IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 5, 2018, accepted October 28, 2018, date of publication November 9, 2018, date of current version November 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2879271

A Provably Secure Group Key Agreement Scheme With Privacy Preservation for Online Social Networks Using Extended Chaotic Maps

CHUN-TA LI[1], TSU-YANG WU[2,3], AND CHIEN-MING CHEN[2]

[1]Department of Information Management, Tainan University of Technology, Tainan 71002, Taiwan
[2]College of Computer Science and Engineering, Shandong University of Science and Technology, Shandong 266510, China
[3]Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350108, China

Corresponding author: Chien-Ming Chen (chienming.taiwan@gmail.com)

This work was supported by the Ministry of Science and Technology, Taiwan, under Contract MOST 107-2410-H-165-001.

ABSTRACT With the rapid growth in the popularity of mobile devices and development of network technologies, various online social networking applications have grown in popularity. While these social networking sites provide benefits in terms of enhanced connectivity with people all around the world, they can also pose security threats and raise privacy concerns due to their vulnerability to be exploited by malicious agents. Such social networking sites where members can transmit or share their social information via public channels increase the risk that these information may be exposed to unwanted users. Therefore, it is important to enhance the security of these online social network services using group session keys. These group session keys also need to be updated in case a new member joins the group or an old member leaves the social group. To enhance the trustworthiness of the online social network systems, in this paper, we propose a secure chaotic maps-based group key agreement scheme. In this proposed scheme, we also provide member anonymity to ensure the privacy of the communication between the social networking platform and the members. The proposed solution does not rely on a centralized online key center or a trusted group chairman, thus ensuring fairness. We integrate the mechanisms of message encryption and member verification into the scheme to allow the members to anonymously interact with the services of the online social network. We verify the formal security of the proposed solution using the widely accepted BAN logic analysis and simulation verification with Proverif to prove that our scheme is secure against both passive and active attacks. We also demonstrate that the proposed scheme is efficient in its implementation and achieves greater functionality criteria in comparison with similar existing proposals.

INDEX TERMS Group key agreement, extended chaotic maps, online social networks, privacy preserving, social member.

I. INTRODUCTION

The fast growing popularity of Internet technologies has opened up new ways for people to connect. This has led to the growth of online social networks (OSNs) which aim to create and foster social connections between people and their interests. As shown in Fig. 1, we are surrounded by different online social networks which serve different purposes and target different user groups. Social networking sites such as BeeTalk [3] and Paktor [27] are used exclusively for dating while LINE [23] and WeChat [36] are popular for making friends and staying connected. Social networks such as Facebook [9], Twitter [31] and Google+ [10] while mostly are used for sharing life-experiences, have also started affecting how people and organizations do business.

Online social networks are ever-growing in popularity and provide several useful functionalities such as the ability to exchange messages, share content, and disseminate information among social users. However, often they contain people's personal profiles (including addresses, photographs) and sensitive information. This data must be protected to prevent it's access by unwanted entities. There is also the external threat of security attacks on these online social networks which can compromise their user data. Therefore, preservation of privacy while disseminating social network data is an important and urgent concern which needs to be addressed.

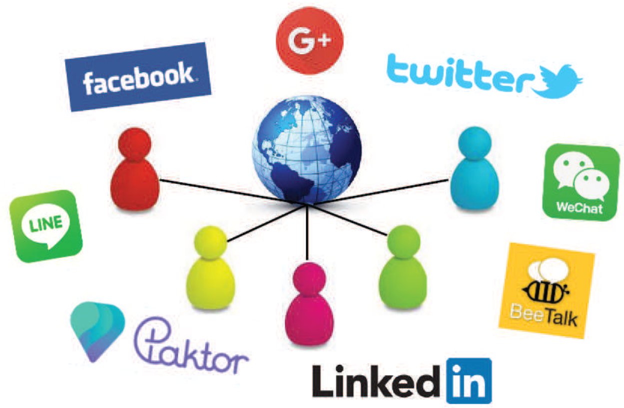

FIGURE 1. Different online social networks.

Since online social networks contain and transmit user's personal and sensitive data, encryption of this data and authentication of it's members are crucial needs of OSNs.

Technologies such as group key agreement are helpful for establishing common session keys to protect sensitive social data being shared in specific groups from being spied on by illegitimate social members or external entities. In this paper, we develop a secure group key agreement scheme using extended chaotic maps, where only participating members of a social group can construct the group session key without the help of a trusted key center or a centralized key distributor. Similar to existing group communication systems, this proposed scheme for online social networks is designed to fulfill certain security goals.

A. SECURITY GOALS OF ONLINE SOCIAL NETWORKS

In this subsection, we briefly describe the essential criteria that a secure online social network system should satisfy. The requirements are as follows:

- Member verification: On a secure online social networking system, every member should be able to confirm that all participants in a social group have been securely verified. This is to ensure that only legitimate members of the social networking group can send messages to other members of the social group.
- Group identification: In real social networking systems, a member may join different online social groups. Therefore, it is essential that a member is able to clearly identify the social group from where the received message originated.
- Privacy preservation: A secure social networking system must ensure that the real identities of it's social members are masked. Even if a malicious attacker is able to eavesdrop on the messages being transmitted between a member and a group, the attacker should not be able to connect the transmitted messages to the real identity of the member.
- Security in social group communication: To ensure the privacy and security of the communication within a social group, there is a need to construct a common group session key shared among all legitimate members of the social group. This should be done without the help of any third party and the key should be kept secret from all external unwanted parties.
- Fairness in group key establishment: For a group key establishment scheme, no single entity should control the generation of the group session key. To ensure the fairness in the process of establishment of the group session key, the final group session key should contain equal contributions from all the members in the social group.
- Dynamic group key management: When a new social member joins a social group or an old social member leaves the social group, the group key should be updated securely. The mechanism to generate the new key should prevent new members from accessing previous data and old members from accessing future data of the social group.
- System efficiency: A secure social network system would be called efficient if it has a low communication overhead and computational complexity during the group key establishment phase.

