An Improved Group Key Agreement Scheme with Privacy Preserving Based on Chaotic Maps

Chun-Ta Li, Chi-Yao Weng, Chien-Ming Chen, and Cheng-Chi Lee

***Abstract*—The decentralized group key agreement scheme allows a group of participants to exchange private multicast messages via the protection of a group session key in the group. Recently, Zhu proposed the first group authentication key agreement scheme based on chaotic maps and the structure of a group is organized in an ordered chain. Thus each participant needs to establish two temporary two-party session keys with its predecessor and successor in a parallel algorithm. In order to cope with dynamic groups, the group session keys are frequently updated whenever a new member joins or a member leaves the group. Zhu claimed that the proposed scheme is secure against various attacks such as replay, man-in-the-middle, impersonation and key compromise attacks. Furthermore, Zhu extended the proposed scheme to high level security attributes such as privacy preserving, no clock synchronization problem, mutual and group authentication and perfect forward secrecy etc. However, in this paper, we found that Zhu's scheme is vulnerable to successor impersonation problem and this weakness leads to a malicious adversary from deriving group session keys after impersonate attack. Moreover, their scheme is vulnerable to known key attack and this problem may lead to an adversary to compromise the previous and future group session keys. To overcome these security flaws, in this paper, we significantly improve the security of Zhu's group key agreement scheme without increasing the communication overhead and computation complexity.**

***Index Terms*—Cryptanalysis, chaotic maps, group key management, impersonation attack, privacy preserving.**

I. Introduction

The rise of chaos theory has incurred great benefits on cryptography [1]-[5], many chaotic maps based user authentication and key agreement schemes are widely being introduced for network-based applications, such as cloud-assisted system [6], [7], WSN system [8], multi-server environment [9], [10], ubiquitous computing environment [11], medical care system [12]-[14] and so on. For N-party authenticated key agreement and mutual authentication literatures, the first work is proposed by Bresson *et al.* [15] in 2002. Their solution is based on a human-memorable password and they prove its security in both the random oracle and the ideal cipher models. In 2003, Sherman and McGrew proposed one-way function trees based key establishment scheme [16] in large dynamic groups. In 2004, Lee *et al.* [17] introduced the 2-round password-based authenticated group key exchange protocol with 3-exponentiations required for each participant in the group. In 2006, Abdalla *et al.* [18] suggested a password-based constant-round group key exchange protocol with provable security. Their solution requires 4-rounds of communication and 4-exponentiations per user. In the same year, Dutta *et al.* [19] proposed a new encrypted group key agreement protocol and their method requires 2-rounds of communication, 3-exponentiations, 4 one-way hash function computations, 2 encryptions and $n + 1$ decryptions per user, where n is the number of group members. In 2008, Xu and Huang proposed a multicast key distribution scheme [20] and the computational complexity of their scheme is reduced by using maximum distance separable codes. In 2009, Zheng *et al.* [21] presented an efficient and provably secure password-based group key agreement protocol without using public key infrastructure and each user only requires 4-rounds of communication and 4-exponentiations. In 2010, Je *et al.* proposed an efficient key tree management protocol [22] and the computation and storage costs of their protocol is reduced by examining the resource information of each group member's device. In 2013, Vijayakumar *et al*. proposed a greatest common divisor based centralized key distribution protocol [23] to ensure high security with less computation, communication and storage complexity. In 2014, Vijayakumar *et al*. further proposed a new centralized group key management based on the Chinese remainder theorem called CRTGKM algorithm [24]. In the key server side, the computation complexity of their algorithm is $O(1)$ when a member joins or leaves from the multicast group. Moreover, in the group member side, the computation complexity is minimized and a multicast group member performs only one modulo division operation.

In 2016, Zhu proposed a new group key agreement scheme based on chaotic maps [25]. Firstly each group participant requires six communication rounds to compute two two-party agreement keys with its successor and predecessor. Then all group members use hash function and exclusive OR operations to authenticate each other and compute a secure group session key. In case of members' revocation or join, Zhu's scheme refrains from using heavyweight computations and preserves perfect forward secrecy with privacy preserving. However, we analyze the security of Zhu's scheme and show

Manuscript received February 4, 2018; revised April 20, 2018. This work was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under Grant MOST 106-3114-C-165-001-ES.

Chun-Ta Li is with the Department of Information Management, Tainan University of Technology, Tainan City 71002, Taiwan, R.O.C. (e-mail: th0040@mail.tut.edu.tw).

Chi-Yao Weng is with the Department of Computer Science, Pingtung City 90003, Taiwan, R.O.C. (e-mail: cyweng@mail.nptu.edu.tw).

