ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **12**, December **2016** pp. **2817–2822**

THE WEAKNESSES OF KARUPPIAH ET AL.'S REMOTE USER MUTUAL AUTHENTICATION SCHEME USING SMART CARD

Wen-Gong Shieh and Ping Yu*

Department of Information Management
Chinese Culture University
No. 55, Hwa-Kang Rd., Yang-Ming-Shan, Taipei 11114, Taiwan
wgshieh@faculty.pccu.edu.tw; *Corresponding author: yp@faculty.pccu.edu.tw

Received May 2016; accepted August 2016

Abstract. *Remote user authentication schemes with smart card become more and more important due to the popularity of Internet. In 2014, Karuppiah et al. proposed a remote authentication scheme using smart card with the functions of session key agreement, mutual authentication, and forward secrecy. However, we find that Karuppiah et al.'s scheme is still vulnerable to various attacks when the login request message of a user is intercepted and the contents of the user's smart card are extracted. Firstly, an attacker may successfully perform an offline password guessing attack and then perform an offline identity guessing attack. With the correct identity and password, the attacker can successfully perform an impersonation attack that breaks mutual authentication. In addition, if the long-term secret keys are compromised, then all the used session keys will be revealed. That is, Karuppiah et al.'s scheme does not satisfy the property of perfect forward secrecy. Moreover, we find some errors in their performance evaluation. Their scheme requires higher computation cost than the claimed cost in their evaluation. Through entire analysis, we find that Karuppiah et al.'s scheme may not be suitable for network applications requiring user privacy and security.*
Keywords: Security, Remote user authentication, Smart card, Anonymity, Forward secrecy

1. **Introduction.** Remote mutual authentication has been an important issue for network applications of E-commerce. Many researchers have proposed authentication schemes to improve the security and efficiency in this field. In 2008, using smart cards, Tsai [1] proposed a multi-server authentication scheme that does not need to store any verification table in the server. Juang et al. [2] proposed a password-authenticated key agreement scheme without time-synchronization problem. To achieve user anonymity, Liao and Wang [3] proposed a scheme based on the idea of dynamic ID for a multi-server environment. In 2010, Li et al. [4] also proposed a scheme that addresses the property of un-traceability over communication channels. In 2013, Tsai et al. [5] presented an anonymous authentication scheme to offer initiator un-traceability without requiring server's database support. In 2014, Yu et al. [6] proposed a generic three-factor framework for authentication using password, smart card, and biometrics. In the same year, Kumari et al. [7] and Karuppiah and Saravanan [8] also proposed their remote authentication schemes using smart cards. The former provides user anonymity with un-traceability while the latter uses exponential operations for achieving forward secrecy. However, we find that Karuppiah et al.'s scheme is still vulnerable to many attacks when the attacker gets the smart card and extracts its contents. At the login stage, the attacker can intercept the login message sent from the card to perform an offline password guessing attack. After obtaining the user's password, the attacker can break the user anonymity un-traceability and authentication. Besides, their scheme does not satisfy the property of perfect forward secrecy. We also find some errors in their performance evaluation. The remainder of this

paper is presented as follows. In the next section, we briefly review Karuppiah et al.'s scheme. After that, we propose our attacks to their scheme in Section 3. In Section 4, we discuss their performance evaluation problem. Finally, we give our conclusion in the last section.

2. **Review of Karuppiah et al.'s Scheme.** There are five phases in Karuppiah et al.'s scheme [7,8] including initialization, registration, login, verification, and password change phases. The notations used in this paper are summarized in Table 1.

