Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Analysis and design of a smart card based authentication protocol*

Kuo-Hui YEH†1, Kuo-Yu TSAI†‡2, Jia-Li HOU1

(1*Department of Information Management, National Dong Hwa University, Taiwan 974, Hualien*)

(2*Department of Management Information Systems, Hwa Hsia Institute of Technology, Taiwan 235, New Taipei City*)

†E-mail: khyeh@mail.ndhu.edu.tw; kytsai@cc.hwh.edu.tw

Received June 4, 2013; Revision accepted Sept. 22, 2013; Crosschecked Nov. 18, 2013

Abstract: Numerous smart card based authentication protocols have been proposed to provide strong system security and robust individual privacy for communication between parties these days. Nevertheless, most of them do not provide formal analysis proof, and the security robustness is doubtful. Chang and Cheng (2011) proposed an efficient remote authentication protocol with smart cards and claimed that their proposed protocol could support secure communication in a multi-server environment. Unfortunately, there are opportunities for security enhancement in current schemes. In this paper, we identify the major weakness, i.e., session key disclosure, of a recently published protocol. We consequently propose a novel authentication scheme for a multi-server environment and give formal analysis proofs for security guarantees.

Key words: Authentication, Privacy, Security, Smart card
doi:10.1631/jzus.C1300158 **Document code:** A **CLC number:** TP309

1 Introduction

Recently, various remote user authentication protocols (Lin *et al*., 2003; Chang and Lee, 2004; Juang, 2004; Liaw *et al*., 2006; Lee *et al*., 2008; Chang and Tsai, 2010; Chang and Cheng, 2011; Chen *et al*., 2011) have been proposed to authenticate valid users before they can access remote services. That will prevent remote services from being accessed by unauthorized users. Some of the proposed protocols (Liaw *et al*., 2006; Chang and Tsai, 2010; Chen *et al*., 2011) are just for the single-server environment. Consider the scenario in which a user intends to remotely access several services provided by different servers. The user must register with those service servers, and then he/she may obtain various user identities and the corresponding passwords. For accessing different remote services, the user must prove to the servers that he/she is a legal user by repetitively inputting his/her identities and passwords. It is very inconvenient for a user. Hence, it is crucial to provide secure authentication compatible with the multi-server environment.

Based on the Euclidean plane, Lin *et al*. (2003) proposed a remote authentication protocol for multi-server architecture. In their proposed protocol, a user can access services from different servers with his/her smart card. However, Juang (2004) mentioned that Lin *et al*. (2003)'s protocol is inefficient because the user's smart card has to store a large number of public system parameters for each service server. Juang (2004) further proposed an efficient password authenticated key agreement scheme using smart cards for a multi-server environment. In the same year, Chang and Lee (2004) showed that Juang's scheme is still inefficient and presented a password authentication scheme for a multi-server environment. Change and Lee (2004)'s scheme not only achieves system efficiency but also preserves security robustness. Lee *et al*. (2008) mentioned that the related authentication schemes with using smart cards are not sufficiently suitable for the real world since tamper-resistant readers are not always available everywhere.

‡ Corresponding author

* Project (Nos. 102-2218-E-259-004, 102-2218-E-146-002, and 102-2218-E-011-012) supported by Taiwan Information Security Center (TWISC) and National Science Council, Taiwan

Furthermore, Lee *et al*. (2008) proposed a password-based authentication protocol for a multi-server environment without using smart cards. In Lee *et al*. (2008)'s protocol, the password with low entropy is too weak to be implemented as a protection module for remote authentication. Recently, Chang and Cheng (2011) and Chen *et al*. (2011) proposed two remote authentication protocols with a smart card. Chen *et al*. (2011) proposed a smart card based mobile medical treatment examination report transaction system for providing a secure transaction model. Chang and Cheng (2011) presented a smart card based remote authentication mechanism for a multi-server. Chang and Cheng (2011) claimed that security could be more convincing after one major security problem shown in Section 2.2 is solved. Some crucial security requirements for a remote authentication mechanism are as follows:

1. Mutual authentication: A user and a server can authenticate with each other.

2. Session key agreement: A user and a server can share a session key used for securing their consequent communications.

3. Robustness: An authentication mechanism must resist the well-known attacks, including the replay attack, the server spoofing attack, and the user impersonation attack. In addition, important security properties, such as session key security, forward secrecy, and known-key security, must be satisfied in an authentication mechanism.

4. No timestamp: Using a timestamp may give rise to a time synchronization problem. Hence, it is suggested to design an authentication mechanism without using any timestamp.

5. Freely chosen password: A user can freely choose his/her password by himself/herself.

6. Single registration: A user just registers at the registration center once, and then the user can access all authorized services in eligible servers.

7. Low communication and computation cost: In designing remote authentication mechanisms, both a low communication overhead and a low computation complexity must be satisfied.

2 Chang and Cheng's authentication scheme

In this section, we will investigate the security of Chang and Cheng (2011)'s authentication scheme. We first review Chang and Cheng's scheme in Section 2.1 and then give the security analysis in Section 2.2.

2.1 Review

Chang and Cheng's scheme consists of four phases: registration, login, authentication and key agreement, and password modification. The authors assume that a registration center RC is trustworthy and each authorized service provider SP_j shares a secret key $KRS_j=H(SID_j||k)$ with RC, where k is a private key chosen by RC, SID_j is SP_j's identifier, and $H()$ is the private one-way hash function known only by RC. Note that there is another public one-way hash function $h()$ in Chang and Cheng's scheme.

Registration phase: A user U_i freely chooses and submits his/her identifier id_i and password pw_i to RC via a secure channel. If RC accepts U_i's registration request, RC computes $TID_i=T_i||id_i$ and stores it in the backend database, where T_i is U_i's registration time. After that, RC calculates $TPW_i=h(pw_i)$ and $\sigma_i=H(TID_i||k)\oplus pw_i$ and stores TID_i, $h()$, TPW_i, and σ_i in the smart card.