Chien-Ming Chen is with the School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, P.R. China (e-mail: chienming.taiwan@gmail.com).

Cheng-Chi Lee is with the Department of Library and Information Science, Fu Jen Catholic University, New Taipei City 24205, Taiwan, R.O.C. and with the Department of Photonics and Communication Engineering, Asia University, Taichung City 41354, Taiwan, R.O.C. (e-mail: cclee@mail.fju.edu.tw).

doi: 10.18178/jacn.2018.6.1.245

that their scheme fails to prevent successor impersonation and known key attacks. These design flaws may lead an adversary to get a new group session key and other group members are not aware of having caused problems. So as to overcome the aforementioned flaws of the scheme in [25], in this paper, we design an improved version of Zhu's scheme while keeping the efficiency and preserving the merits of Zhu's scheme.

The remainder of the paper is organized as follows. In Section II, we briefly review Zhu's chaotic maps-based group key agreement scheme. We demonstrate two vulnerabilities of Zhu's scheme in Section III and Section IV, respectively. Our improved scheme and the security analysis of our improved scheme are given in Section V. Finally, we conclude this paper in Section VI.

II. Zhu's Chaotic Maps-Based Group Key Agreement and Privacy Preserving Scheme

In this section, we review Zhu's group key agreement scheme with privacy preserving [25] and the security of their scheme is based on Chebyshev chaotic maps, a pair of symmetric encryption/decryption functions and a chaotic maps-based one-way hash function. Zhu's scheme is divided into five phases: (i) the setup phase, (ii) the authentication and two-party agreement phase, (iii) the broadcast and group key agreement generated phase, (iv) the member revocation phase, and (v) the member join phase. Table I describes the notations used in this paper and the details of each phase are briefly illustrated in the following subsections.

TABLE I: Notations Used in the Paper

Symbol	Description
U_i	The ith participant
ID_i	The identity of U_i
U	A set of protocol participants, where $U = \{U_1, U_2, \ldots, U_n\}$
S_i	U_i's secret key based on Chebyshev chaotic maps
$(x, Ts_i(x))$	U_i's public key based on Chebyshev chaotic maps
r_i	A random number generated by U_i
$E_K[.]/D_K[.]$	A pair of symmetric encryption/decryption functions with key K
$H(.)$	A chaotic maps-based one-way hash function [21]
SK	The session key, which is established between user U_i and user U_{i+1}
GSK	The group session key, which is established between U
$\|\|$	Message concatenation
$\oplus$	A bitwise exclusive OR operation
?=	The comparison operation, judge whether two values are identical or not

A. Setup Phase

In this phase, each U_i maintains its identity ID_i , a chaotic maps-based one-way hash function $H(.)$, a secret key Si and public key $(x, T_{Si}(x))$ based on Chebyshev chaotic maps, a random number generator, and a pair of symmetric encryption/decryption functions $E_K[.]/D_K[.]$ with the key K.

B. Authentication and Two-Party Agreement Phase

In this phase, Zhu's scheme assumes that all group participants $U_1, U_2, \ldots, U_n$ are organized in an ordered chain and U_{i+1} is the successor of U_i . In addition, all ID information and their corresponding public keys have been arranged and all the participants perform the following steps.

Step 1. U_i computes $K_{i,i+1} = Tr_iT_{Si+1}(x)$, $C_i = E_{Ki,i+1}[ID_i \| ID_{i+1} \| Tr_i(x)]$ and $MAC_i = H(ID_i \| ID_{i+1} \| C_i \| H(K_{i,i+1}) \| Tr_i(x))$ and sends $\{C_i , Tr_i(x), MAC_i\}$ to its successor U_{i+1}, where r_i is a random number chosen by U_i and $T_{Si+1}(x)$ is U_{i+1}'s public key.

Step 2. Upon receiving the message from U_i, U_{i+1} computes $K'_{i,i+1} = T_{Si+1}Tr_i(x)$ and reveals ID information by computing $D_{K'i,i+1}[C_i]$. Then U_{i+1} judges whether $H(ID_i \| ID_{i+1} \| C_i \| H(K'_{i,i+1}) \| Tr_i(x))$? = MAC_i. If the equation is equivalent, U_{i+1} computes $K_{i+1,i} = Tr_{i+1}T_{Si}(x)$, $SK = Tr_{i+1}Tr_i(x)$, $C_{i+1} = E_{Ki+1,i}[ID_i \| ID_{i+1} \| Tr_{i+1}(x)]$ and $MAC_{i+1} = H(ID_i \| ID_{i+1} \| C_{i+1} \| Tr_{i+1}(x) \| H(K_{i+1,i}) \| SK)$ and sends $\{C_{i+1}, Tr_{i+1}(x), MAC_{i+1}\}$ to its predecessor U_i, where r_{i+1} is a random number chosen by U_{i+1} and $T_{Si}(x)$ is U_i's public key.