B. RELATED WORKS FOR GROUP KEY ESTABLISHMENT

Group key establishment is the process which enables a group of members to establish a common group key. This is helpful for sending and exchanging sensitive messages among all legitimate members of the group. In general there are two types of group key establishment schemes:

1) group key distribution scheme [17], [25], [33] and
2) group key agreement scheme [8], [13]–[15], [32], [34], [41]–[43].

In the *group key distribution scheme*, a trusted key center or a centralized key distributor is responsible for generating the group key and securely distributing it to all the group members. In the *group group key agreement scheme*, the participants establish the group key without the help of a trusted key center or a centralized key distributor. In recent years, several centralized group key management schemes [1], [17], [18], [25], [28]–[30], [33], [35], [37], [39] have been proposed in the literature. In 2000, Steiner et al. [30] proposed a key management scheme for dynamic peer groups based on the Diffie-Hellman key exchange scheme. However, Steiner *et al.*'s scheme [30] is not suitable for large groups. Subsequently, Wong *et al.* [37] proposed a multicast key management system based on key graphs to solve the scalability problem of Steiner *et al.* However, their scheme suffered from high computational costs. In 2003, Sherman and McGrew proposed a key establishment scheme [28] based on one-way function trees for handling large dynamic groups. In 2006, Wang and Laih used a technique called merging to propose a timebound hierarchical key assignment scheme [35], which greatly reduced the communication and storage requirements of Wang *et al.*'s scheme. In 2008, Xu and Huang [39] proposed a multicast key distribution scheme which used maximum distance separable codes to reduce the computational complexity of the key generation process. In 2010, Je *et al.* [18] proposed an efficient key tree management scheme which examined the resource information of each group member's device to reduce the computational complexity and storage costs of their scheme.

In 2013 and 2016, Lou *et al.* [25] and Jaiswal and Tripathi [17] individually proposed different implementations of the group key distribution schemes based on elliptic curve cryptography (ECC) [20]. Both their implementations involved selecting a chairman or an initiator to distribute information related to the group key to all other participants of the group. In 2014, Vijayakumar *et al.* [33] further proposed a new centralized group key management based on the Chinese remainder theorem called CRTGKM algorithm. At the server-side, the computational complexity of this algorithm is $O(1)$ for the case when a member joins or leaves the multicast group. At the group member side, this computational complexity is minimized and a multicast group member performs only a single modulo division operation.

There are several other works of research which have focussed on how to design authenticated group key agreement schemes [8], [41]–[43]. In 2006, Dutta and Barua [8] proposed a password-based encrypted group key agreement scheme. The security of their scheme is based on the computational Diffie-Hellman assumption and a cryptographically secure one-way hash function. In 2009, Zheng *et al.* [42] proposed an efficient password-based group key agreement scheme with resistance to dictionary attacks. Their scheme has proven to be secure against online and off-line dictionary attacks under the decisional Diffie-Hellman assumption for both the ideal cipher model and the random oracle model. In 2010, Zhao *et al.* [41] proposed an efficient fault-tolerant group key agreement scheme which not only assures that all illegitimate participants are excluded from the group but also ensures efficiency when compared with other fault-tolerant group key agreement schemes. More recently, in 2016, Zhu introduced a multi-party authenticated group key agreement and privacy preserving scheme [43] based on the Chebyshev chaotic maps [26] and a pair of secure symmetric encryption and decryption functions. Zhu's scheme is secure against various security attacks such as replay, impersonation, man-in-the-middle and key compromise. Moreover, Zhu extended the proposed scheme to high level security attributes such as privacy preservation, mutual and group authentication, perfect forward secrecy and fairness in group key establishment.

C. MAIN CONTRIBUTIONS

In this paper, we put forward a secure and efficient group key agreement and privacy preservation scheme based on extended chaotic maps. The main contributions of this paper are listed below:

- Our proposed solution does not need to adopt a centralized online key center or a group chairman to distribute the group session key. We also suggest a fairness mechanism for balancing the computation and communication overheads among the members.
- Once a member leaves the social group or a new member joins the social group, all legal members of the group can prove the validity of the group session key with only a single communication round for each session.
- In practical applications of online social networks, a social member may join multiple social groups. Our proposed scheme introduces a group identification method for enhancing the efficiency of identification of the online social network system.
- To the best of our knowledge, this work is the first attempt to provide a chaotic maps-based group key agreement scheme for online social networks. The proposed scheme is provably secure in the BAN logic model under the chaotic-based decisional Diffie-Hellman (CDDH) and the chaotic-based computational Diffie-Hellman (CCDH) problems.

D. ORGANIZATION OF THE PAPER

The remainder of the paper is organized as follows. In Section 2, we present some preliminaries of this paper. In Section 3 we propose the new group key agreement and privacy preservation scheme for online social network systems. The new, informal and formal security proofs of the proposed scheme are presented in Sections 4 and 5, respectively. In Section 6, we present the performance analysis and functionality comparisons of the proposed scheme with other related schemes. Finally, we present our conclusions in Section 7.