TABLE 1. The notations used in this paper

Notations	**Description**
U_i	the ith user
ID_i	the identity of user U_i
PWD_i	the password of user U_i
S	the server/central authority system
r	a random number chosen by server S
p, q	two large prime numbers selected by server S
n, $\phi(n)$	$n = p \times q$ and $\phi(n) = (p-1) \times (q-1)$
e	a prime number such that $gcd(e, \phi(n)) = 1$ and $1 < e < \phi(n)$
d	the secret key of the server S such that $d \equiv e^{-1} \bmod \phi(n)$
g	a generator of Z_p^*
$h(\cdot)$	a cryptographic one-way hash function
T	the current timestamp of the smart card reader clock
T_S	the current timestamp of the server clock
T_R	the registration timestamp of user based on the server clock
ΔT	an expected legal time interval for transmission delay
$\oplus$	the bitwise X-OR operation
$\|\|$	String concatenation operation

2.1. **Initialization phase.** Assume a server wants to provide the remote user authentication service. Firstly, it selects two large prime numbers p and q and a generator g from a finite field in Z_p^*. Secondly, it computes $n = p \times q$ and $\phi(n) = (p-1) \times (q-1)$. Thirdly, it selects an integer e such that $gcd(e, \phi(n)) = 1$ and $1 < e < \phi(n)$. Then, it computes an integer $d \equiv e^{-1} \bmod \phi(n)$, where d is the secret key (private key) and $y = g^d \bmod n$ is the public key of the server. Finally, the server keeps $(d,\ p,\ q)$ secretly.

2.2. **Registration phase.** Assume a user U_i wants to register to a server S. U_i chooses her/his identity ID_i, password PWD_i and a random number b. Then, U_i computes $h(b \oplus PWD_i)$ and sends the message $\{ID_i, h(b \oplus PWD_i)\}$ to S for registration via a secure channel. When S receives the request, S verifies the ID_i. If ID_i does not exist, S computes $B_1 = h(ID_i)^{h(b \oplus PWD_i)} \bmod n$ and $C_{in} = y^{h(d||T_R||ID_i)+h(b \oplus PWD_i)} \bmod n$. Then S stores (ID_i, T_R) in its database and gives U_i a smart card SC_i containing $\{C_{in}, B_1, g, y, n, h(\cdot)\}$ via a secure channel, where T_R is the registration timestamp based on the server clock. After receiving the smart card, U_i inserts the random number b into it. The contents in SC_i become $\{C_{in}, B_1, g, y, n, b, h(\cdot)\}$.

2.3. **Login and verification phase.** When U_i wants to login to the server, U_i inserts the smart card SC_i and inputs ID_i and PWD_i. The smart card SC_i computes $B_1^* = h(ID_i)^{h(b \oplus PWD_i)} \bmod n$ and checks whether B_1^* is equivalent to the stored B_1. If it fails, SC_i terminates the session. Otherwise, SC_i selects a random number j and computes $B_2 = g^j \bmod n$, $B_3 = y^j \bmod n$, $C = ID_i \oplus h(B_2 \oplus B_3)$, $C'_{in} = C_{in} \times y^{-h(b \oplus PWD_i)} \bmod n =$

$y^{h(d||T_R||ID_i)} \bmod n$ and $M = h(C'_{in}||C)$. Then SC_i sends the message $SRQ = \{B_2, M, C\}$ to the server S.

After receiving the message from U_i at time T_S, the server S computes $B'_3 = (B_2)^d \bmod n$ and $ID_i = C \oplus h(B_2 \oplus B'_3)$. If the format of ID_i is valid, S computes $C^* = y^{h(d||T_R||ID_i)} \bmod n$ and checks whether $M^* = h(C^*||C)$ is equivalent to the received M. If it fails, S rejects U_i's login request. Otherwise, it accepts U_i's request. Using the current timestamp T_S and a new random number r, S computes $t = h(T_S \oplus ID_i \oplus ID_S \oplus B'_3)$, $C_1 = (C^*)^{r+t} \bmod n$ and $h(C_1)$ and sends a message $X = \{h(C_1), r, T_S\}$ to the user U_i. Upon receiving the message at time T from S, if T_S is invalid or $(T - T_S) > \Delta T$, SC_i terminates the session, where ΔT is the predetermined time interval for message traveling. Otherwise, SC_i computes $t^* = h(T_S \oplus ID_i \oplus ID_S \oplus B_3)$ and $C_2 = (C'_{in})^{r+t^*} \bmod n$ and checks whether $h(C_2)$ is equivalent to the received $h(C_1)$. If it fails, U_i terminates the session. Otherwise, U_i computes $M_1 = h(C_2 \oplus ID_i)^T \bmod n$. U_i sends message $Z = \{M_1, T\}$ to server and uses $S^U_{Key} = h(ID_i||ID_S||C_2)$ as the session key for the communication session.