Login phase (Fig. 1): While U_i intends to access the service provider SP_j, he/she first inserts his/her smart card into the card reader and then keys in the password pw_i^*. First, the smart card computes $h(pw_i^*)$ and compares the result with the stored TPW_i. If it holds, the smart card calculates $\alpha=h(\sigma_i\oplus pw_i^*\oplus N_U\oplus SID_j)$, where N_U is a nonce. Next, the smart card transmits $\{TID_i, \alpha, N_U\}$ to SP_j. Once SP_j receives this request, SP_j computes $\beta=h(KRS_j\oplus N_S)$ and sends $\{TID_i, \alpha, N_U, SID_j, \beta, N_S\}$ to RC, where N_S is a nonce generated by SP_j.

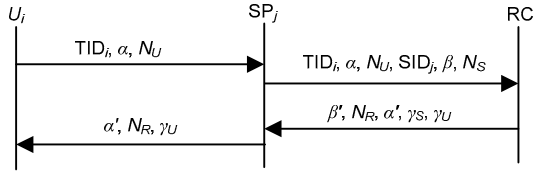


Fig. 1 The login phase and authentication and key agreement phase

Authentication and key agreement phase (Fig. 1): Upon obtaining the login message, RC verifies the validity of N_U, N_S, and TID_i. Then RC verifies the following two equations:

$$h(H(TID_i||k)\oplus N_U\oplus SID_j)=\alpha,$$
$$h(H(SID_j||k)\oplus N_S)=\beta.$$

If all verifications hold, RC subsequently chooses a random number ran and calculates $\beta'=h(H(\mathrm{SID}_j||k)\oplus N_R)$, $\alpha'=h(H(\mathrm{TID}_i||k)\oplus N_R)$, $\gamma_S=h(H(\mathrm{SID}_j||k))\oplus(\mathrm{ran}||h(H(\mathrm{TID}_i||k)))$, and $\gamma_U=(h(H(\mathrm{SID}_j||k))||\mathrm{ran})\oplus h(H(\mathrm{TID}_i||k))$, where N_R is a nonce generated by RC. Finally, RC responds $\{\beta', N_R, \alpha', \gamma_S, \gamma_U\}$ to SP_j. Upon receiving the response messages, SP_j verifies N_R and $h(\mathrm{KRS}_j\oplus N_R)=\beta'$. After that, SP_j performs the following computations: $s=\gamma_S\oplus h(\mathrm{KRS}_j)$ and $\mathrm{SK}=h((h(\mathrm{KRS}_j)||s)\oplus N_U\oplus N_S\oplus N_R)$, where SK is the session key.

Next, SP_j responds $\{\alpha', N_R, \gamma_U\}$ to U_i. When U_i gets the message, the smart card will verify the freshness for N_R and the correctness of $h(\sigma_i\oplus \mathrm{pw}_i^*\oplus N_R)=\alpha'$. If all verifications are successful, U_i calculates $u=\gamma_U\oplus h(\sigma_i\oplus \mathrm{pw}_i^*)$ and $\mathrm{SK}=h((u||h(\sigma_i\oplus \mathrm{pw}_i^*))\oplus N_U\oplus N_S\oplus N_R)$.

Password modification phase: If U_i wants to change the password, he/she needs to insert his/her smart card into the device and enter his/her old password pw_i^*. Once the examination of $h(\mathrm{pw}_i^*)=\mathrm{TPW}_i$ holds, U_i can choose a new password $\mathrm{pw}_i^{\mathrm{new}}$, and the smart card updates $\mathrm{TPW}_i^{\mathrm{new}}=h(\mathrm{pw}_i^{\mathrm{new}})$ and $\sigma_i^{\mathrm{new}}=\sigma_i\oplus \mathrm{pw}_i^*\oplus \mathrm{pw}_i^{\mathrm{new}}$.

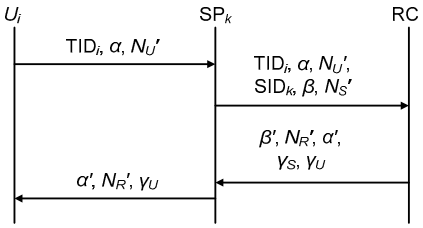
2.2 Security analysis

Chang and Cheng (2011)'s authentication scheme could be better after the major security vulnerability, i.e., session key disclosure, is solved. Once the session key is disclosed, the user impersonation attack, the server counterfeit attack, and the man-in-the-middle attack are all inevitable.

Scenario I (A legal but malicious service provider SP_j) In the authentication and key agreement phase, RC responds $\{\beta', N_R, \alpha', \gamma_S, \gamma_U\}$ to SP_j, and then SP_j verifies N_R and $h(\mathrm{KRS}_j\oplus N_R)=\beta'$ and retrieves $s=\gamma_S\oplus h(\mathrm{KRS}_j)$ to compute $\mathrm{SK}=h((h(\mathrm{KRS}_j)||s)\oplus N_U\oplus N_S\oplus N_R)$. From the verification for $s=\gamma_S\oplus h(\mathrm{KRS}_j)$, we can see that the value $(\mathrm{ran}||h(H(\mathrm{TID}_i||k)))$ can easily be derived, where $\mathrm{KRS}_j=H(\mathrm{SID}_j||k)$.

$$
\begin{aligned}
s&=\gamma_S\oplus h(\mathrm{KRS}_j)\\
&=h(H(\mathrm{SID}_j||k))\oplus(\mathrm{ran}||h(H(\mathrm{TID}_i||k)))\oplus h(\mathrm{KRS}_j)\\
&=h(H(\mathrm{SID}_j||k))\oplus(\mathrm{ran}||h(H(\mathrm{TID}_i||k)))\oplus h(H(\mathrm{SID}_j||k))\\
&=(\mathrm{ran}||h(H(\mathrm{TID}_i||k))).
\end{aligned}
$$

Now SP_j possesses the value $h(H(\mathrm{TID}_i||k))$. In the future, if U_i intends to launch a new authentication session with another service provider SP_k, where $k\neq j$, the legal but malicious service provider SP_j will be able to utilize a series of attack procedures to derive the current session key SK shared between U_i and SP_k (Fig. 2).



