# Two-Factor User Authentication in Multi-Server Networks

Chun-Ta Li[1], Chi-Yao Weng[2,*] and Chun-I Fan[2]

[1] *Department of Information Management, Tainan University of Technology*
*529 Zhongzheng Road, Tainan City 71002, TAIWAN (R.O.C.)*
*th0040@mail.tut.edu.tw*
[2] *Department of Computer Science and Engineering, National Sun Yat-sen University*
*70 Lienhai Road, Kaohsiung City 80424, TAIWAN (R.O.C.)*
[*]*Corresponding author: cyweng@mail.cse.nsysu.edu.tw*

***Abstract***

*Recently, Chang and Cheng proposed a robust mechanism for smart card based remote logins in a multi-server architecture. However, based on the security analyzes conducted by us, we find their mechanism is vulnerable against smart card lost problems, leak-of-verifier attack and session key disclosure attack. To eliminate all identified security threats in their mechanism, we further proposed an improved version of two-factor based user authentication protocol in multi-server networks.*

***Keywords:*** *Information security; Key agreement; Multi-server architecture; Password; Smart card*

## 1. Introduction

In order to frustrate illegal users' attempts of getting the serviceable resources maintained in remote servers, two-factor (password and smart card) user authentication is the widely accepted and most adopted method in client-server architecture [5, 6, 7, 8]. However, in traditional remote login mechanisms for a multi-server architecture [2], [9], a user needs to register with different service providers and remember the various identities and passwords for ensuring higher security. Therefore, single registration is the most important feature in a multi-server architecture and any user can take desired services from various network servers without repeating registration to each service provider. Password authentication with smart card is one of the mechanisms that were widely used to authenticate the validity of participants between a login user, the service providers and a trusted registration center.

In 2001, Li, Lin and Hwang proposed an authentication scheme [4] using neural networks and asymmetric key cryptosystems. In 2004, Juang proposed an efficient and smart card based multi-server authentication scheme [3] for enhancing the system performance. In 2008, Tsai proposed a multi-server authentication scheme [11] based on one-way hash function without verification table. In 2011, Chang and Cheng developed a more robust and efficient smart card based remote login mechanism [1] in which only lightweight one-way hash function and exclusive OR operation are required during multi-server authentication processes. Unfortunately, based on the security analyses conducted by us, Chang-Cheng's login mechanism is still vulnerable against the smart card lost problems, leak-of-verifier attack and session key disclosure attack. In this paper, we will demonstrate a series of steps to show how the above-mentioned attacks can be invoked on their mechanism in the presence of a malicious adversary.

## 2. Review of Chang-Cheng's Mechanism

Chang-Cheng's mechanism consists of one trusted registration center (*RC*), service providers ($SP_j$) and login users ($U_i$). *RC* is responsible for registration of $SP_j$ and $U_i$. When $SP_j$ register with *RC* use identifier $SID_j$, *RC* computes a secret key $KRS_j = H(SID_j||k)$ and shares it with $SP_j$, where || is the string concatenation symbol, *k* is the private key of *RC* and *H*(.) is a private one-way hash function only known by *RC*.

### 2.1. Registration Phase

When a user $U_i$ wants to get the service granted in this system, $U_i$ must perform registration with *RC*. First, $U_i$ chooses his/her identifier $id_i$ and password $pw_i$ and sends the registration message {*id*, $pw_i$, *Personal Information*} to *RC* through a secure channel. Then, *RC* checks $U_i$ 's personal information and credit. If above does not hold, *RC* rejects $U_i$ 's registration; otherwise, *RC* computes a transformed identifier $TID_i = T_i||\ id_i$ as $U_i$ 's account number and saves it in the database, where $T_i$ is the registration time. The mechanism concatenates $T_i$ with $id_i$ can prevent the duplication of account numbers and $U_i$ can freely select his/her own $id_i$ and $pw_i$. In addition, *RC* computes $TPW_i = h(pw_i)$ and $\sigma_i = H(TID_i||k) \oplus pw_i$, stores ($TID_i$, *h*(.), $TPW_i$, $\sigma_i$ ) into $U_i$ 's smart card and issues this smart card to $U_i$, where $\oplus$ is the exclusive-OR symbol and *h*(.) is a public one-way hash function.

### 2.2. Login Phase

In this phase, we assume that $U_i$ wants to log in the server $SP_j$ and asks a service from $SP_j$. $U_i$ inserts his/her smart card to a input device and enters password $pw_i^*$. Then the smart card performs the flowing steps:

**Step 1**: The smart card computes $h(\ pw_i^*)$ and checks whether $h(\ pw_i^*) = TPW_i$. If it does not hold, the smart card stops the login procedure; otherwise, it goes to Step 2.