After receiving the message Z from U_i at time T_S, the server S checks the freshness of T. Reject U_i's request if the difference between T_S and T is greater than ΔT. Then, S checks whether $M_2 = h(C_1 \oplus ID_i)^T \bmod n$ is equivalent to the received M_1. If it fails, S rejects U_i's login request. Otherwise, S successfully authenticates U_i and uses $S^S_{Key} = h(ID_i||ID_S||C_1)$ as the session key.

2.4. **Password change phase.** In Karuppiah et al.'s scheme, the password change phase is simple that the smart card SC_i alone can accept or reject the password change request of U_i. When U_i wants to change the password, she/he inserts her/his smart card SC_i into a card reader and inputs her/his ID_i and PWD_i. The smart card SC_i computes $h(ID_i)^{h(b \oplus PWD_i)} \bmod n$ and checks if it is equal to the stored B_1. If it fails, SC_i rejects U_i's password change request. Otherwise, it accepts U_i's request and allows U_i to input two times the new password PWD_{i_new} to confirm the input. Then, SC_i computes $B_{1\_new} = h(ID_i)^{h(b \oplus PWD_{i\_new})} \bmod n$ and $C_{in\_new} = y^{h(d||T_R||ID_i)+h(b \oplus PWD_{i\_new})} \bmod n$, and replaces (C_{in}, B_1) by (C_{in_new}, B_{1_new}).

3. **Our Attacks to Karuppiah et al.'s Scheme.** In this section, we demonstrate the weaknesses of Karuppiah et al.'s scheme. We follow two assumptions regarding capabilities of an adversary as suggested by Kocher et al. [9] and Messerges et al. [10] respectively. Firstly, an adversary has control over the communication channel connecting the users and the remote server in login/verification phase that the adversary can intercept, insert, delete, or modify any message transmitted via a common channel. Secondly, an adversary may either steal a user's smart card or obtain a user's password, but not both. From previous two assumptions, we can analyze the security problems existing in Karuppiah et al.'s scheme.

3.1. **Offline password guessing.** The password guessing attack has two types, online password guessing, and offline password guessing [7]. In Karuppiah et al.'s scheme, the smart card is designed to allow only three continuous login attempts within a short time interval to confirm the correctness of entered identity and password before computing any login request. Hence, their scheme can withstand online password guessing attack.

For the offline password guessing attack, as the previous assumption one and two, the attack U_a can intercept a successful login request message $SRQ = \{B_2, M, C\}$ of U_i, get U_i's smart card SC_i and extract all its values $\{C_{in}, B_1, g, y, n, h(\cdot), b\}$. Then U_a can perform offline password guessing by computing first the value $C'_{in} = C_{in} \times y^{-h(b \oplus PWD'_i)} \bmod n$ using a guessed password PWD'_i, where C_{in}, y, b, and n are extracted from SC_i. Next, U_a checks if the equation $M = h(C'_{in}||C)$ is true to verify the correctness of the guessed PWD'_i, where M and C are from the intercepted SRQ. If it is true, U_a has successfully guessed U_i's password. That is, $PWD'_i = PWD_i$. At the same time, U_a knows the value

C'_{in}. Otherwise, U_a repeats all the steps with some other guessed PWD'_i until he/she succeeds. In other words, the attacker can verify the correctness of the guessed password in an offline manner.