II. PRELIMINARIES

Before proposing the chaotic map scheme, in this section we introduce the concepts of the Chebyshev chaotic maps [4], [26], and the decisional discrete logarithm (DDL) and the decisional Diffie-Hellman (DDH) problems used in our proposed scheme. The Chebyshev polynomial $T_n(x)$ is a polynomial in x of degree n, where x is a variable with value lying in the interval [-1, 1] and n is an integer. The Chebyshev polynomial $T_n(x) : [-1, 1] \to [-1, 1]$ is defined as

$$T_n(x) = \cos(n \cdot \arccos(x)) \tag{1}$$

and the recurrence relation of Chebyshev polynomial is defined as:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), n \geq 2 \tag{2}$$

where $T_0(x) = 1$, $T_1(x) = x$ and $\cos(x)$ and $\arccos(x)$ are the trigonometric functions. They are defined as $\cos : R \to [-1, 1]$ and $\arccos : [-1, 1] \to [0, \pi]$.

We give some examples of Chebyshev polynomials as follows:

$$T_1(x) = x,$$
$$T_2(x) = 2x^2 - 1,$$
$$T_3(x) = 4x^3 - 3x,$$

$$T_4(x) = 8x^4 - 8x^2 + 1,$$
$$T_5(x) = 16x^5 - 20x^3 + 5x.$$

The Chebyshev polynomials have two important properties [11], [21], [24], namely: the semigroup property and the chaotic property.

(1) The semigroup property:

$$\begin{aligned} T_r(T_s(x)) &= \cos(r\ \cos^{-1}(\cos(s\ \cos^{-1}(x)))) \\ &= \cos(rs\ \cos^{-1}(x)) \\ &= T_{sr}(x) \\ &= T_s(T_r(x)), \end{aligned}$$

where r and s are two positive integers and $x \in [-1, 1]$.

(2) The chaotic property:
When the degree $n > 1$, the Chebyshev polynomial map $T_n(x) : [-1, 1] \to [-1, 1]$ of degree n is a chaotic map with invariant density $f^*(x) = 1/(\pi\sqrt{1-x^2})$ for the positive Lyapunov exponent $\lambda = \ln n > 0$.

Zhang [40] proved that in the interval $(-\infty, +\infty)$, the semigroup property holds for Chebyshev polynomials. This can enhance the property as follows:

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) \bmod p,$$

where $n \geq 2$, $x \in (-\infty, +\infty)$ and p is a large prime. We can state that $T_r(T_s(x)) \equiv T_{sr}(x) \equiv T_s(T_r(x)) \bmod p$. The semigroup property also holds in this case, and the enhanced Chebyshev polynomials still satisfy the commutative property under composition.

Based on two well-known Diffie-Hellman problems which are assumed to be difficult to solve within polynomial time, the Chebyshev polynomial also meets the following problems [7], [12], [22], [43].

(1) The chaotic maps-based discrete logarithm problem (CMBDLP):
Given two elements x and y, it is intractable to find the integer r, such that $T_r(x) \bmod p = y$.

(2) The chaotic maps-based Diffie-Hellman problem (CMBDHP):
Given three elements x, $T_r(x) \bmod p$ and $T_s(x) \bmod p$, it is intractable to compute the value $T_{rs}(x) \bmod p$.

III. THE PROPOSED SCHEME

We assume that a set of n members $M = \{M_1, M_2, \ldots, M_n\}$ of a social group G_i wants to securely communicate and share sensitive data in the group. Each member M_i needs to maintain a private/public key pair (S_i, PK_i) such that $PK_i = T_{S_i}(x) \bmod p$, where p is a large prime number and $x \in Z_p$ is a random number generated and published by a trusted authority TA. The long term private/public key pair of M_i is authenticated by the TA with it's corresponding certificate. Additionally, each member M_i needs to maintain his or her identity ID_i, a chaotic maps-based one-way hash function $H(.)$, a random number generator, and a pair of symmetric encryption/decryption functions $E_K[.]/D_K[.]$ with the key K. The notations used throughout the paper are shown in Table 1.

The proposed scheme consists of five phases, namely: A) the social group establishment phase, B) the predecessor and successor authentication phase, C) the group key agreement phase, D) the social member withdrawal phase, E) and social member joining phase. We describe the details of these phases as below:

A. SOCIAL GROUP ESTABLISHMENT PHASE

Let us suppose that the initiator of the social group would like to invite a set of n members to participate in the specific social group. He/she submits a request for the establishment of the social group with a list of the identities of all the participants to the OSN via a secure channel. Here, we assume that $M = \{M_1, M_2, \ldots, M_n\}$ is the set of n members in that particular social group and all social members $M_i = M_1, M_2, \ldots, M_n$ are organized as an ordered chain, and M_1 is the initiator of the social group. After receiving the request from M_1, the OSN creates a social group and generates a unique identity GID_i for M. Then the OSN multicasts the $\{GID_i, ID_{member}\}$ to M via a secure channel

B. PREDECESSOR AND SUCCESSOR AUTHENTICATION PHASE

In this phase, we define the M_{i-1} as the predecessor of M_i and M_{i+1} as the successor of M_i. In addition, all ID information and their corresponding public keys are arranged and all the social members perform the following steps:

Step 1. M_i chooses a random number $r_i \in [1, p+1]$ and computes $K_{i,i+1} = T_{r_i}T_{S_{i+1}}(x) \bmod p$, $C_i = E_{K_{i,i+1}}[GID_i||ID_i||ID_{i+1}||T_{r_i}(x) \bmod p]$ and $MAC_i = H(GID_i||ID_i||ID_{i+1}||C_i||H(T_{S_i}T_{S_{i+1}}(x) \bmod p)||T_{r_i}(x) \bmod p)$. Then M_i sends $\{C_i, T_{r_i}(x) \bmod p, MAC_i\}$ to its successor M_{i+1}, where $T_{S_{i+1}}(x) \bmod p$ is the public key of M_{i+1}.