**Step 2**: The smart card computes $\alpha = h(\sigma_i \oplus pw_i^* \oplus N_U \oplus SID_j)$ and sends the login message {$TID_i$, $\alpha$ , $N_U$} to $SP_j$, where $N_U$ is a nonce chosen by $U_i$'s smart card.

**Step 3**: Upon receiving the login message from $U_i$, $SP_j$ 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 chosen by $SP_j$.

### 2.3. Authentication and Key Agreement Phase

Upon receiving the message from $SP_j$, *RC*, $SP_j$ and $U_i$ perform the following steps to achieve mutual authentication and construct a common session key shared between $U_i$ and $SP_j$.

**Step 1**: *RC* checks the freshness of $N_U$ and $N_S$ and the validity of $U_i$'s account number $TID_i$. If they does not hold, *RC* rejects this login; otherwise, it goes to Step 2.

**Step 2**: *RC* checks whether $h(H(TID_i||k) \oplus N_U \oplus SID_j) = \alpha$ and $h(H(SID_j||k) \oplus N_S) = \beta$ . If either one does not hold, *RC* rejects this connection; otherwise, *RC* convinces that $U_i$ and $SP_j$ are legal participants and goes to Step 3.

**Step 3**: *RC* chooses a random number *ran* and computes $\beta' = h(H(SID_j||k) \oplus N_R)$, $\alpha' = h(H(TID_i||k) \oplus N_R)$, $\gamma_S = h(H(SID_j||k)) \oplus (ran||h(H(TID_i||k)))$ and $\gamma_U = h(H(TID_i||k)) \oplus (h(H(SID_j||k))||ran)$, where $N_R$ is a nonce chosen by *RC*. Finally *RC* sends { $\beta'$ , $N_R$, $\alpha'$ , $\gamma_S$ , $\gamma_U$ } to $SP_j$.

**Step 4**: Upon receiving the message from *RC*, $SP_j$ checks the freshness of $N_R$. If it is invalid, $SP_j$ terminates the connection; otherwise, it goes to Step 5.

**Step 5**: $SP_j$ checks whether $h(KBS_j \oplus N_R) = \beta'$. If it holds, it goes to Step 6; otherwise, $SP_j$ stops the procedure.

**Step 6**: $SP_j$ computes $s = \gamma_S \oplus h(KRS_j)$ and $SK = h((h(KRS_j)||s) \oplus N_U \oplus N_S \oplus N_R)$ and sends $\{\alpha', N_R, \gamma_U\}$ to $U_i$, where $SK$ is the common session key shared between $SP_j$ and $U_i$.

**Step 7**: Upon receiving the response message from $SP_j$, the smart card checks the freshness of $N_R$ and verifies whether $h(\sigma_i \oplus pw_i^* \oplus N_R) = \alpha'$. If they are invalid, then authentication fails; otherwise, $U_i$ convinces that $SP_j$ is a legal participant and computes $u = \gamma_U \oplus h(\sigma_i \oplus pw_i^*)$ and the common session key $SK = h((u||h(\sigma_i \oplus pw_i^*)) \oplus N_U \oplus N_S \oplus N_R)$.

## 3. Attacks on Chang-Cheng's Mechanism

Although Chang and Cheng claimed that their mechanism can resist many types of attacks and satisfy all the essential requirements for multi-server architecture authentication. However, the actual situation is not the case and the cryptanalysis of Chang-Cheng's mechanism has been made in this section. The detailed cryptanalysis is presented as follows.

### 3.1. Smart Card Lost Problems

In this attack, we assume that $U_i$'s smart card is stolen by an adversary $U_a$ and the secret information ($TID_i$, $h(.)$, $TPW_i$, $\sigma_i$) which is stored in the smart card can be extracted by monitoring its power consumption [10].

**Off-line password guessing attack**: As we know, the content of the smart card is ($TID_i = T_i||id_i$; $h(.)$; $TPW_i = h(pw_i)$; $\sigma_i = H(TID_i||k) \oplus pw_i$). With this information, $U_a$ can select a guessable password $pw_i'$ and compute $h(pw_i')$. If $h(pw_i')$ is equal to $TPW_i$, it indicates the correct guess of $U_i$'s low-entropy password and Chang-Cheng's mechanism cannot withstand off-line password guessing attack.