SP_j eavesdrops message $\{\beta', N_R', \alpha', \gamma_S, \gamma_U\}$ and performs the following computations:

1. $\gamma_U\oplus h(H(\mathrm{TID}_i||k))=(h(H(\mathrm{SID}_k||k))||\mathrm{ran}')$
2. $\gamma_S\oplus(\mathrm{ran}'||h(H(\mathrm{TID}_i||k)))=h(H(\mathrm{SID}_k||k))$
3. SP_j can compute SK shared between SP_k and U_i, i.e.,

$$
\begin{aligned}
\mathrm{SK}&=h((h(\mathrm{KRS}_k)||s)\oplus N_U'\oplus N_S'\oplus N_R')\\
&=h(h(H(\mathrm{SID}_k||k))||(\mathrm{ran}'||h(H(\mathrm{TID}_i||k)))\oplus N_U'\oplus N_S'\oplus N_R')
\end{aligned}
$$

Fig. 2 Session key disclosure (Scenario I)

1. Let U_i and SP_k communicate with each other completely. Note that all messages between U_i and SP_k are as follows: TID_i, $\alpha=h(\sigma_i\oplus \mathrm{pw}_i^*\oplus N_U'\oplus \mathrm{SID}_k)$, N_U', SID_k, $\beta=h(\mathrm{KRS}_k\oplus N_S')$, N_S', $\beta'=h(H(\mathrm{SID}_k||kN_R')$, N_R', $\alpha'=h(H(\mathrm{TID}_i||k)\oplus N_R')$, $\gamma_S=h(H(\mathrm{SID}_k||k))\oplus(\mathrm{ran}'||h(H(\mathrm{TID}_i||k)))$, and $\gamma_U=(h(H(\mathrm{SID}_k||k))||\mathrm{ran}')\oplus h(H(\mathrm{TID}_i||k))$. All random numbers and nonce are newly generated at the current session, such as N_U', N_S', N_R', and ran′.

2. SP_j eavesdrops the messages $\{\beta', N_R', \alpha', \gamma_S, \gamma_U\}$ and performs the following computations:

(i) $\gamma_U\oplus h(H(\mathrm{TID}_i||k))=(h(H(\mathrm{SID}_k||k))||\mathrm{ran}')$, where $h(H(\mathrm{TID}_i||k))$ is derived at the previous session communicated with U_i.

(ii) $\gamma_S\oplus(\mathrm{ran}'||h(H(\mathrm{TID}_i||k)))=h(H(\mathrm{SID}_k||k))$, where ran′ is obtained by step 2(i).

SP_j can easily calculate the session key SK shared between SP_k and U_i, i.e., $\mathrm{SK}=h((h(\mathrm{KRS}_k)||s)\oplus N_U'\oplus N_S'\oplus N_R')=h(h(H(\mathrm{SID}_k||k))||(\mathrm{ran}'||h(H(\mathrm{TID}_i||k)))\oplus N_U'\oplus N_S'\oplus N_R')$.

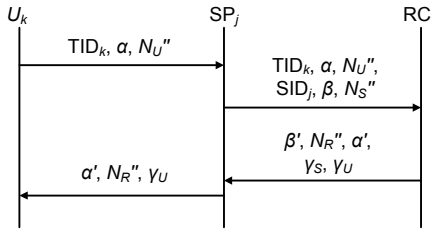
Based on the above analysis, we find that the session key can be derived by the legal but malicious SP_j, where U_i once communicated with SP_j. With this derived session key, SP_j can cheat user U_i and current service provider SP_k, respectively, or both.

Scenario II (A legal but malicious user U_i) Similarly, in the authentication and key agreement phase, SP_j responds $\{\beta', N_R, \gamma_U\}$ to U_i. The smart card will examine N_R and whether the computed value $h(\sigma_i\oplus$

$pw_i^*\oplus N_R$) equals the received value α'. If it is successfully verified, U_i calculates $u=\gamma_U\oplus h(\sigma_i\oplus pw_i^*)$ and $SK=h((u||h(\sigma_i\oplus pw_i^*))\oplus N_U\oplus N_S\oplus N_R)$. Assuming that the user U_i is legal but malicious, U_i will be able to retrieve SP_j's secret value $(h(H(SID_j||k))||ran)$ via the following equations (note that $\sigma_i=H(TID_i||k)\oplus pw_i$):

$$\begin{aligned}u&=\gamma_U\oplus h(\sigma_i\oplus pw_i^*)\\&=(h(H(SID_j||k))||ran)\oplus h(H(TID_i||k))\oplus h(\sigma_i\oplus pw_i^*)\\&=(h(H(SID_j||k))||ran)\oplus h(H(TID_i||k))\\&\quad\oplus h(H(TID_i||k)\oplus pw_i\oplus pw_i^*)\\&=(h(H(SID_j||k))||ran)\oplus h(H(TID_i||k))\oplus h(H(TID_i||k))\\&=(h(H(SID_j||k))||ran).\end{aligned}$$

Now U_i possesses the value $h(H(SID_j||k))$. Assuming that SP_j wants to launch a new authentication session with another legal user U_k, where $k\neq i$, U_i can use a series of attack procedures to derive the current session key SK shared between U_k and SP_j (Fig. 3).