Step 3. Upon receiving the message from U_{i+1}, U_i computes $K'_{i+1,i} = T_{Si}Tr_{i+1}(x)$ and $SK' = Tr_iTr_{i+1}(x)$ and reveals ID information by computing $D_{K'i+1,i}[C_{i+1}]$. Then U_i judges whether $H(ID_i \| ID_{i+1} \| C_{i+1} \| Tr_{i+1}(x) \| H(K'_{i+1,i}) \| SK')$? = MAC_{i+1}. If the equation is not equivalent, the authentication is failed and the session is terminated by U_i. Otherwise, U_i computes $MAC'_i = H(ID_i \| ID_{i+1} \| H(K'_{i+1,i}) \| SK')$ and $SK_{i,i+1} = H(ID_i \| ID_{i+1} \| Tr_iTr_{i+1}(x))$ and takes $SK_{i,i+1}$ as the session key shared between U_i and U_{i+1}. To achieve the property of mutual authentication, U_i sends MAC'_i to user U_{i+1}.

Step 4. Upon receiving MAC'_i from U_i, U_{i+1} judges whether $H(ID_i \| ID_{i+1} \| H(K_{i+1,i}) \| SK)$?= MAC'_i. If not, the authentication is failed and the session is terminated by U_{i+1}. Otherwise, U_{i+1} computes $SK_{i,i+1} = H(ID_i \| ID_{i+1} \| Tr_{i+1}Tr_i(x))$ and takes $SK_{i,i+1}$ as the session key.

Note that the above-mentioned steps can be simultaneous and parallel and each participant can establish two session keys $SK_{i,i+1}$ and $SK_{i-1,i}$ with its successor and predecessor (U_1 establishes $SK_{1,2}$ and $SK_{n,1}$; U_n establishes $SK_{n,1}$ and $SK_{n-1,n}$), respectively.

C. Broadcast and Group Key Agreement Generated Phase

Each group participant U_i computes $X_i = B_{i-1} \oplus B_i = H(SK_{i-1,i}, ID_{session}) \oplus H(SK_{i,i+1}, ID_{session})$ and broadcasts X_i to the group, where $ID_{session}$ is the public ephemeral information that consists of all participants' identities and a nonce. After getting all the X_i , each U_i judges whether $X_1 \oplus X_2 \oplus \ldots, \oplus X_{n-1} \oplus X_n$? = 0. If not, U_i outputs an error symbol $\perp$ and aborts this phase. Otherwise, U_i can use B_i and X_i to get all $B_j (j = 1, \ldots, n)$ by using continuous XOR method. For example, U_1 uses its $B_1 = H(SK_{1,2}, ID_{session})$ to get U_2's $B_2 = H(SK_{2,3}, ID_{session})$ by computing $X_2 \oplus B_1$, where $X_2 = B_1 \oplus B_2$. After getting B_2, U_1 can further use it to get U_3's $B_3 = H(SK_{3,4}, ID_{session})$ by computing $X_3 \oplus B_2$. Finally, after getting all B_j, all group participants can establish the common group session key GSK_i by computing $GSK_i = H(B_1 \| B_2 \| \ldots \| B_n)$, where $GSK_1 = GSK_2 = \ldots = GSK_n$.

D. Member Revocation Phase

In case of a participant U_x leaves the group and the group size changes into $(n-1)$, in order to secure later communications, all remaining participants must update group key and avoid the leaving U_x to know the updated group key. First of all, U_{x-1} and U_{x+1} remove the shared values $SK_{x-1,x}$ and $SK_{x, x+1}$ with U_x and U_{x+1} becomes the new successor of U_{x-1}. Then, U_{x-1} needs to generate a new message $\{C_{x-1}, Tr_{x-1}(x), MAC_{x-1}\}$ and sends it to its new successor U_{x+1}.

Upon receiving $\{C_{x\text{-}1}, Tr_{x-1}(x), MAC_{x-1}\}$ from its new predecessor U_{x-1}, U_{x+1} verifies the validity of the message $\{C_{x\text{-}1}, Tr_{x-1}(x), MAC_{x-1}\}$ and agrees the new secret $SK_{x\text{-}1,\ x+1}$ shared between U_{x+1} and $U_{x\text{-}1}$. At the same time, U_{x+1} generate a new message $\{C_{x+1}, Tr_{x+1}(x), MAC_{x+1}\}$ and sends it to its new predecessor U_{x+1}. Then $U_{x\text{-}1}$ verifies the validity of the message $\{C_{x+1}, Tr_{x+1}(x), MAC_{x+1}\}$ and gets the new secret $SK_{x\text{-}1,\ x+1}$ shared between $U_{x\text{-}1}$ and U_{x+1}. Finally, each participant U_j that follows U_x changes its index to $(j-1)$ and all the existing $(n\text{-}1)$ participants can get a new group session key by recomputing the protocol of Section II.*C*.