3.2. **Breaking user anonymity and un-traceability.** In Karuppiah et al.'s scheme, if the attacker U_a gets the password PWD_i of U_i, U_a can break the anonymity of the user U_i. As given in Section 3.1, after the attacker U_a successfully guesses the password PWD_i of U_i, the attacker U_a can perform offline identity guessing attack to derive ID_i. First, U_a computes $B'_1 = h(ID'_i)^{h(b \oplus PWD_i)} \bmod n$ using a guessed ID'_i, where b and n are extracted from SC_i. Next, if $B'_1 = B_1$, where B_1 is extracted from SC_i, it implies that U_a has successfully guessed user identity. That is, $ID'_i = ID_i$. Otherwise, U_a repeats the process with some other guessed ID'_i until she/he succeeds. Therefore, Karuppiah et al.'s scheme does not provide user anonymity and un-traceability.

3.3. **Breaking session key forward secrecy.** Forward secrecy guarantees that a session key derived from a set of long-term keys cannot be compromised if one of the long-term keys is compromised in the future. In Karuppiah et al.'s scheme, assume that an attacker U_a gets the server secret key d, the successful login message $SRQ = \{B_2, M, C\}$ of U_i, the response message $X = \{h(C_1), r, T_S\}$ from the server, and all values stored in the smart card SC_i of U_i. There are two methods that the attacker can compute the session key of the user, using the equation $S^U_{Key} = h(ID_i||ID_S||C_2)$ or $S^S_{Key} = h(ID_i||ID_S||C_1)$.

Firstly, consider $S^U_{Key} = h(ID_i||ID_S||C_2)$. From Section 3.1, U_a can offline guess a weak password PWD_i successfully and get the value C'_{in} at the same time. Next, using the values B_2 and C in SRQ and the server secret key d, U_a can compute $B'_3 = (B_2)^d \bmod n$, $ID_i = C \oplus h(B_2 \oplus B'_3)$ and $t = h(T_S \oplus ID_i \oplus ID_S \oplus B'_3)$ where T_S is available in the response message $X = \{h(C_1), r, T_S\}$. At this point, U_a can compute $C_2 = (C'_{in})^{r+t} \bmod n$ and the session key $S^U_{Key} = h(ID_i||ID_S||C_2)$.

Secondly, consider $S^S_{Key} = h(ID_i||ID_S||C_1)$. Instead of offline guessing the password PWD_i to get the value C'_{in} at the same time, U_a can perform an offline guessing attack to derive the value T_R and get the value C^* at the same time. Note that $C'_{in} = C^*$. Again, using the values B_2 and C in a login message SRQ and the server secret key d, U_a can compute the identity ID_i of every login message. By computing $B'_3 = (B_2)^d \bmod n$ and $ID_i = C \oplus h(B_2 \oplus B'_3)$. It allows U_a to trace every U_i's successful login and response messages. Because T_R is the registration timestamp of U_i, the time in the timestamp should be close to but before the time T_{see} of the first observed login message of U_i. Therefore, U_a can guess a value T'_R whose time is close but before T_{see}, compute $C^* = y^{h\left(d||T'_R||ID_i\right)} \bmod n$ and check if $M = h(C^*||C)$ is true, where M and C are in U_i's login message SRQ. If $M = h(C^*||C)$ is true, it implies that U_a successfully guesses T_R, $T_R = T'_R$ and get the value C^* at the same time. Otherwise, U_a repeats the process with some other guessed T'_R until she/he succeeds. After that, U_a can compute $t = h(T_S \oplus ID_i \oplus ID_S \oplus B'_3)$, $C_1 = (C^*)^{r+t} \bmod n$ and the session key $S^U_{Key} = h(ID_i||ID_S||C_1)$. Therefore, Karuppiah et al.'s scheme cannot provide session key forward secrecy.