**Impersonation attack**: Once the adversary $U_a$ got the secret information ($TID_i$, $h(.)$, $TPW_i$, $\sigma_i$) and correctly derived $U_i$'s password $pw_i'$, he/se can make a valid login request with ease. For example, $U_a$ computes $\alpha = h(\sigma_i \oplus pw_i' \oplus N_A \oplus SID_j)$ and makes a valid login message to impersonate $U_i$ by sending $\{TID_i, \alpha, N_A\}$ to the service provider $SP_j$, where $N_A$ is a nonce chosen by $U_a$.

### 3.2. Leak-of-Verifier Attack

In Chang-Cheng's mechanism, we found that their mechanism may suffer from leak-of-verifier attack and any legitimate user $U_i$ who possesses the smart card can easily derive service provider $SP_j$'s secret $h(H(SID_j||k))$ by performing the following steps:

**Step 1**: In Step 7 of authentication and key agreement phase, $U_i$ receives the response message $\{\alpha', N_R, \gamma_U\}$ from $SP_j$.

**Step 2**: Then, $U_i$ computes $u = \gamma_U \oplus h(\sigma_i \oplus pw_i^*) = h(H(SID_j||k))||ran$ and removes $ran$ from $h(H(SID_j||k))||ran$. Finally, $U_i$ derives $SP_j$'s secret $h(H(SID_j||k))$.

### 3.3. Session Key Disclosure Attack

In case of $SP_j$'s secret $h(H(SID_j||k))$ is successfully derived by $U_a$, $U_a$ can use it to derive the previous and subsequent session keys which are constructed by other users and $SP_j$. We assume that some victim user $U_i$'s login message $\{TID_i, \alpha, N_A\}$, $SP_j$'s authentication message $\{TID_i, \alpha, N_U, SID_j, \beta, N_S\}$ and $RC$'s response message $\{\beta', N_R, \alpha', \gamma_S, \gamma_U\}$ are collected by $U_a$. Then, the session key disclosure attack can be launched by performing the following steps:

**Step 1**: $U_a$ eavesdrops above-mentioned messages and obtains three nonces $N_U$, $N_S$, and $N_R$ from $U_i$, $SP_j$, and $RC$, respectively.

**Step 2**: $U_a$ uses $SP_j$'s secret $h(H(SID_j||k))$ to derive $s$ by computing $s = \gamma_S \oplus h(H(SID_j||k)) = ran||h(H(\mathrm{TID_i}||k))$.

**Step 3**: $U_a$ derives the session keys of past and future sessions by computing $SK = h((h(H(SID_j||k))||s) \oplus N_U \oplus N_S \oplus N_R)$.

## 4. The Proposed Protocol

According to our cryptanalysis, we proposed a more secure remote login protocol to remove the security weaknesses existing in Chang-Cheng's mechanism.

### 4.1. Registration Phase

When a user $U_i$ wants to get the service granted in this system, $U_i$ must perform registration with $RC$ as follows.

**Step 1**: $U_i$ chooses his/her identifier $id_i$ and password $pw_i$ and generates a random number $b$. Then, $U_i$ computes $h((id_i||\ pw_i) \oplus b)$ and sends the registration message $\{id_i, h((id_i||\ pw_i) \oplus b)$, *Personal Information*$\}$ to $RC$ through a secure channel.

**Step 2**: Upon receiving the registration message from $U_i$, $RC$ checks $U_i$'s personal information and credit. If above does not hold, $RC$ rejects $U_i$'s registration; otherwise, it goes to Step 3.

**Step 3**: $RC$ computes a transformed identifier $TID_i = T_i||id_i$ as $U_i$'s account number and saves it in the database, where $T_i$ is the registration time of $U_i$.

**Step 4**: $RC$ computes $\sigma_i = H(TID_i||k) \oplus h((id_i||\ pw_i) \oplus b)$ and stores $(\sigma_i, h(TID_i), h(.), T_i)$ into $U_i$'s smart card and issues this smart card to $U_i$.

**Step 5**: Upon receiving the smart card, $U_i$ stores the random number b into the smart card and $U_i$ does not need to remember $b$ after finishing the phase.

### 4.2. Login Phase

In case of $U_i$ wants to log in the server $SP_j$ and asks a service from $SP_j$, $U_i$ inserts his/her smart card to a input device and enters identifier $id_i$, password $pw_i$, and $SID_j$. Then the smart card performs the flowing steps:

**Step 1**: The smart card retrieves $T_i$ and $b$ to compute $TID_i^{'} = T_i||id_i$ and $h((id_i||\ pw_i) \oplus b)$, respectively. Then, the smart card checks whether $h(TID_i^{'}) = h(TID_i)$. If it does not hold, the smart card terminates this login; otherwise, the smart card generates a nonce $N_U$ and sends the login message $\{TID_i, \alpha_1, \alpha_2\}$ to $SP_j$, where $\alpha_1 = \sigma_i \oplus h((id_i||\ pw_i) \oplus b) \oplus N_U$ and $\alpha_2 = h((TID_i||SID_j) \oplus N_U)$.