E. Member Join Phase

In case of a new participant is authorized to join the group of which size is n, the new participant U_{n+1} becomes the successor of participant U_n and the participant U_1 becomes the successor of participant U_{n+1}. First of all, U_n needs to send a new message $\{C_n, Tr_n(x), MAC_n\}$ to its new successor U_{n+1} while U_{n+1} sends message $\{C_{n+1}, Tr_{n+1}(x), MAC_{n+1}\}$ to its new successor U_1. Then U_{n+1} verifies the validity of the message $\{C_n, Tr_n(x), MAC_n\}$ and computes the new secret $SK_{n,n+1}$ shared between U_{n+1} and its new predecessor U_n. Similarly, the first participant U_1 updates its new secret with $SK_{n+1,1}$. Finally, all the $(n+1)$ participants in the group can get a new group session key by recomputing the protocol of Section II.*C*.

III. Successor Impersonation Attack on Zhu's Group Key Agreement and Privacy Preserving Scheme

In this section, we found Zhu's scheme is insecure against successor impersonation attack in the authentication and two-party agreement phase and this design flaw can lead an adversary U_A to impersonate as a legitimate successor to establish a two-party agreement key $SK_{i,i+1}$ with a victim predecessor. We further provide the detailed explanation of this attack through the following steps:

Step 1. In Step 1 of authentication and two-party agreement phase of Zhu's scheme, the participant U_i sends the message $\{C_i, Tr_i(x), MAC_i\}$ to its successor U_{i+1} and U_A intercepts this message to prevent it arrives U_{i+1}.

Step 2. U_A generates a random number r_a and computes $K^*_{i+1,i} = Tr_aTs_i(x)$, $SK^* = Tr_aTr_i(x)$, $C^*_{i+1} = E_{K^*i+1,i}[ID_i||ID_{i+1}||Tr_a(x)]$ and $MAC^*_{i+1} = H(ID_i||ID_{i+1}||C^*_{i+1}||Tr_a(x)||H(K^*_{i+1,i})||SK^*)$ and sends $\{C^*_{i+1}, Tr_a(x), MAC^*_{i+1}\}$ to the victim predecessor U_i, where $Ts_i(x)$ is U_i's public key.

Step 3. In Step 3 of authentication and two-party agreement phase of Zhu's scheme, U_i computes $K^*_{i+1,i} = Ts_iTr_a(x)$ and $SK^*_i = Tr_iTr_a(x)$ and reveals ID information by computing $D_{K^*i+1,i}[C^*_{i+1}]$. Then U_i judges whether $H(ID_i||ID_{i+1}||C^*_{i+1}||Tr_a(x)||H(K^*_{i+1,i})||SK^*_i)$? $= MAC^*_{i+1}$. If the equation is equivalent, U_i believes that the message is generated by U_{i+1}. Then U_i further computes $MAC^*_i = H(ID_i||ID_{i+1}||H(K^*_{i+1,i})||SK^*_i)$ and $SK^*_{i,i+1} = H(ID_i||ID_{i+1}||Tr_iTr_a(x))$ and takes $SK^*_{i,i+1}$ as the session key. Finally, U_i sends MAC^*_i to its successor U_{i+1} and U_A intercepts this acknowledgement message to prevent it arrives U_{i+1}. Note that U_A can compute the session key $SK^*_{i,i+1} = H(ID_i||ID_{i+1}||Tr_aTr_i(x))$ after intercepting $Tr_i(x)$.

From the above-mentioned steps show, the adversary U_A can not only successfully impersonate as the legal successor after sending the message $\{C^*_{i+1}, Tr_a(x), MAC^*_{i+1}\}$ from the intercepted message $\{C_i, Tr_i(x), MAC_i\}$ sent by U_i to U_{i+1} during authentication and two-party agreement phase but also establish a common session key $SK^*_{i,i+1}$ agreed with the victim predecessor U_i. Thus Zhu's scheme fails to prevent this kind of impersonation attack. For clarity, the details of this attack are given in Fig. 1.

In the broadcast and group key agreement generated phase of Zhu's scheme, each group participant U_i (including U_A) broadcasts its $X_i = B_{i-1} \oplus B_i$ to the group. Then U_A can use its B_i and X_i to derive all B_j ($j = 1, \ldots, n$) by using continuous XOR method. Finally U_A can establish the common group session key GSK_i and other participants are not aware of having caused problem because the property of mutual authentication between predecessor and successor is broke down.