U_i eavesdrops $\{\beta', N_R'', \alpha', \gamma_S, \gamma_U\}$ and performs the following computations:
1. $\gamma_S\oplus h(H(SID_j||k))=(ran''||h(H(TID_i||k)))$
2. $\gamma_U\oplus(h(H(SID_j||k))||ran'')=h(H(TID_k||k))$
3. U_i can compute SK shared between SP_j and U_k, i.e., $SK=h((h(KRS_j)||s)\oplus N_U''\oplus N_S''\oplus N_R'')=h(h(H(SID_j||k))||(ran''||h(H(TID_k||k)))\oplus N_U''\oplus N_S''\oplus N_R'')$

Fig. 3 Session key disclosure (Scenario II)

1. Let the session between U_k and SP_j be executed completely. Similarly, all messages between U_k and SP_j include TID_k, $\alpha=h(\sigma_k\oplus pw_k^*\oplus N_U''\oplus SID_j)$, N_U'', SID_j, $\beta=h(KRS_j\oplus N_S'')$, N_S'', $\beta'=h(H(SID_j||k)\oplus N_R'')$, N_R'', $\alpha'=h(H(TID_k||k)\oplus N_R'')$, $\gamma_S=h(H(SID_j||k))\oplus(ran''||h(H(TID_k||k)))$, and $\gamma_U=(h(H(SID_j||k))||ran'')\oplus h(H(TID_k||k))$. Note that all random numbers and nonce, such as N_U'', N_S'', N_R'', and ran″, are newly generated at the current session.

2. U_i eavesdrops the messages $\{\beta', N_R'', \alpha', \gamma_S, \gamma_U\}$ and the following computations are executed:

(i) $\gamma_S\oplus h(H(SID_j||k))=h(H(SID_j||k))\oplus(ran''||h(H(TID_k||k)))\oplus h(H(SID_j||k))=(ran''||h(H(TID_i||k)))$, where $h(H(TID_k||k))$ is derived from the previous session.

(ii) $\gamma_U\oplus(h(H(SID_j||k))||ran'')=h(H(TID_k||k))$, where ran″ is obtained in step 2(i).

Now U_i can derive the session key SK agreed by SP_j and U_k, i.e., $SK=h((h(KRS_j)||s)\oplus N_U''\oplus N_S''\oplus N_R'')=h(h(H(SID_j||k))||(ran''||h(H(TID_k||k)))\oplus N_U''\oplus N_S''\oplus N_R'')$.

We demonstrate that the session key can be derived by the legal but malicious U_i, where SP_j communicated with it before. Now U_i can cheat both SP_j and the current user U_k.

3 The proposed authentication protocol

We introduce a new smart card based authentication scheme for a multi-server environment. We propose two novel designs to construct a novel authentication protocol. The first design is to correlate the two messages $\{TID_i, \alpha, N_U\}$ and $\{SID_j, \beta, N_S\}$ such that the man-in-the-middle attack can be prevented during the authentication session. That is, the adversary cannot misuse a single message such as $\{TID_i, \alpha, N_U\}$ or $\{SID_j, \beta, N_S\}$ since they are strongly connected. The second is that we totally modify the values γ_S and γ_U as these two ciphers play important roles for entity authentication and session key agreement. That is, in our protocol, U_i and SP_j use γ_S and γ_U to authenticate with each other. However, the protocol will not retrieve the plain secrets regarding SP_j (or U_i) at U_i's (or SP_j's) side in each authentication session. This prevents the secret disclosure problem.

Our proposed protocol also consists of four phases: registration, login, authentication and key agreement, and password modification. We assume that RC is trustworthy, and each service provider SP_j is authorized by RC in advance and shares a secret $KRS_j=H(SID_j||k)$ with RC, where k is a secret chosen by RC, SID_j is SP_j's identifier, and $H()$ is the private one-way hash function known only by RC. In addition, RC maintains one more secret rc which is utilized to construct a valid session key at each authentication round. Similarly, we have a public one-way hash function $h()$.

Registration phase: First of all, a user U_i submits his/her identifier id_i and password pw_i to RC via a secure channel. Once RC accepts U_i's registration request, RC stores $TID_i=h(T_i||id_i)$ in the backend database, where T_i is the U_i's registration timestamp.

Then, RC computes $TPW_i=h(id_i \oplus pw_i)$ and $\sigma_i=H(TID_i||k) \oplus id_i \oplus pw_i$ and stores TID_i, $h()$, TPW_i, and σ_i in U_i's smart card.

Login phase: When U_i intends to access SP_j, he/she inserts the smart card into the card reader and keys in his/her identity id_i^* and password pw_i^*. After that, the smart card computes $h(id_i^* \oplus pw_i^*)$ and compares it with TPW_i. If it holds, the smart card calculates value α, where N_U is a nonce newly generated by U_i.

$$\begin{aligned}\alpha&=h(\sigma_i \oplus id_i^* \oplus pw_i^* \oplus N_U \oplus SID_j)\\&=h(H(TID_i||k) \oplus N_U \oplus SID_j).\end{aligned}$$

Next, U_i's smart card sends $\{TID_i, \alpha, N_U\}$ to SP_j. Once SP_j receives this request, SP_j computes β and sends $\{TID_i, \alpha, N_U, SID_j, \beta, N_S\}$ to RC, where N_S is a nonce created by SP_j.

$$\beta=h(KRS_j \oplus N_S \oplus TID_i)=h(H(SID_j||k) \oplus N_S \oplus TID_i).$$

Authentication and key agreement phase: Upon receiving $\{TID_i, \alpha, N_U, SID_j, \beta, N_S\}$, RC first verifies the freshness for N_U and N_S, and the validity of TID_i, followed by the verification of α and β.

$$\begin{aligned}\alpha&=h(H(TID_i||k) \oplus N_U \oplus SID_j),\\\beta&=h(H(SID_j||k) \oplus N_S \oplus TID_i).\end{aligned}$$

If all the verifications hold, RC generates a random number ran and calculates values β', α', γ_S, and γ_U, where N_R is a nonce generated by RC.