Step 2. Upon receiving the message from M_i, M_{i+1} computes $K'_{i,i+1} = T_{S_{i+1}}T_{r_i}(x) \bmod p$ and reveals the value of $(GID_i||ID_i||ID_{i+1}||T_{r_i}(x) \bmod p)$ by computing $D_{K'_{i,i+1}}[C_i]$. Then M_{i+1} judges whether $H(GID_i||ID_i||ID_{i+1}||C_i||H(T_{S_{i+1}}T_{S_i}(x) \bmod p)||T_{r_i}(x) \bmod p) \stackrel{?}{=} MAC_i$ if the two sides of the equation are equivalent, then M_{i+1} chooses a random number $r_{i+1} \in [1, p+1]$ and computes $K_{i+1,i} = T_{r_{i+1}}T_{S_i}(x) \bmod p$, $SK_{i,i+1} = T_{r_{i+1}}T_{r_i}(x) \bmod p$ $C_{i+1} = E_{K_{i+1,i}}[GID_i||ID_i||ID_{i+1}||T_{r_{i+1}}(x) \bmod p]$ and $MAC_{i+1} = H(GID_i||ID_i||ID_{i+1}||C_{i+1}||H(T_{S_{i+1}}T_{S_i}(x) \bmod p)||T_{r_{i+1}}(x) \bmod p||SK_{i,i+1})$. Then M_{i+1} sends $\{C_{i+1}, T_{r_{i+1}}(x) \bmod p, MAC_{i+1}\}$ to its predecessor M_i, where $T_{S_i}(x) \bmod p$ is M_i's public key.

Step 3. Upon receiving the message from M_{i+1}, M_i computes the values of $K'_{i+1,i} = T_{S_i}T_{r_{i+1}}(x) \bmod p$ and $SK'_{i,i+1} = T_{r_i}T_{r_{i+1}}(x) \bmod p$. Then M_i reveals

TABLE 1. Notations used throughout the paper.

Symbol	Description
OSN	The online social networking platform.
M_i	The ith social member.
ID_i	The identity of M_i.
GID_i	The identity of the social group M.
M	A set of n social members, where $M = \{M_1, M_2, \ldots, M_n\}$ is the initial social set of GID_i.
ID_{member}	The identities of all existing n social members of GID_i, where $ID_{member} = (ID_1 \|\| ID_2 \|\| \ldots \|\| ID_n)$.
S_i	M_i's private key based on extended chaotic maps.
PK_i	M_i's public key based on extended chaotic maps, where $PK_i = T_{S_i}(x) \bmod p$.
$E_K[.]/D_K[.]$	A pair of symmetric encryption/decryption functions with the key K.
$H(.)$	A chaotic maps-based one-way hash function [38].
$SK_{i,i+1}$	The common session key, which is established between M_i and its successor M_{i+1}.
GSK_i	The group session key, which is established between the members of M.
$\|\|$	Message concatenation.
$\oplus$	A bitwise exclusive OR operation.
$\stackrel{?}{=}$	The comparison operation, judge whether two values are identical or not.

the value of $(GID_i||ID_i||ID_{i+1}||T_{r_{i+1}}(x) \bmod p)$ by computing $D_{K'_{i+1,i}}[C_{i+1}]$. Next M_i judges whether $H(GID_i||ID_i||ID_{i+1}||C_{i+1}||H(T_{S_i}T_{S_{i+1}}(x) \bmod p)|| T_{r_{i+1}}(x) \bmod p||SK'_{i,i+1}) \stackrel{?}{=} MAC_{i+1}$. If the two sides of the equation are not equivalent, the authentication fails and the session is terminated by M_i. Otherwise, M_i computes $MAC'_i = H(GID_i|| ID_i||ID_{i+1}||H(T_{S_i}T_{S_{i+1}}(x) \bmod p)||SK'_{i,i+1})$ and takes $SK'_{i,i+1}$ as the common session key shared between M_i and M_{i+1}. To achieve the property of mutual authentication, M_i sends MAC'_i to its successor M_{i+1}.

Step 4. Upon receiving the response MAC'_i from its predecessor, M_{i+1} judges whether $H(GID_i||ID_i||ID_{i+1}|| H(T_{S_{i+1}}T_{S_i}(x) \bmod p)||SK_{i,i+1}) \stackrel{?}{=} MAC'_i$. If not, the authentication fails and the session is terminated by M_{i+1}. Otherwise, M_{i+1} takes $SK_{i,i+1}$ as the common session key shared between M_{i+1} and M_i.

Note that the above-mentioned steps can be executed simultaneously. Thus, each social member M_i can establish two common session keys $SK_{i,i+1}$ and $SK_{i-1,i}$ with its successor M_{i+1} and predecessor M_{i-1}, respectively. Here, M_1 establishes two session keys $SK_{1,2}$ and $SK_{n,1}$ with its successor M_2 and its predecessor M_n. M_n establishes two session keys $SK_{n,1}$ and $SK_{n-1,n}$ with its successor M_1 and its predecessor M_{n-1}. The details of this phase are depicted in Fig. 2.