IV. Known Key Attack on Zhu's Group Key Agreement and Privacy Preserving Scheme

Given two participants U_i and U_{i+1}, if one of their previous instances are known to an adversary U_A, the previous and future session keys can be easily derived by U_A. We further provide the detailed explanation of this attack through the following steps:

Step 1. In Step 2 of authentication and two-party agreement phase of Zhu's scheme, if two used parameters $SK = Tr_{i+1}Tr_i(x) = Tr_iTr_{i+1}(x) = SK'$ and $ID_{session}$ are compromised by U_A, U_A can retrieve them to compute the session key $SK_{i,i+1} = H(ID_i||ID_{i+1}||Tr_iTr_{i+1}(x))$ and the value of $B_i = H(SK_{i\text{-}1,i}, ID_{session})$.

Step 2. Then U_A can collect all broadcasting messages X_i ($i = 1, \ldots, n$) in broadcast and group key agreement generated phase.

Step 3. After collecting all the X_i, U_A can retrieve B_i and Xi to derive all B_j ($j = 1, \ldots, n$) by using continuous XOR method. Finally, after deriving all B_j, U_A can easily establish the common group session key GSK_i by computing $GSK_i = H(B_1||B_2||\ldots||B_n)$.

In case of a member revocation or a member join, U_A only needs to update the value of $B_i = H(SK_{i,i+1}, ID^{new}_{session})$ and re-computes the protocol of Section II.*C* to get a new group session key, where $ID^{new}_{session}$ is current group information that consists of active participants' identities.

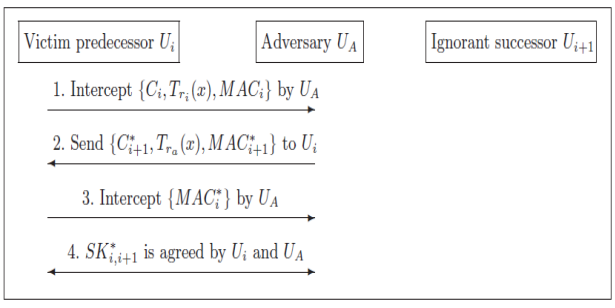

Fig. 1. Successor impersonation attack on Zhu's scheme.

V. Our Improved Scheme

In order to overcome the identified security imperfections of Zhu's group key agreement scheme with privacy preserving, in this section, we suggest an improved scheme for preventing successor/predecessor impersonation attacks. The improved scheme can be described in the following phases.

A. Setup Phase

In this phase, the executed steps are the same as in Zhu's scheme.

B. Authentication and Two-Party Agreement Phase

In this phase, Zhu's scheme assumes that all group participants U_1, U_2, …, U_n are organized in an ordered chain and U_{i+1} is the successor of U_i. In addition, all ID information and their corresponding public keys have been arranged and all the participants perform the following steps.

Step 1. U_i computes $K_{i,i+1} = Tr_iT_{Si+1}(x)$, $C_i = E_{Ki,i+1}[ID_i \| ID_{i+1} \| Tr_i(x)]$ and $MAC_i = H(ID_i \| ID_{i+1} \| C_i \| H(K_{i,i+1}) \| Tr_i(x))$ and sends $\{C_i , Tr_i(x), MAC_i\}$ to its successor U_{i+1}, where r_i is a random number chosen by U_i and $T_{Si+1}(x)$ is U_{i+1}'s public key.

Step 2. Upon receiving the message from U_i, U_{i+1} computes $K'_{i,i+1} = T_{Si+1}Tr_i(x)$ and reveals ID information by computing $D_{K'i,i+1}[C_i]$. Then U_{i+1} judges whether $H(ID_i \| ID_{i+1} \| C_i \| H(K'_{i,i+1}) \| Tr_i(x))$? = MAC_i. If the equation is equivalent, U_{i+1} computes $K_{i+1,i} = Tr_{i+1}T_{Si}(x)$, $SK = Tr_{i+1}Tr_i(x)$, $C_{i+1} = E_{Ki+1,i}[ID_i \| ID_{i+1} \| Tr_{i+1}(x)]$ and $MAC_{i+1} = H(ID_i \| ID_{i+1} \| T_{Si+1}Tr_i(x) \| H(K_{i+1,i}) \| SK)$ and sends $\{C_{i+1}, Tr_{i+1}(x), MAC_{i+1}\}$ to its predecessor U_i, where r_{i+1} is a random number chosen by U_{i+1} and $T_{Si}(x)$ is U_i's public key.