3.4. **Breaking user authentication.** For Karuppiah et al.'s scheme, we can show that an attacker U_a can successfully login to the server S_i using the stolen smart card and an intercepted login message of U_i. From previous Sections 3.1 and 3.2, the attacker U_a can get the password PWD_i and identity ID_i of U_i. From those values, U_a can select random number j', to compute $B'_2 = g^{j'} \bmod n$, $B'_3 = y^{j'} \bmod n$ and $C' = ID_i \oplus h(B'_2 \oplus B'_3)$. With the known value C'_{in} from Section 3.1, U_a can compute $M' = h(C'_{in}||C')$ and send the login request message $SRQ' = \{B'_2, M', C'\}$ to the server. After receiving the fake message SRQ' form U_a at time T_S, the server S first computes $B''_3 = (B'_2)^d \bmod n$ and checks the valid of ID_i from computing $ID_i = C' \oplus h\left(B'_2 \oplus B''_3\right)$. Because $B''_3 =$

$(B_2')^d \bmod n = \left(g^{j'} \bmod n\right)^d \bmod n = g^{j' \times d} \bmod n = \left(g^d \bmod n\right)^{j'} \bmod n = y^{j'} \bmod n = B_3'$ and the identity $ID_i = C' \oplus h\left(B_2' \oplus B_3''\right)$ is valid. The S computes $C^* = y^{h(d||T_R||ID_i)} \bmod n$ and checks whether $M^{*\prime} = h(C^*||C')$ is equivalent to the received M'. Because C^* is equal to C_{in}' and $M' = h(C_{in}'||C')$, the server S accepts U_a's request. Using the current timestamp T_S' and a random number r generated, S computes $t' = h\left(T_S' \oplus ID_i \oplus ID_S \oplus B_3''\right)$ and $C_1' = (C^*)^{r+t'} \bmod n$. Then, the server computes $h(C_1')$ and sends message $X = \{h(C_1'), r, T_S'\}$ to the attacker U_a. Upon receiving the message form S at time T_a, U_a computes $t^{*\prime} = h\left(T_S' \oplus ID_i \oplus ID_S \oplus B_3'\right)$ and $C_2' = (C_{in}')^{r+t^{*\prime}} \bmod n$ and computes $M_1' = h(C_2' \oplus ID_i)^{T_a} \bmod n$. Then, U_a sends message $Z' = \{M_1', T_a\}$ to server and prepares to use $S_{Key}^{U'} = h(ID_i||ID_S||C_2')$ as the session key. After receiving the message Z' form U_a at time T_S', the server S checks the freshness of T_a. The difference between T_S' and T_a is smaller than ΔT. S will find $M_2' = h(C_1' \oplus ID_i)^{T_a} \bmod n$ is equivalent to the received M_1', because $M_1' = h(C_2' \oplus ID_i)^{T_a} \bmod n$ with $C_2' = (C_{in}')^{r+t^{*\prime}} \bmod n = (C^*)^{r+t'} \bmod n = C_1'$. The server S will successfully authenticate U_a, and uses $S_{Key}^S = h(ID_i||ID_S||C_1')$ as the session key.

4. **The Performance Evaluation Problem.** Karuppiah et al. compare the performance of their scheme with related schemes. To facilitate the computational costs analysis, they define each computational cost including hash operation t_h, modular exponent t_{mexp}, symmetric key encryption/decryption t_{sym} and multiplication/division t_m. However, we find some errors in their performance evaluation that the cost of login phase of their scheme should be $9t_{mexp} + 1t_m + 13t_h$, instead of $6t_{mexp} + 1t_m + 5t_h$. We can find that $B_1^* = h(ID_i^*)^{h(b \oplus PWD_i^*)} \bmod n$ with $1t_{mexp}$ and $2t_h$, $B_2 = g^j \bmod n$ with $1t_{mexp}$, $B_3 = y^j \bmod n$ with $1t_{mexp}$, $C = ID_i \oplus h(B_2 \oplus B_3)$ with $1t_h$, $C_{in}' = y^{h(d||T_R||ID_i)} \bmod n$ with $1t_{mexp}$ and $1t_h$, $M = h(C_{in}'||C)$ with $1t_h$, $B_3' = (B_2)^d \bmod n$ with $1t_{mexp}$, $ID_i = C \oplus h(B_2 \oplus B_3')$ with $1t_h$, $C^* = y^{h(d||T_R||ID_i)} \bmod n$ with $1t_{mexp}$ and $1t_h$, $M^* = h(C^*||C)$ with $1t_h$, $t = h(T_S \oplus ID_i \oplus ID_S \oplus B_3')$ with $1t_h$, $C_1 = (C^*)^{r+t} \bmod n$ with $1t_{mexp}$, $X = \{h(C_1), r, T_S\}$ with $1t_h$, $t^* = h(T_S \oplus ID_i \oplus ID_S \oplus B_3)$ with $1t_h$, $C_2 = (C_{in}')^{r+t^*} \bmod n$ with $1t_{mexp}$, $h(C_2) = h(C_1)$ with $1t_h$ and $M_1 = h(C_2 \oplus ID_i)^T \bmod n$ with $1t_{mexp}$ and $1t_h$. From this result, we can observe that Karuppiah et al.' scheme imposes a greater computational cost due to the modular exponential operations. Moreover, their scheme cannot protect users and the server against many attacks discussed in the previous sections as the other related schemes.