C. GROUP KEY AGREEMENT PHASE

In this phase, each social member M_i of GID_i computes the value of $X_i = B_{i-1} \oplus B_i = H(GID_i \oplus ID_{member} \oplus SK_{i-1,i} \oplus T_{S_i}T_{S_{i-1}}(x) \bmod p) \oplus H(GID_i \oplus ID_{member} \oplus SK_{i,i+1} \oplus T_{S_i}T_{S_{i+1}}(x) \bmod p)$ and multicasts X_i to the social group G_i. We can see the value of B_i in Table 2. After receiving all the values for X_i, each M_i validates whether $X_1 \oplus X_2 \oplus \ldots \oplus X_{n-1} \oplus X_n \stackrel{?}{=} 0$. If not, then M_i outputs an error symbol $\perp$ and aborts this phase. Otherwise, M_i can use B_i and X_i to obtain all the values of $B_j (j = 1, \ldots, n)$ by using the continuous XOR method. For example, M_1 uses its $B_1 = H(GID_i \oplus ID_{member} \oplus SK_{1,2} \oplus T_{S_1}T_{S_2}(x) \bmod p)$ to get M_2's $B_2 = H(GID_i \oplus ID_{member} \oplus SK_{2,3} \oplus T_{S_2}T_{S_3}(x) \bmod p)$ by computing $X_2 \oplus B_1$, where $X_2 = B_1 \oplus B_2$. After obtaining B_2, M_1 can further use it to obtain M_3's $B_3 = H(GID_i \oplus ID_{member} \oplus SK_{3,4} \oplus T_{S_3}T_{S_4}(x) \bmod p)$ by computing $X_3 \oplus B_2$. Finally, after obtaining all the values of B_j, all social members of GID_i can establish the common group session key GSK_i by computing $GSK_i = H(B_1||B_2||\ldots||B_n)$, where $GSK_1 = GSK_2 = \ldots = GSK_n$. Once GSK_i is established, all members of GID_i can use it for secure communication. No outsider no outsiders (including the original OSN) can spy on the communication of the newly created social group. The details of this phase are depicted in Fig. 3.

TABLE 2. The value of B_i.

Parameter	Value
B_1	$H(GID_i \oplus ID_{member} \oplus SK_{1,2} \oplus T_{S_1}T_{S_2}(x) \bmod p)$
$\vdots$	$\vdots$
B_i	$H(GID_i \oplus ID_{member} \oplus SK_{i,i+1} \oplus T_{S_i}T_{S_{i+1}}(x) \bmod p)$
$\vdots$	$\vdots$
B_n	$H(GID_i \oplus ID_{member} \oplus SK_{n,1} \oplus T_{S_n}T_{S_1}(x) \bmod p)$

D. SOCIAL MEMBER WITHDRAWAL PHASE

In case a social member M_j withdraws from the social group, the member set M' will have $(n-1)$ remaining members, where $M' = \{M_1, M_2, \ldots, M_{j-1}, M_{j+1}, \ldots, M_{n-1}\}$. To secure further communications, and prevent the member who has exited the group from accessing the content of the group, all the remaining social members must update the group key. In the design of our scheme, M_{j+1} is the new successor of M_{j-1} and only two members M_{j-1} and M_{j_1} must generate a new session key $SK_{j-1,j+1}$ shared between them. In this way, M_{j-1} and M_{j+1} can remove the pre-shared session keys $SK_{j-1,j}$ and $SK_{j,j+1}$ with M_j. Therefore, M_{j-1} needs to generate a new authentication message $\{C_{j-1}, T_{r_{j-1}}(x) \bmod p, MAC_{j-1}\}$ and send it to its new successor M_{j+1}. Upon receiving $\{C_{j-1}, T_{r_{j-1}}(x) \bmod p, MAC_{j-1}\}$ from its new predecessor M_{j-1}, M_{j+1} verifies the validity of the message $\{C_{j-1}, T_{r_{j-1}}(x) \bmod p, MAC_{j-1}\}$ and agrees to the new shared session key $SK_{j-1,j+1}$. Finally, all the existing $(n-1)$ social members in GID_i can agree to a new group session key by

Member M_i | Member M_{i+1}

Choose a random number r_i

Compute $K_{i,i+1} = T_{r_i} T_{S_{i+1}}(x) \bmod p$

Compute C_i

Compute MAC_i

$$\xrightarrow{\{C_i, T_{r_i}(x) \bmod p, MAC_i\}}$$

Compute $K'_{i,i+1} = T_{S_{i+1}} T_{r_i}(x) \bmod p$

Compute $D_{K'_{i,i+1}}[C_i]$

Check the validity of MAC_i

Choose a random number r_{i+1}

Compute $K_{i+1,i} = T_{r_{i+1}} T_{S_i}(x) \bmod p$

Compute $SK_{i,i+1} = T_{r_{i+1}} T_{r_i}(x) \bmod p$

Compute C_{i+1}

Compute MAC_{i+1}

$$\xleftarrow{\{C_{i+1}, T_{r_{i+1}}(x) \bmod p, MAC_{i+1}\}}$$

Compute $K'_{i+1,i} = T_{S_i} T_{r_{i+1}}(x) \bmod p$

Compute $SK'_{i,i+1} = T_{r_i} T_{r_{i+1}}(x) \bmod p$

Compute $D_{K'_{i+1,i}}[C_{i+1}]$

Check the validity of MAC_{i+1}

Compute MAC'_i

$$\xrightarrow{\{MAC'_i\}}$$

Check the validity of MAC'_i

FIGURE 2. The schematic of predecessor and successor authentication phase.