Step 3. Upon receiving the message from U_{i+1}, U_i computes $K'_{i+1,i} = T_{Si}Tr_{i+1}(x)$ and $SK' = Tr_iTr_{i+1}(x)$ and reveals ID information by computing $D_{K'i+1,i}[C_{i+1}]$. Then U_i computes $Tr_iT_{Si+1}(x)$ and judges whether $H(ID_i \| ID_{i+1} \| Tr_iT_{Si+1}(x) \| H(K'_{i+1,i}) \| SK')$? = MAC_{i+1}. If the equation is not equivalent, the authentication is failed and the session is terminated by U_i. Otherwise, U_{i+1} is authenticated by U_i. Next, U_i computes $MAC'_i = H(ID_i \| ID_{i+1} \| H(K'_{i+1,i}) \| SK')$ and $SK_{i,i+1} = H(ID_i \| ID_{i+1} \| Tr_iTr_{i+1}(x))$ and takes $SK_{i,i+1}$ as the session key shared between U_i and U_{i+1}. To achieve the property of mutual authentication, U_i sends the acknowledgement message MAC'_i to user U_{i+1}.

Step 4. Upon receiving MAC'_i from U_i, U_{i+1} judges whether $H(ID_i \| ID_{i+1} \| H(K_{i+1,i}) \| SK)$?= MAC'_i. If not, the authentication is failed and the session is terminated by U_{i+1}. Otherwise, U_i is authenticated by U_{i+1}. Finally, U_{i+1} computes $SK_{i,i+1} = H(ID_i \| ID_{i+1} \| Tr_{i+1}Tr_i(x))$ and takes $SK_{i,i+1}$ as the session key.

C. Broadcast and Group Key Agreement Generated Phase

Each group participant U_i computes $X_i = B_{i-1} \oplus B_i = H(SK_{i-1,i}, Ts_iTs_{i-1}(x), ID_{session}) \oplus H(SK_{i,i+1}, Ts_iTs_{i+1}(x), ID_{session})$ and broadcasts X_i to the group, where $Ts_{i-1}(x)$ is the public key of U_i's successor and $ID_{session}$ is the public ephemeral information that consists of all participants' identities and a nonce. To sum it up, we can see the value of B_i in Table II. After getting all the X_i , each U_i judges whether $X_1 \oplus X_2 \oplus \ldots, X_{n-1} \oplus X_n$? = 0. If not, U_i outputs an error symbol $\perp$ and aborts this phase. Otherwise, U_i can use B_i and X_i to get all B_j (j = 1, …, n) by using continuous XOR method. For example, U_1 uses its $B_1 = H(SK_{1,2}, Ts_1Ts_2(x), ID_{session})$ to get U_2's $B_2 = H(SK_{2,3}, Ts_2Ts_3(x), ID_{session})$ by computing $X_2 \oplus B_1$, where $X_2 = B_1 \oplus B_2$. After getting B_2, U_1 can further use it to get U_3's $B_3 = H(SK_{3,4}, Ts_3Ts_4(x), ID_{session})$ by computing $X_3 \oplus B_2$. Finally, after getting all B_j, all group participants can establish the common group session key GSK_i by computing $GSK_i = H(B_1 \| B_2 \| \ldots \| B_n)$, where $GSK_1 = GSK_2 = \ldots = GSK_n$.

TABLE II: The Value of B_I

Parameter	Value
B_1	$H(SK_{1,2}, Ts_1Ts_2(x), ID_{session})$
…	…
B_i	$H(SK_{i,i+1}, Ts_iTs_{i+1}(x), ID_{session})$
…	…
B_n	$H(SK_{n,1}, Ts_nTs_1(x), ID_{session})$

D. Member Revocation Phase

In this phase, the executed steps are the same as in Zhu's scheme.

E. Member Join Phase

In this phase, the executed steps are the same as in Zhu's scheme.

F. Security Analysis of Our Improved Scheme

As shown in Section V.*B*, we can see that our improved scheme is similar to the original Zhu's scheme. The two differences are the Step 2 and Step 3 of authentication and two-party agreement phase. Therefore, our improved scheme inherits the auxiliary functions of Zhu's scheme. Here, we just analyze why our improved scheme can resist our proposed attack in Section III. Step 1 is the same step as in Section III. The Step 2 is described as follows:

Remark 1. *In order to put emphasis on describing the security of our improved scheme, we assume that each participant's secret key S_i has been well-protected by himself/herself.*

Step 2. U_A generates a random number r_a and computes $K^*_{i+1,i} = Tr_aTs_i(x)$, $SK^* = Tr_aTr_i(x)$, $C^*_{i+1} = E_{K^*i+1,i}[ID_i \| ID_{i+1} \| Tr_a(x)]$. However, in computation of MAC^*_{i+1}, the adversary U_A cannot generate the parameter $Ts_{i+1}Tr_i(x)$ because U_A do not possess U_{i+1}'s secret key S_{i+1}. As a result, U_A is impossible to response a valid $MAC^*_{i+1} = H(ID_i \| ID_{i+1} \| T_{Si+1}Tr_i(x) \| H(K^*_{i+1,i}) \| SK^*)$ to pass U_i's verification in Step 3 of our attack. That is to say, our improved scheme is secure against successor impersonation attack mentioned in Section 3.