**Step 2**: Upon receiving the login message from $U_i$, $SP_j$ computes $\beta_1 = KRS_j \oplus N_S$ and sends $\{TID_i, \alpha_1, \alpha_2, SID_j, \beta_1, \beta_2\}$ to *RC*, where $N_S$ is a nonce chosen by $SP_j$ and $\beta_2 = h((SID_j || TID_i) \oplus N_S)$.

### 4.3. Authentication and Key Agreement Phase

Upon receiving the message from $SP_j$, *RC*, $SP_j$ and $U_i$ perform the following steps to achieve mutual authentication and construct a common session key shared between $U_i$ and $SP_j$.

**Step 1**: *RC* checks the validity of $U_i$'s account number $TID_i$ and $SP_j$'s identifier $SID_j$. If they does not hold, *RC* rejects this login; otherwise, it goes to Step 2.

**Step 2**: *RC* computes $N_U^{'} = \alpha_1 \oplus H(TID_i||k)$ and checks the freshness of $N_U^{'}$ and the validity of $h((TID_i||SID_j) \oplus N_U^{'}) = \alpha_2$. If either one does not hold, *RC* rejects this connection; otherwise, *RC* convinces that $U_i$ is a legal user and goes to Step 3.

**Step 3**: *RC* computes $N_S^{'} = \beta_1 \oplus H(SID_j||k)$ and checks the freshness of $N_S^{'}$ and the validity of $h((SID_j || TID_i) \oplus N_S^{'}) = \beta_2$. If either one does not hold, *RC* rejects this connection; otherwise, *RC* convinces that $SP_j$ is a legal service provider and goes to Step 4.

**Step 4**: *RC* computes $\alpha' = h(N_U^{'}) \oplus N_S^{'} \oplus N_R$, $\gamma_U = h(H(TID_i||k) \oplus SK)$, $\beta' = h(N_S^{'}) \oplus N_U^{'} \oplus N_R$ and $\gamma_S = h(H(SID_j||k) \oplus SK)$, where $N_R$ is a nonce chosen by *RC* and *SK* is a common session key which is constructed by computing $SK = h(N_U^{'} \oplus N_S^{'} \oplus N_R)$. Finally *RC* sends $\{\alpha', \gamma_U, \beta', \gamma_S\}$ to $SP_j$.

**Step 5**: Upon receiving the message from *RC*, $SP_j$ computes $\beta'' = \beta' \oplus h(N_S)$and $SK_S = h(\beta'' \oplus N_S)$ and checks whether $h(H(SID_j||k) \oplus SK_S) = \gamma_S$. If it is invalid, $SP_j$ terminates the connection; otherwise, $SP_j$ convinces that *RC* and $U_i$ are legal participants and sends $\{\alpha', \gamma_U\}$ to $U_i$. Note that $SK_S = SK$ is the common session key shared between $SP_j$, $U_i$ and *RC*.

**Step 6**: Upon receiving the response message from $SP_j$, the smart card computes $\alpha'' = \alpha' \oplus h(N_U)$ and $SK_U = h(\alpha'' \oplus N_U)$ and checks whether $h(H(TID_i||k) \oplus SK_U) = \gamma_U$. If it is invalid, $U_i$ terminates the connection; otherwise, $U_i$ convinces that *RC* and $SP_j$ are legal participants and $SK_U = SK = SK_S$ is the common session key shared between $U_i$, $SP_j$ and *RC*.

## 5. Security Analysis

Here we mainly discussed and analyzed the security of the proposed protocol on various known cryptographic attacks and demonstrated its strength in terms of security.

**Off-line password guessing attack**: In case of $U_i$'s smart card is stolen by an adversary $U_a$ and he/she has the ability to obtain the contents of the smart card ($\sigma_i$, $h(TID_i)$, $h(.)$, $T_i$, $b$). It does not help $U_a$ to derive or modify $U_i$'s password without knowing $U_i$'s identity $id_i$ and password $pw_i$. In addition, from the proposed protocol, we can see that the login messages $\{TID_i, \alpha_1, \alpha_2\}$ and $\{TID_i, \alpha_1, \alpha_2, SID_j, \beta_1, \beta_2\}$ only contain information about $H(TID_i||k)$, $H(SID_j||k)$, $N_U$ and $N_S$. They do not include any information about the password. Therefore, this design of the proposed protocol can protect the system against off-line password guessing attack.