$$\begin{aligned}\beta'&=h(H(SID_j||k) \oplus N_R),\ \alpha'=h(H(TID_i||k) \oplus N_R),\\\gamma_S&=h(H(SID_j||k)||N_R) \oplus [h(H(rc||ran))\\&\quad ||h(H(TID_i||k)||N_R)],\\\gamma_U&=[h(H(SID_j||k)||N_R)||h(H(rc||ran))]\\&\quad \oplus h(H(TID_i||k)||N_R).\end{aligned}$$

Then RC responds $\{\alpha', \beta', N_R, \gamma_S, \gamma_U\}$ to SP_j which verifies the freshness for N_R and the validity of β'. After that, SP_j performs the following computations and computes the session key SK:

$$\begin{aligned}s&=\gamma_S \oplus h(KRS_j||N_R)\\&=h(H(SID_j||k)||N_R) \oplus [h(H(rc||ran))||h(H(TID_i||k)||N_R)]\\&\quad \oplus h(KRS_j||N_R)\\&=h(H(SID_j||k)||N_R) \oplus [h(H(rc||ran))||h(H(TID_i||k)||N_R)]\\&\quad \oplus h(H(SID_j||k)||N_R)\\&=[h(H(rc||ran))||h(H(TID_i||k)||N_R)],\end{aligned}$$

$$\begin{aligned}SK&=h((h(KRS_j||N_R)||s) \oplus N_U \oplus N_S \oplus N_R)\\&=h((h(KRS_j||N_R)||[h(H(rc||ran))||h(H(TID_i||k)||N_R)])\\&\quad \oplus N_U \oplus N_S \oplus N_R)\\&=h((h(H(SID_j||k)||N_R)||[h(H(rc||ran))||h(H(TID_i||k)\\&\quad ||N_R)]) \oplus N_U \oplus N_S \oplus N_R)\\&=h([h(H(SID_j||k)||N_R)||h(H(rc||ran))||h(H(TID_i||k)\\&\quad ||N_R)] \oplus N_U \oplus N_S \oplus N_R).\end{aligned}$$

Finally, SP_j responds $\{\alpha', N_R, N_S, \gamma_U\}$ to U_i. When U_i gets $\{\alpha', N_R, N_S, \gamma_U\}$, the smart card verifies the freshness for N_R and the correctness of $h(\sigma_i \oplus id_i^* \oplus pw_i^* \oplus N_R)=\alpha'$. If all verifications are successful, U_i calculates the session key SK via the following computations:

$$\begin{aligned}u&=\gamma_U \oplus h((\sigma_i \oplus id_i^* \oplus pw_i^*)||N_R)\\&=[h(H(SID_j||k)||N_R)||h(H(rc||ran))] \oplus h(H(TID_i||k)\\&\quad ||N_R) \oplus h((\sigma_i \oplus id_i^* \oplus pw_i^*)||N_R)\\&=[h(H(SID_j||k)||N_R)||h(H(rc||ran))] \oplus h(H(TID_i||k)\\&\quad ||N_R) \oplus h(H(TID_i||k)||N_R)\\&=[h(H(SID_j||k)||N_R)||h(H(rc||ran))],\\SK&=h((u||h((\sigma_i \oplus id_i^* \oplus pw_i^*)||N_R)) \oplus N_U \oplus N_S \oplus N_R)\\&=h(([h(H(SID_j||k)||N_R)||h(H(rc||ran))]||h((\sigma_i \oplus id_i^*\\&\quad \oplus pw_i^*)||N_R)) \oplus N_U \oplus N_S \oplus N_R)\\&=h([h(H(SID_j||k)||N_R)||h(H(rc||ran))||h(H(TID_i||k)||\\&\quad N_R)] \oplus N_U \oplus N_S \oplus N_R).\end{aligned}$$

Password modification phase: If U_i intends to update the password, he/she needs to insert his/her smart card into the card reader, and enters id_i^* and pw_i^*. Once the examination of $h(id_i^* \oplus pw_i^*)=TPW_i$ holds, U_i can choose a new password pw_i^{new}, and the smart card computes $TPW_i^{new}=h(id_i^* \oplus pw_i^{new})$ and $\sigma_i^{new}=\sigma_i \oplus pw_i^* \oplus pw_i^{new}$.

4 Security and performance analysis

We investigate the security and performance analysis of our proposed protocol in terms of the following perspectives.

Theorem 1 Let A be an adversary of the authenticated key exchange (AKE) security of the proposed protocol with fewer than q_s interactions with the communication entities, also asking q_h public one-way hash-queries, i.e., $h()$, and q_H private one-way hash-queries, i.e., $H()$. Then,

$$\mathrm{Adv}_P^S(A) \le \frac{q_s + q_h + {q_h}^2}{2^{l+1}} + \frac{{q_H}^2}{2^{k+1}} + \frac{q_s + q_H}{\Lambda_{AH} 2^{k+1}}.$$

Proof A sequence of game reductions is involved in the proof. We introduce a sequence of games starting at the real game G_0.

Game G_0: This is the real attack game in the random oracle model. Several oracles are available for the adversary: all users and servers instances U^i and S^j, a public hash oracle, i.e., $h()$, and a private hash oracle, i.e., $H()$. For any game G_n, we define that the event S_n occurs if $b=b'$, where b is the binary bit involved in the Test-query, and b' is the output of the adversary. By this definition, we have $\mathrm{Adv}_P^{\mathrm{ake}}(A) = 2\Pr[S_0]-1$. In addition, if the adversary has not stopped playing the game after q_s Send-queries which consume more than time t, we terminate the game and choose a random bit b' as the output, where q_s and t are pre-defined upper bounds.