Similarly, in Step 3 of our improved scheme, it is impossible for the adversary U_A to launch predecessor impersonation attack because U_A do not possess U_i's secret key S_i, so he/she cannot compute a valid acknowledgement $MAC_i' = H(ID_i \| ID_{i+1} \| T_{Si}Tr_a(x) \| \| SK')$ to pass U_{i+1}'s verification in Step 4 of our improved scheme.

On the other hand, we further analyze why our improved scheme can resist our proposed attack in Section IV. We assume two used parameters $SK = Tr_{i+1}Tr_i(x) = Tr_iTr_{i+1}(x) = SK'$ and $ID_{session}$ are compromised by U_A, he/she still cannot impact on previous or future session keys. In our improved scheme, knowing $SK_{i,i+1}$, $Ts_{i+1}(x)$ and $ID_{session}$ are useless for U_A to compute $B_i = H(SK_{i,i+1}, Ts_iTs_{i+1}(x), ID_{session})$, since U_A is

unable to calculate the parameter $Ts_iTs_{i+1}(x)$ without knowing U_i's secret key S_i. Without having B_i, U_A cannot re-compute the protocol of Section V.*C* to derive the common group session key GSK_i. As a result, the improved scheme can prevent known key attack.

In addition, the computation overhead of authentication and two-party agreement phase of our improved scheme is almost the same as Zhu's scheme because we only add a chaotic maps operation in MAC_{i+1} and replace Zhu's $MAC_{i+1} = H(ID_i||ID_{i+1}||C_{i+1}||Tr_{i+1}(x)||H(K_{i+1,i})||SK)$ with our $MAC_{i+1} = H(ID_i||ID_{i+1}||T_{Si+1}Tr_i(x)||H(K_{i+1,i})||SK)$. Moreover, the computation overhead of broadcast and group key agreement generated phase of our improved scheme is almost the same as in Zhu's scheme because we only add two chaotic maps operations in B_{i-1} and B_i and replace Zhu's $B_{i-1} = H(SK_{i-1,i}, ID_{session})$ and $B_i = H(SK_{i,i+1}, ID_{session})$ with our $B_{i-1} = H(SK_{i-1,i}, Ts_iTs_{i-1}(x), ID_{session})$ and $B_i = H(SK_{i,i+1}, Ts_iTs_{i+1}(x), ID_{session})$, respectively. Therefore, the efficiency of our improved scheme is similar to Zhu's scheme.

VI. Conclusions

In this paper, we have first reviewed Zhu's group key agreement and privacy preserving scheme and shown that the process of authentication and two-party agreement phase is insecure, that is, an adversary can maliciously intercept the transmitted message and generate an intentional response message to impersonate as the legitimate successor. In addition, we have demonstrated that this design flaw may cause the victim predecessor into establishing a common session key with the adversary and damage the security of group key in group key agreement generated phase. To avoid the security problem on Zhu's scheme, we have proposed security improvements which not only repair the design flaw of Zhu's scheme but also inherit the merits and efficiencies of their scheme.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This research was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C., under contract no.: MOST 106-3114- C-165-001-ES.

References

[1] P. Bergamo, P. Arco, A. Santis, and L. Kocarev, "Security of public-key cryptosystems based on Chebyshev polynomials," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 7, pp. 1382-1393, 2005.

[2] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*, Chapman & Hall, CRC Press, 2003.

[3] X. Wang and J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 12, pp. 4052-4057, 2010.

[4] D. Xiao, F. Y. Shih, and X. Liao, "A chaos-based hash function with both modification detection and localization capabilities," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2254-2261, 2010.

[5] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos, Solitons & Fractals*, vol. 37, no. 3, pp. 669-674, 2008.

[6] C. T. Li, C. W. Lee, and J. J. Shen, "An extended chaotic maps based keyword search scheme over encrypted data resist outside and inside keyword guessing attacks in cloud storage services," *Nonlinear Dynamics*, vol. 80, no. 3, pp. 1601-1611, 2015.

[7] C. T. Li, C. C. Lee, and C. Y. Weng, "A secure cloud-assisted wireless body area network in mobile emergency medical care system," *Journal of Medical Systems*, vol. 40, no. 5, pp. 1-15, 2016.