5. **Conclusion.** In this paper, we analyze the weaknesses of Karuppiah et al.'s remote user authentication scheme and show its vulnerability in many attacks. When an attacker gets a smart card with its contents and an interpreted login request message, the attacker can perform an offline password guessing attack to derive the weak password of the owner of the smart card. After getting the password, the attacker can perform offline identity guessing to derive the user's identity. That is, their scheme does not provide user anonymity and un-traceability. Moreover, with the correct identity and password, the attacker can perform user impersonation attack to login into the server. In addition, if the long-term secret key in the server is compromised, we show that all the used short term session keys will be revealed. That is, Karuppiah et al.'s scheme does not satisfy the property of perfect forward secrecy. Moreover, we find some errors in their performance evaluation. The computational cost of the login phase in their scheme should be nine modular exponents, one multiplication and thirteen hash operations, instead of six modular exponents, one multiplication and five hash operations. Through entire analysis, we find that Karuppiah et al.'s scheme may not be suitable for network applications requiring user privacy and security. We plan in the future using the biometric information to implement a new remote user authentication scheme with smart card that can satisfy

all desirable security requirements. The scheme that can survive in smart card loss and withstand the threats also belongs to our future plan.

REFERENCES

[1] J.-L. Tsai, Efficient multi-server authentication scheme based on one-way hash function without verification table, *Computers & Security*, vol.27, pp.115-121, 2008.

[2] W.-S. Juang, S.-T. Chen and H.-T. Liaw, Robust and efficient password-authenticated key agreement using smart cards, *IEEE Trans. Industrial Electronics*, vol.55, pp.2551-2556, 2008.

[3] Y.-P. Liao and S.-S. Wang, A secure dynamic ID based remote user authentication scheme for multi-server environment, *Computer Standards & Interfaces*, vol.31, pp.24-29, 2009.

[4] X. Li, W. Qiu, D. Zheng, K. Chen and J. Li, Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards, *IEEE Trans. Industrial Electronics*, vol.57, pp.793-800, 2010.

[5] J.-L. Tsai, N.-W. Lo and T.-C. Wu, Novel anonymous authentication scheme using smart cards, *IEEE Trans. Industrial Informatics*, vol.9, pp.2004-2013, 2013.

[6] J. Yu, G. Wang, Y. Mu and W. Gao, An efficient generic framework for three-factor authentication with provably secure instantiation, *IEEE Trans. Information Forensics and Security*, vol.9, pp.2302-2313, 2014.

[7] S. Kumari, M. K. Khan and X. Li, An improved remote user authentication scheme with key agreement, *Computers & Electrical Engineering*, vol.40, pp.1997-2012, 2014.

[8] M. Karuppiah and R. Saravanan, A secure remote user mutual authentication scheme using smart cards, *Journal of Information Security and Applications*, vol.19, pp.282-294, 2014.

[9] P. Kocher, J. Jaffe and B. Jun, Differential power analysis, *Proc. of Advances in Cryptology CRYPTO'99*, 1999.

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