Game G_1: This game simulates the public hash oracle $h()$: $\{0, 1\}^* \to \{0, 1\}^l$ and private hash oracle $H()$: $\{0, 1\}^* \to \{0, 1\}^k$ with hash lists Λ_h and Λ_H, respectively. Note that all instances such as Send-, Execute-, and Test-queries can be simulated as real players do. From this simulation, we know that this game is indistinguishable from the real attack unless the permutation properties of $h()$ and $H()$ do not hold. As a result, according to the birthday paradox, the probability that a collision occurs is at its highest:

$$\left|\Pr[S_1 - S_0]\right| \le \frac{{q_h}^2}{2^{l+1}} + \frac{{q_H}^2}{2^{k+1}}.$$

Game G_2: We avoid collisions amongst the hash queries asked by the adversary to the ephemeral secrets $H(\mathrm{SID}_j||k)$, $H(\mathrm{TID}_i||k)$, or $H(\mathrm{rc}||\mathrm{ran})$. Since only RC knows the private hash function $H()$: $\{0, 1\}^* \to \{0, 1\}^k$, we assume that the adversary maintains a list of hash functions Λ_{AH}. The adversary first chooses a random element $r \in \{0, 1\}^k$, and checks if $(*, r) \in \Lambda_{AH} \cap \Lambda_H \cap \Lambda_A$ holds, where Λ_A denotes the queries list of adversaries. If it holds, we abort this game. Games G_2 and G_1 are indistinguishable unless game G_2 is aborted. Therefore, the game will be aborted with the probability bounded for

$$\left|\Pr[S_2 - S_1]\right| \le \frac{q_s + q_H}{\Lambda_{AH} 2^{k+1}}.$$

Game G_3: We define the case where the adversary may have been lucky in guessing the session key via the public hash function $h()$: $\{0, 1\}^* \to \{0, 1\}^l$ with the query list Λ_A. Similarly, the adversary generates a random element $p \in \{0, 1\}^l$, and checks whether $(*, p) \in \Lambda_h \cap \Lambda_A$ holds or not. If it holds, we abort this game. Games G_3 and G_2 are indistinguishable unless game G_3 is aborted. The probability of this game being aborted is at its highest:

$$\left|\Pr[S_3 - S_2]\right| \le \frac{q_s + q_h}{2^{l+1}}.$$

This concludes the proof.

Theorem 2 The proposed authentication protocol possesses mutual authentication.

Proof We prove mutual authentication for our protocol based on BAN logic (Burrows *et al*., 1990). Basic constructs and logic postulates are defined as follows. Note that in this section the symbols P and Q range over principals, X and Y range over statements, and K ranges over encryption keys.

Constructs:

P believes X: The principal P believes that X is true.

P sees X: Someone has sent a message containing X to P, who can read and repeat X (possibly after doing some decryption).

P said X: P has actually sent a message including statement X in the current session of the protocol or before.

P controls X: P has jurisdiction over X; i.e., the principal P is an authority on X and this matter should be trusted.

fresh(X): X has not been sent in a message before the current session of the protocol.

$P \xleftrightarrow{K} Q$: The key K is shared between the principals P and Q.

$P \xleftrightarrow{X} Q$: The formula X is a secret known only to P and Q. Only P and Q may use X to prove their identities to each other.

$\{X\}_K$: This symbol represents the formula X encrypted or protected under key K.

Logical postulates:

Rule 1 (Message-meaning rules) If P believes $P \xleftrightarrow{K} Q$ and P sees $\{X\}_K$, then we postulate P

believes Q said X.

Rule 2 (Nonce-verification rule) If P believes fresh(X) and P believes Q said X, then we postulate P believes Q believes X.

Rule 3 (Jurisdiction rule) If P believes Q controls X and P believes Q believes X, then we postulate P believes X.

Rule 4

(1) If P sees (X, Y) then P sees X.

(2) If P believes $P \xleftrightarrow{X} Q$ and P sees $\{X\}_K$, then P sees X.

Rule 5 If one part of a formula is fresh, then the entire formula must also be fresh. If P believes fresh(X), then P believes fresh(X, Y).

Assumptions:

Before analyzing the authentication scheme, the assumptions are given as follows. Note that all symbols are the same as those in our protocol presented in Section 3.

Assumption 1 U_i and RC believe $U_i \xleftrightarrow{H(\mathrm{TID}_i||k),\mathrm{TID}_i}$ RC.

Assumption 2 SP_j and RC believe $\mathrm{SP}_j \xleftrightarrow{H(\mathrm{SID}_j||k),\mathrm{SID}_j}$ RC.

Assumption 3 U_i, SP_j, and RC believe fresh(N_U), fresh(N_S), and fresh(N_R).

Assumption 4 U_i and SP_j believe RC controls N_R, H(rc||ran).

Concrete realization of our scheme:

Step 1: $U_i \rightarrow \mathrm{SP}_j \rightarrow$RC: TID_i, SID_j, N_U, $\{N_U, \mathrm{SID}_j\}_{H(\mathrm{TID}_i||k)}$, N_S, $\{N_S, \mathrm{TID}_i\}_{H(\mathrm{SID}_j||k)}$.

Step 2: RC$\rightarrow \mathrm{SP}_j$: N_R, $\{N_R\}_{H(\mathrm{SID}_j||k)}$, $\{N_R\}_{H(\mathrm{TID}_i||k)}$, $\{H(\mathrm{rc}||\mathrm{ran}), H(\mathrm{TID}_i||k)\}_{H(\mathrm{SID}_j||k),N_R}$, $\{H(\mathrm{rc}||\mathrm{ran}), H(\mathrm{SID}_j||k)\}_{H(\mathrm{SID}_i||k),N_R}$.

Step 3: $\mathrm{SP}_j \rightarrow U_i$: N_S, N_R, $\{N_R\}_{H(\mathrm{TID}_i||k)}$, $\{H(\mathrm{rc}||\mathrm{ran}), H(\mathrm{SID}_j||k)\}_{H(\mathrm{TID}_i||k),N_R}$.