[8] S. Kumari, X. Li, F. Wu, A. K. Das, H. Arshad, and M. K. Khan, "A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps," *Future Generation Computer Systems*, vol. 63, pp. 56-75, 2016.

[9] C. C. Lee, D. C. Lou, C. T. Li, and C. W. Hsu, "An extended chaotic-maps-based protocol with key agreement for multiserver environments," *Nonlinear Dynamics*, vol. 76, no. 1, pp. 853-866, 2014.

[10] C. T. Li, "A secure chaotic maps-based privacy-protection scheme for multi-server environments," *Security and Communication Networks*, vol. 9, no. 14, pp. 2276-2290, 2016.

[11] C. T. Li, C. C. Lee, and C. Y. Weng, "An extended chaotic maps based user authentication and privacy preserving scheme against DoS attacks in pervasive and ubiquitous computing environments," *Nonlinear Dynamics*, vol. 74, no. 4, pp. 1133-1143, 2013.

[12] X. Hao, J. Wang, Q. Yang, X. Yan, and P. Li, "A chaotic map-based authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 37, no. 2, p. 9919, 2013.

[13] T. F. Lee, "An efficient chaotic map-based authentication and key agreement scheme using smartcards for telecare medicine information systems," *Journal of Medical Systems*, vol. 37, no. 6, p. 9985, 2013.

[14] C. T. Li, C. C. Lee, and C. Y. Weng, "A secure chaotic maps and smart cards based password authentication and key agreement scheme with user anonymity for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, no. 9, pp. 1-11, 2014.

[15] E. Bresson, O. Chevassut, and D. Pointcheval, "Group diffie-hellman key exchange secure against dictionary attacks," *Lecture Notes in Computer Science*, vol. 2501, pp. 497-514, 2002.

[16] A. T. Sherman and D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444-458, 2003.

[17] S. M. Lee, J. Y. Hwang, and D. H. Lee, "Efficient password-based group key exchange," *Lecture Notes in Computer Science*, vol. 3184, pp. 191-199, 2004.

[18] M. Abdalla, E. Bresson, O. Chevassut, and D. Pointcheval, "Password-based hroup key exchange in a constant number of rounds," *Lecture Notes in Computer Science*, vol. 3958, pp. 427-442, 2006.

[19] R. Dutta and R. Barua, "Password-based encrypted group key agreement," *International Journal of Network Security*, vol. 3, no. 1, pp. 23-34, 2006.

[20] L. Xu and C. Huang, "Computation-efficient multicast key distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 5, pp. 577-587, 2008.

[21] M. H. Zheng, H. H. Zhou, J. Li, and G. H. Cui, "Efficient and provably secure password-based group key agreement protocol," *Computer Standards & Interfaces*, vol. 31, no. 5, pp. 948-953, 2009.

[22] D. H. Je, J. S. Lee, Y. Park, and S. W. Seo, "Computation-and-storage-efficient key tree management protocol for secure multicast communications," *Computer Communications*, vol. 33, no. 2, pp. 136-148, 2010.

[23] P. Vijayakumar, S. Bose, and A. Kannan, "Centralized key distribution protocol using the greatest common divisor method," *Computers & Mathematics with Applications*, vol. 65, no. 9, pp. 1360-1368, 2013.

[24] P. Vijayakumar, S. Bose, and A. Kannan, "Chinese remainder theorem based centralised group key management for secure multicast communication," *IET Information Security*, vol. 8, no. 3, pp. 179-187, 2014.

[25] H. Zhu, "Secure chaotic maps-based group key agreement scheme with privacy preserving," *International Journal of Network Security*, vol. 18, no. 6, pp. 1001-1009, 2016.

Chun-Ta Li received the Ph.D degree in computer science and engineering from National Chung Hsing University, Taiwan, in 2008. He is currently an associate professor with the Department of Information Management at Tainan University of Technology, Taiwan. His research interests include information and network security, wireless sensor networks, mobile computing, and security protocols for RFID, IoTs and cloud computing.

Chi-Yao Weng received the Ph.D degree in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 2011. He is currently an assistant professor with the Department of Computer Science at National Pingtung University. His current research interests include data analysis, mobile security, multimedia security, digital right management and information forensics.

Chien-Ming Chen is currently an associate professor with the School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, China. His research interests include network security, mobile Internet, wireless sensor network and cryptography.

Cheng-Chi Lee received the Ph.D degree in computer science from National Chung Hsing University (NCHU), Taiwan, in 2007. He is currently a professor with the Department of Library and Information Science at Fu Jen Catholic University. His current research interests include data security, cryptography, network security, mobile communications and computing, wireless communications.