**Impersonation attack**: The proposed protocol must protect the system against impersonation attacks on both the user, service provider and the registration center side. First, on the user and service provider side, it is difficult for $U_a$ to create correct $\alpha_1$ and $\beta_1$ without the user's secret token $H(TID_i||k)$ and the service provider's secret token $H(SID_j||k)$. Moreover, it is difficult for $U_a$ to get the registration center's secret key $k$. On the registration center side, $U_a$ cannot derive the random nonces $N_U$ and $N_S$ without $H(TID_i||k)$ and $H(SID_j||k)$, which means $U_a$ cannot compute the correct hash values $h(N_U)$ and $h(N_S)$ in order to impersonate the registration center.

**Leak-of-verifier attack**: In our proposed protocol, a registered user $U_i$ may try to obtain the secret token $H(TID_i||k)$ from his/her smart card by extracting $\sigma_i$ and $b$ and computing $H(TID_i||k) = \sigma_i \oplus h((id_i||\ pw_i)\oplus b)$. However, it does not help $U_i$ to obtain $SP_j$'s secret token $H(SID_j||k)$ and nonce $N_S$ from the login parameters $\beta_1$ and $\beta_2$ due to the protection of one-way hash function and the security of large nonce (e.g., bigger than 256 bits). On the other hand, the random nonce $N_R$ is known to only registration center $RC$ and it is masked by $\alpha' = h(N_U)\oplus N_S\oplus N_R$ and $\beta' = h(N_S)\oplus N_U\oplus N_R$. Therefore, $U_i$ and $SP_j$ cannot derive $N_R$ from $\alpha'$ and $\beta'$ without knowing $SP_j$'s $N_S$ and $U_i$'s $N_U$, respectively.

**Session key disclosure attack**: After the successful user authentication, the user, the service provider and the registration center computes the common session key $SK= h(N_U\oplus N_S\oplus N_R)$ and an adversary $U_a$ may try to derive $SK$ to damage the later communications between them. However, in Step 4 of authentication and key agreement phase, $U_a$ cannot obtains $N_S\oplus N_R$ and $N_U\oplus N_R$ without knowing $h(N_U)$ and $h(N_S)$, respectively. As a result, $U_a$ cannot get success in the proposed protocol by session key disclosure attack.

## 6. Conclusions

In this paper, we showed that Chang and Cheng's two-factor based multi-server authentication protocol is insecure. To offset these security weaknesses found in Chang-Cheng's protocol, we have proposed an improved version of two-factor based multi-server authentication protocol. Security analysis shows that our proposed protocol is resilient to various attacks and achieves mutual authentication and session key agreement.

## References

[1] C. C. Chang and T. F. Cheng, "A robust and efficient smart card based remote login mechanism for multi-server architecture", International Journal of Innovative Computing, Information and Control, 7(8), **(2011)**, pp. 4589-4602.

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

[3] W. Juang, "Efficient multi-server password authenticated key agreement using smart cards", IEEE Transactions on Consumer Electronics, 50(1), **(2004)**, pp. 251-255.

[4] L. H. Li, I. C. Lin and M. S. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks", IEEE Transactions on Neural Network, 12(6), **(2001)**, pp. 1498-1504.

[5] C. T. Li and C. C. Lee, "A robust remote user authentication scheme using smart card", Information Technology and Control, 40(3), **(2011)**, pp. 236-245.

[6] C. T. Li and C. C. Lee, "A novel user authentication and privacy preserving scheme with smart cards for wireless communications", Mathematical and Computer Modelling, 55(1-2), **(2012)**, pp. 35-44.

[7] C. T. Li and M. S. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks", Information Sciences, 181(23), **(2011)**, pp. 5333-5347.

[8] C. T. Li, C. C. Yang and M.S. Hwang, "A secure routing protocol with node selfishness resistance in MANETs", International Journal of Mobile Communications, 10(1), **(2012)**, pp. 103-118.

[9] X. Li, Y. Xiong, J. Ma and W. Wang, "An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards", Journal of Network and Computer Applications, 35(2), **(2012)**, pp. 763-769.

[10] T. S. Messerges, E. A. Dabbish and R.H. Sloan, "Examining smart-card security under the threat of power analysis attacks", IEEE Transactions on Computers, 51(5), **(2002)**, pp. 541-552.

[11] J. L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table", Computers & Security, 27(3-4), **(2008)**, pp. 115-121.