Member M_{i-1} | Member M_i | Member M_{i+1}

(Has got $SK_{i-1,i}$ together with M_{i-1}) (Has got $SK_{i,i+1}$ together with M_{i+1})

Compute $X_i = B_{i-1} \oplus B_i$

$$\xleftarrow{\text{Multicast } \{X_i\}} \qquad \xrightarrow{\text{Multicast } \{X_i\}}$$

Get all $X_i (i = 1, 2, \ldots, n)$

Check if $X_1 \oplus X_2 \oplus \ldots \oplus X_{n-1} \oplus X_n \stackrel{?}{=} 0$

Derive all $B_i = X_i \oplus B_{i-1}$

Establish $GSK_i = H(B_1 || B_2 || \ldots || B_n)$

FIGURE 3. The schematic description of the group key agreement phase.

recomputing the protocol presented in Section 3.2. Note that $ID_{member} = (ID_1 || ID_2 || \ldots || ID_{j-1} || ID_{j+1} || \ldots || ID_n)$.

E. SOCIAL MEMBER JOINING PHASE

In case a new social member is authorized to join the social group which has an existing size of n, the size of the new set $M"$ will change to $(n + 1)$ members, and $M" = \{M_1, M_2, \ldots, M_{n+1}\}$. Thus the new social member M_{n+1} will become the successor of the member M_n and the member M_1 will become the successor of the member M_{n+1}. In the design of our scheme, only three members M_n, M_{n+1} and M_1 must generate two new session keys $SK_{n,n+1}$ and $SK_{n+1,1}$, one session key being shared between M_n and M_{n+1} and the other between M_{n+1} and M_1. As a result, M_n will need to send a new message $\{C_n, T_{r_n}(x) \bmod p, MAC_n\}$ to its new successor M_{n+1}. The new member M_{n+1} will in turn need to send the message $\{C_{n+1}, T_{r_{n+1}}(x) \bmod p, MAC_{n+1}\}$ to its new successor M_1. Then M_{n+1} will verify the validity of the message $\{C_n, T_{r_n}(x) \bmod p, MAC_n\}$ and compute the new session key $SK_{n,n+1}$ shared between M_{n+1} and its new predecessor M_n. Similarly, the first social member M_1 will update its new session key with $SK_{n+1,1}$. Finally, all the $(n+1)$ social members in GID_i will obtain a new group session key by recomputing the protocol of Section 3.2. Note that $ID_{member} = (ID_1||ID_2||\ldots||ID_n||ID_{n+1})$.

IV. INFORMAL SECURITY ANALYSIS

According to the security goals mentioned in Section 1.1, in this section we provide an informal analysis of the security properties of our proposed scheme.

A. PROVISION OF SOCIAL GROUP IDENTIFICATION

When a social member M_i joins different online social groups on the social network platform, it is essential to help M_i identify the correct source of each received message. In order to facilitate this property, during the member authentication and group key agreement phases of our proposed scheme, the group identity GID_i is embedded in the transmitted messages which ensures a high rate of efficiency in the authentication procedures.

B. PROVISION OF PRIVACY PRESERVATION

Based on the design of our proposed scheme, the original identity ID_i of the member M_i is masked during the communications with social members over public channels. Therefore, any malicious outsider $\mathcal{M}_{\mathcal{E}}$ who attempts to eavesdrop on the communication cannot launch security attacks to compromise M_i's real identity ID_i. During the predecessor and successor authentication phase, M_i's real identity is shared via the encrypted message $C_i = E_{K_{i,i+1}}[GID_i||ID_i||ID_{i+1}||T_{r_i}(x) \bmod p]$ using $K_{i,i+1}$. Therefore, without the knowledge of r_i, the malicious outsider $\mathcal{M}_{\mathcal{E}}$ cannot obtain the ids ID_i and ID_{i+1}. Moreover, untraceability is ensured by the participating member since a random number is always changed for each authentication request. In the proposed scheme, all the identities of the members of the social group are always transmitted in cipher format instead of plaintext to guarantee the privacy and security of the members of the OSN.

C. PROVISION OF MUTUAL AUTHENTICATION AND MEMBER VERIFICATION

In the predecessor and successor authentication phase of our proposed scheme, after receiving the message $\{C_i, T_{r_i}(x) \bmod p, MAC_i\}$, the member M_{i+1} is allowed to authentication the member M_i by means of the equivalence relation $H(GID_i||ID_i||ID_{i+1}||C_i||H(T_{S_{i+1}}T_{S_i}(x) \bmod p)||T_{r_i}(x) \bmod p) \stackrel{?}{=} MAC_i$. This is because only the legitimate member M_i can provide the correct value $H(T_{S_i}T_{S_{i+1}}(x) \bmod p)$ of M_i to M_{i+1} after embedding it in MAC_i. After receiving the message $\{C_{i+1}, T_{r_{i+1}}(x) \bmod p, MAC_{i+1}\}$ from M_{i+1}, M_i verifies the validity of the message M_{i+1} by means of a similar equivalence relation $H(GID_i||ID_i||ID_{i+1}||C_{i+1}||H(T_{S_i}T_{S_{i+1}}(x) \bmod p)||T_{r_{i+1}}(x) \bmod p||SK'_{i,i+1}) \stackrel{?}{=} MAC_{i+1}$. Only the legitimate member M_{i+1} can utilize its private key S_{i+1} to compute the correct value of $H(T_{S_{i+1}}T_{S_i}(x) \bmod p)$ and embed it in MAC_{i+1}. For the group key authentication, each participating member of the group $M_i(i = 1, \ldots, n)$ checks for the equality $X_1 \oplus X_2 \oplus \ldots \oplus X_n \stackrel{?}{=} 0$. If the left-hand side (LHS) of the equation does not equal zero (the right-hand side (RHS)) then an error symbol $\perp$ is outputted and the group session key GSK_i is invalidated. If the LHS and RHS match, then the condition that the group session key GSK_i is authentic is satisfied.

