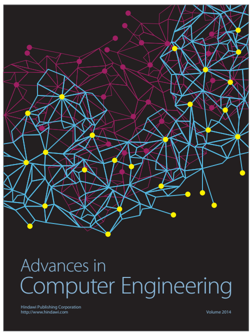

Applied Computational Intelligence and Soft Computing


Advances in
Fuzzy Systems

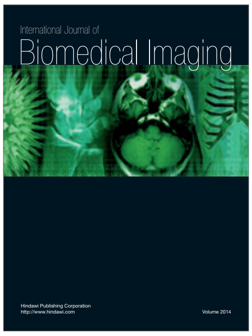

Modelling & Simulation in Engineering

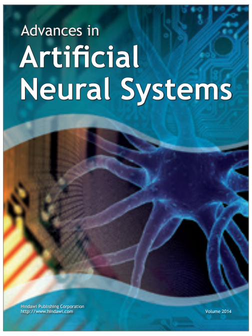

Journal of
Computer Networks and Communications

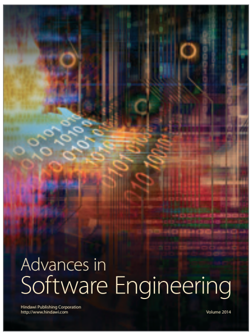

Advances in
Artificial Intelligence


Advances in
Computer Engineering

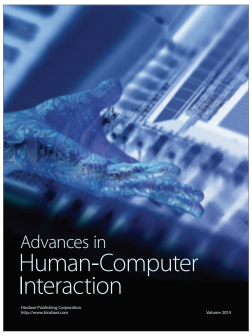
International Journal of
Computer Games Technology

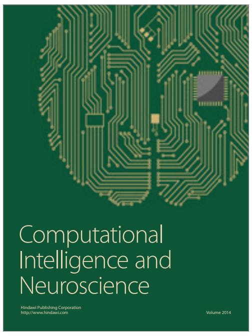

International Journal of
Biomedical Imaging


Advances in
Artificial Neural Systems


Advances in
Software Engineering

Journal of
Robotics


Advances in
Human-Computer Interaction


Computational Intelligence and Neuroscience

International Journal of
Reconfigurable Computing

Journal of
Electrical and Computer Engineering