Formal analysis of mutual authentication:

(1) SP_j sees TID_i, N_U, N_R, $\{N_R\}_{H(\mathrm{SID}_j||k)}$, $\{H(\mathrm{rc}||\mathrm{ran}), H(\mathrm{TID}_i||k)\}_{H(\mathrm{SID}_j||k),N_R}$ (from steps 1 and 2 of our scheme).

(2) SP_j believes $\mathrm{SP}_j \xleftrightarrow{H(\mathrm{SID}_j||k),\mathrm{SID}_j}$RC (from Assumption 2).

(3) SP_j believes RC said $\{s, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ ((1) & (2), inferred by Rule 1).

(4) SP_j believes fresh(N_U), fresh(N_R) (from Assumption 3).

(5) SP_j believes RC believes $\{s, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ ((3) & (4), inferred by Rule 2).

(6) SP_j believes RC controls $\{s, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ (from Assumption 4).

(7) SP_j believes $\{s, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ ((5) & (6), inferred by Rule 3).

(8) U_i sees N_S, N_R, $\{N_R\}_{H(\mathrm{TID}_i||k)}$, $\{H(\mathrm{rc}||\mathrm{ran}), H(\mathrm{SID}_j||k)\}_{H(\mathrm{TID}_i||k),N_R}$ (from step 3 of our scheme).

(9) U_i believes $U_i \xleftrightarrow{H(\mathrm{TID}_i||k),\mathrm{TID}_i}$RC (from Assumption 1).

(10) U_i believes RC said $\{u, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ ((8) & (9), inferred by Rule 1).

(11) U_i believes fresh(N_S), fresh(N_R) (from Assumption 3).

(12) U_i believes RC believes $\{u, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ ((10) & (11), inferred by Rule 2).

(13) U_i believes RC controls $\{u, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ (from Assumption 4).

(14) U_i believes $\{u, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ ((12) & (13), inferred by Rule 3).

Final results are as follows:

SP_j believes RC believes $\{s, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ (from (5)).

SP_j believes $\{s, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ (from (7)).

U_i believes RC believes $\{u, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ (from (12)).

U_i believes $\{u, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ (from (14)).

Based on these four results and the assumption of the trustworthiness of RC, both remote user U_i and server SP_j can authenticate with each other via RC. In addition, session key SK can be perfectly constructed by U_i and SP_j as only they have the ability to derive $\{s, N_R, H(\mathrm{rc}||\mathrm{ran})\}$ and $\{u, N_R, H(\mathrm{rc}||\mathrm{ran})\}$, respectively.

We further adopt the AVISPA (Automated Validation of Internet Security Protocols and Applications) Tool (AVISPA Project, 2003) to analyze the security of our proposed protocol via the AVISPA Tool web interface. The AVISPA Tool consists of On-the-fly Model-Checker (Basin *et al*., 2005), CL-based Attack Search (Turuani, 2006), SAT-based Model-Checker (Armando and Compagna, 2004), and Tree Automata-based Protocol Analyzer (Boichut *et al*., 2004). The results show that no attacks are found in the proposed protocol with the AVISPA Tool.

In the following, we will discuss data confidentiality and session key security, forward security and known-key security, resistance to the replay attack, and resistance to the man-in-the-middle based attack and session key disclosure.

Data confidentiality and session key security: In our protocol, all transmitted messages are well protected via secrets $H(\mathrm{SID}_j||k)$ and $H(\mathrm{TID}_i||k)$. Without knowing RC's secret k and the private hash function $H()$, attackers cannot obtain any useful information from the transmitted cipher texts such as $\{\mathrm{TID}_i, \alpha, N_U, \mathrm{SID}_j, \beta, N_S\}$ and $\{\alpha', \beta', N_R, \gamma_S, \gamma_U\}$. In addition, the construction of the session key includes RC's another secret rc and a random nonce ran at each session. It is computationally infeasible to counterfeit the session key without knowing the secret rc and the private hash function $H()$. Therefore, data confidentiality and session key security are achieved in our protocol.

Forward security and known-key security: In our protocol, the session key $\mathrm{SK}=h([h(H(\mathrm{SID}_j||k)||N_R)||h(H(\mathrm{rc}||\mathrm{ran}))||h(H(\mathrm{TID}_i||k)||N_R)]\oplus N_U\oplus N_S\oplus N_R)$ is involved with four random numbers N_U, N_S, N_R, and ran at each session. An attacker may acquire one or more previous session keys; however, it is computationally infeasible to obtain any useful information regarding the current session key. In other words, these randomly generated numbers eliminate the correlation among all session keys. Hence, the known-key security is guaranteed. For the same reason, even though the current status of communication entities is compromised by the attacker, our protocol can possess the forward security.

Resistance to the replay attack: The attacker may replay a previously eavesdropped message to impersonate a legal communication party. However, the verification of these already transmitted messages will not pass the authentication phase, as the random numbers N_U, N_S, N_R, and ran are used only once. In that case, the attacker cannot be authenticated and the replay attack will fail.

Resistance to the man-in-the-middle based attack and session key disclosure: The attacker may issue counterfeit messages to deceive the legal communication party U_i or SP_j. However, without the knowledge of the secret $H(\mathrm{SID}_j||k)$ and $H(\mathrm{TID}_i||k)$, it is computationally infeasible for the attacker to generate valid messages such as $\{\mathrm{TID}_i, \alpha, N_U, \mathrm{SID}_j, \beta, N_S\}$ and $\{\alpha', \beta', N_R, \gamma_S, \gamma_U\}$. Even if the attacker uses the previously issued messages, the verification for these messages will fail. This is because N_U, N_S, N_R, and ran are used only once. Moreover, we correlate the two transmitted messages $\{\mathrm{TID}_i, \alpha, N_U\}$ and $\{\mathrm{SID}_j, \beta, N_S\}$ to avoid the man-in-the-middle attack during the authentication session. That is, the adversary cannot exploit a single message such as $\{\mathrm{TID}_i, \alpha, N_U\}$ or $\{\mathrm{SID}_j, \beta, N_S\}$ since these two messages are strongly connected. In addition, the value $H(\mathrm{rc}||\mathrm{ran})$ is produced for resistance to session key disclosure. That is, we totally modify the values γ_S and γ_U to present session key disclosure. The adversary cannot counterfeit the legal message without knowing the value $H(\mathrm{rc}||\mathrm{ran})$. At each new session, as ran is fresh again, the previously retrieved value cannot be used to construct a valid session key or pass the authentication procedures.