D. PROVISION OF FAIRNESS IN GROUP KEY ESTABLISHMENT

As shown in the predecessor and successor authentication phase, the members M_i and M_{i+1} randomly select their contributions $T_{r_i}(x) \bmod p$ and $T_{r_{i+1}}(x) \bmod p$, respectively. Then M_i and M_{i+1} exchange them over a public channel and agree on the common secret value $SK_{i,i+1}$. As shown in the group key agreement phase, each social member M_i of the group GID_i individually computes his or her parameter B_i. The group session key $GSK_i = H(B_1||B_2||\ldots||B_n)$ contains equal contributions from every participating member $M_i(i = 1, \ldots, n)$. Therefore, in the proposed scheme, the fairness of the group key establishment process is ensured without the help of any third party trusted key center or centralized key distributor.

E. PROVISION OF GROUP KEY UPDATE

When a new member wants to join a social group or an existing member wants to withdraw from the group, the scheme should successfully prevent the withdrawing member from accessing newer group data, and the joiing member from accessing existing old data in the group. In the withdrawal phase of our proposed scheme, only M_{j-1} and M_{j+1} need to compute the new secret value $SK_{j-1,j+1}$. After that, M_j is excluded from the social group and any existing member $M_i(i = 1, 2, \ldots, n, i \neq j)$ of M' can update and verify the validity of the new group session key for future use. Similarly, in the social member joining phase of our proposed scheme, only M_n, M_{n+1} and M_1 need to compute the new secret values $SK_{n,n+1}$ and $SK_{n+1,1}$. Afterwards, the new member M_{n+1} is included in the group and each member $M_i(i = 1, 2, \ldots, n, n+1)$ of M'' can generate the new group session key for future use. In this wayour proposed scheme provides the property of dynamic group key management.

V. FORMAL SECURITY PROOF

In this section, we present the proof of the security of all the proposed phases in our scheme using the BAN logic [2].

A. SECURITY OF AUTHENTICATION PHASE

In this subsection, we demonstrate that a member M_i and a member M_{i+1} can achieve mutual authentication using a shared key $SK_{i,i+1}$. First, some notations and rules to describe the BAN logic are provided as follows:

1) NOTATIONS

1) $P \models X$: P believes X also called P would be entitled to believe X. In particular, P may act as though X is true.
2) $P \triangleleft X$: P sees X. Someone has sent a message containing X to P and P can read and repeat X.
3) $P \mid\sim X$: P once said X. P sent a message including X at some time. Note that P does not know whether the message was sent long ago or during the current iteration of the protocol, but P knows that $P \models X$ when the message was sent.
4) $P \Rightarrow X$: P has jurisdiction over X. P controls X which is subject to the jurisdiction of P and P is trusted for X.
5) $\sharp(X)$: X is fresh. X has not been sent in a message at any time before the execution of current iteration of the protocol.
6) $P \overset{K}{\longleftrightarrow} Q$: P and Q may use the shared key K to communicate securely. We say that K is good, if K can never be discovered by any other participant except P or Q, or a participant trusted by either P or Q.
7) $\{X\}_K$: The formula X is encrypted under a key K.

2) RULES

1) Message meaning rule: $\dfrac{P \models Q \overset{K}{\longleftrightarrow} P, P \triangleleft \{X\}_K}{P \models Q \mid\sim X}$.
It means that if P believes that K is a key shared with Q and P sees X encrypted using K, then P believes that Q once said X.
2) Nonce verification rule: $\dfrac{P \models \sharp(X), P \models Q \mid\sim X}{P \models Q \models X}$.
It means that if P believes that X is fresh and Q once said X, then P believes that Q believes X.
3) Jurisdiction rule: $\dfrac{P \models Q \Rightarrow X, P \models Q \models X}{P \models X}$. It means that if P believes that Q has jurisdiction over X and also believes that Q believes X, then P believes X.
4) Belief rule: $\dfrac{P \models Q \models (X, Y)}{P \models Q \models X}$. It means that if P believes that Q believes (X, Y) then P believes that Q believes X.

3) GOALS

We want to show that the authentication phase in our scheme should achieve the following goals:

$G_1 : M_i \models (M_i \overset{SK_{i,i+1}}{\longleftrightarrow} M_{i+1})$.
$G_2 : M_{i+1} \models (M_i \overset{SK_{i,i+1}}{\longleftrightarrow} M_{i+1})$.
$G_3 : M_i \models M_{i+1} \models (M_i \overset{SK_{i,i+1}}{\longleftrightarrow} M_{i+1})$.
$G_4 : M_{i+1} \models M_i \models (M_i \overset{SK_{i,i+1}}{\longleftrightarrow} M_{i+1})$.

4) IDEALIZATION OF THE COMMUNICATION MESSAGES

Here, we idealize the communication messages of the authentication phase using the scheme listed as below:

$Message_1 : M_i \rightarrow M_{i+1} : \{C_i, T_{r_i}(x), MAC_i\}$.
$Message_2 : M_{i+1} \rightarrow M_i : \{C_{i+1}, T_{r_{i+1}}(x), MAC_{i+1}\}$.
$Message_3 : M_i \rightarrow M_{i+1} : \{MAC_i'\}$.

5) INITIAL ASSUMPTIONS

We define some initial assumptions for our scheme as under:

$A_1 : M_i \models \sharp(S_i)$.
$A_2 : M_{i+1} \models \sharp(S_{i+1})$.
$A_3 : M_i \models \sharp(T_{S_i}(x))$.
$A_4 : M_{i+1} \models \sharp(T_{S_{i+1}}(x))$.