Security and computation complexity comparisons: Tables 1 and 2 show the security and performance comparison between our protocol and Chang and Cheng (2011)'s scheme. From the viewpoint of security robustness (Table 1), our proposed protocol is superior to Chang and Cheng's scheme by supporting all security requirements. As the session key is not well protected in Chang and Cheng's scheme,

Table 1 Security comparison

Type of security	Chang and Cheng (2011)'s scheme	Our proposed protocol
Data confidentiality	No	Yes
Session key security	No	Yes
Forward security	No	Yes
Known-key security	No	Yes
Resistance to replay attack	Yes	Yes
Resistance to server spoofing attack	No	Yes
Resistance to user impersonation attack	No	Yes
Resistance to session key disclosure	No	Yes

Table 2 Comparison of computation cost

Phase	Computation cost	
	Chang and Cheng (2011)'s scheme	Our proposed protocol
Registration	2 Hash+1 XOR	3 Hash+3 XOR
Login	3 Hash+4 XOR	3 Hash+6 XOR
Authentication	14 Hash+17 XOR	14 Hash+19 XOR
Total	19 Hash+22 XOR	20 Hash+28 XOR

Note that some computations can be reused in our protocol

many of the security properties cannot be promised. From the protocol efficiency perspective (Table 2), in the registration phase our protocol requires one more hash function and two more XOR operations in comparison with Chang and Cheng's scheme. During the login phase and authentication and key agreement phase, our proposed protocol requires only four more XOR operations for security enhancement. Although our authentication protocol raises few computational complexities, our proposed protocol is efficient since the computational cost of the XOR operation and one-way hash functions is lightweight. We can conclude that our protocol is almost as efficient as Chang and Cheng's scheme.

5 Conclusions

In this paper, we have demonstrated that Chang and Cheng's scheme is vulnerable to session key disclosure. To eliminate the security weakness, a novel authentication scheme with formal analysis proof is presented for security enhancement. Furthermore, the proposed scheme is efficient since only the hash function and XOR operations are adopted. In brief, without any heavy crypto-modules, our protocol is very practical and suitable to be implemented as one of the most promising mechanisms in the smart card based remote authentication protocol for the multi-server based communication environment.

References

Armando, A., Compagna, L., 2004. SATMC: a SAT-based model checker for security protocols. *Log. Artif. Intell.*, **3229**:730-733. [doi:10.1007/978-3-540-30227-8_68]

AVISPA Project, 2003. Automated Validation of Internet Security Protocols and Applications. Available from http://www.avispa-project.org.

Basin, D., Mödersheim, S., Viganò, L., 2005. OFMC: a symbolic model-checker for security protocols. *Int. J. Inf. Secur.*, **4**(3):181-208. [doi:10.1007/s10207-004-0055-7]

Boichut, Y., Héam, P.C., Kouchnarenko, O., Oehl, F., 2004. Improvements on the Genet and Klay Technique to Automatically Verify Security Protocols. Proc. 3rd Int. Workshop on Automated Verification of Infinite States Systems, p.1-11.

Burrows, M., Abadi, M., Needham, R., 1990. A logic of authentication. *ACM Trans. Comput. Syst.*, **8**(1):18-36. [doi:10.1145/77648.77649]

Chang, C.C., Cheng, T.F., 2011. A robust and efficient smart card based remote login mechanism for multi-server architecture. *Int. J. Innov. Comput. Inf. Control*, **7**(8):4589-4602.

Chang, C.C., Lee, J.S., 2004. An Efficient and Secure Multi-server Password Authentication Scheme Using Smart Card. Int. Conf. on Cyberworlds, p.417-422. [doi:10.1109/CW.2004.17]

Chang, C.C., Tsai, H.C., 2010. An anonymous and self-verified mobile authentication with authenticated key agreement for large-scale wireless networks. *IEEE Trans. Wirel. Commun.*, **9**(11):3346-3353. [doi:10.1109/TWC.2010.092410.090022]

Chen, C.L., Lai, Y.L., Chen, C.C., Chen, Y.L., 2011. A smart card-based mobile secure transaction system for medical treatment examination reports. *Int. J. Innov. Comput. Inf. Control*, **7**(5):2257-2267.

Juang, W.S., 2004. Efficient multi-server password authenticated key agreement using smart cards. *IEEE Trans. Consum. Electron.*, **50**(1):251-255. [doi:10.1109/TCE.2004.1277870]

Lee, J.S., Chang, Y.F., Chang, C.C., 2008. A novel authentication protocol for multi-server architecture without smart cards. *Int. J. Innov. Comput. Inf. Control*, **4**(6):1357-1364.

Liaw, H.T., Lin, J.F., Wu, W.C., 2006. An efficient and complete remote user authentication scheme using smart cards. *Math. Comput. Modell.*, **44**(1-2):223-228. [doi:10.1016/j.mcm.2006.01.015]

Lin, I.C., Hwang, M.S., Li, L.H., 2003. A new remote user authentication scheme for multi-server architecture. *Fut. Gener. Comput. Syst.*, **19**(1):13-22. [doi:10.1016/S0167-739X(02)00093-6]

Turuani, M., 2006. The CL-Atse Protocol Analyser. *LNCS*, **4098**:277-286. [doi:10.1007/11805618_21]