A_1 and A_2 indicate that the members M_i and M_{i+1} generate their own private keys as S_i and S_{i+1}. Hence, we assume that they are fresh. Therefore, according to A_1 and A_2, A_3 and A_4 are reasonable.

6) DETAILED DESCRIPTION

Based on the rules of the BAN logic, we prove that our scheme can achieve the defined goals using the initial assumptions.

a: FOR THE GOAL 1

Since $K_{i+1,i} = T_{r_{i+1}}T_{S_i}(x)$, we can obtain $S_1 : M_i \models (M_{i+1} \overset{K_{i+1,i}}{\longleftrightarrow} M_i)$ using $Message_2$ and A_3. From $Message_2$, we have $M_i \triangleleft \{C_{i+1}\}_{K_{i+1,i}}$. Since C_{i+1} contains $T_{r_{i+1}}(x)$, it implies that $S_2 : M_i \triangleleft \{T_{r_{i+1}}(x)\}_{K_{i+1,i}}$. As per the message meaning rule, using S_1 and S_2, we can obtain $S_3 : M_i \models M_{i+1} \mid\sim T_{r_{i+1}}(x)$. As per the nonce verification rule, using $M_i \models \sharp(T_{r_{i+1}}(x))$ and S_3, we can obtain $S_4 : M_i \models M_{i+1} \models T_{r_{i+1}}(x)$. According to the jurisdiction rule, by using $M_i \models M_{i+1} \Rightarrow T_{r_{i+1}}(x)$ from $Message_2$ and S_4, we can obtain $M_i \models T_{r_{i+1}}(x)$. Since $SK_{i,i+1} = T_{r_{i+1}}T_{r_i}(x)$, it implies $M_i \models (M_i \overset{SK_{i,i+1}}{\longleftrightarrow} M_{i+1})$.

b: FOR THE GOAL 2

Since $K_{i,i+1} = T_{r_i}T_{S_{i+1}}(x)$, using $Message_1$ and A_4, we can obtain $S_5 : M_{i+1} \models (M_i \overset{K_{i,i+1}}{\longleftrightarrow} M_{i+1})$. Using $Message_1$, we have $M_{i+1} \triangleleft \{C_i\}_{K_{i,i+1}}$. Since C_i contains $T_{r_i}(x)$, it implies that $S_6 : M_{i+1} \triangleleft \{T_{r_i}(x)\}_{K_{i,i+1}}$. As per the message meaning rule, using S_5 and S_6, we can obtain $S_7 : M_{i+1} \models M_i \mid\sim T_{r_i}(x)$. As per the nonce verification rule, using $M_{i+1} \models \sharp(T_{r_i}(x))$ and S_7 we obtain $S_8 : M_{i+1} \models M_i \models T_{r_i}(x)$. As per the jurisdiction rule, by using $M_{i+1} \models M_i \Rightarrow T_{r_i}(x)$ from $Message_1$ and S_8 we obtain $M_{i+1} \models T_{r_i}(x)$. Since $SK_{i,i+1} = T_{r_{i+1}}T_{r_i}(x)$, it implies $M_{i+1} \models (M_i \overset{SK_{i,i+1}}{\longleftrightarrow} M_{i+1})$.

c: FOR THE GOAL 3

We also have $S_9 : M_i \models (M_i \overset{SK_{i,i+1}}{\longleftrightarrow} M_{i+1})$ from goal 1 and $S_{10} : M_i \triangleleft \{MAC_{i+1}\}_{SK_{i,i+1}}$ from $Message_2$. As per the message meaning rule, using S_9 and S_{10}, we can obtain $S_{11} : M_i \models M_{i+1} \mid\sim MAC_{i+1}$. According to the nonce verification rule, using $M_i \models \sharp(MAC_{i+1})$ and S_{11}, we can obtain $M_i \models M_{i+1} \models MAC_{i+1}$. Since MAC_{i+1} contains

$SK_{i,i+1}$, using the brief rule it implies $M_i \models M_{i+1} \models SK_{i,i+1}$. Therefore, we can obtain $M_i \models M_{i+1} \models (M_i \overset{SK_{i,i+1}}{\longleftrightarrow} M_{i+1})$.

d: FOR THE GOAL 4

From the goal 2, we have S_{12} : $M_{i+1} \models (M_i \overset{SK_{i,i+1}}{\longleftrightarrow} M_{i+1})$. We also have S_{13} : $M_{i+1} \triangleleft \{MAC_i'\}_{SK_{i,i+1}}$ from $Message_3$. According to the message meaning rule, using S_{12} and S_{13}, we can obtain S_{14} : $M_{i+1} \models M_i \mid\sim MAC_i'$. According to the nonce verification rule, using $M_{i+1} \models \sharp(MAC_i')$ and S_{14}, we can obtain $M_{i+1} \models M_i \models MAC_i'$. Since MAC_i' contains $SK_{i,i+1}$, it implies $M_{i+1} \models M_i \models SK_{i,i+1}$ by the brief rule. Thus, we can obtain $M_{i+1} \models M_i \models (M_i \overset{SK_{i,i+1}}{\longleftrightarrow} M_{i+1})$.

B. SECURITY OF GROUP KEY AGREEMENT PHASE

Passive attack is a well known technique of attack for compromising the group key agreement scheme. A passive attacker is a passive adversary who cannot compute the group key by eavesdropping on the transmitted messages over a public channel. In this subsection, we show that our scheme is secure against passive attacks.

Theorem 1: Under the Chaotic-based decisional Diffie-Hellman (CDDH) problem, the proposed group key agreement scheme is secure against passive attacks.

Proof: Suppose that there is an adversary $\mathcal{A}$ who wants to obtain the information about the group session key by eavesdropping on the transmitted messages over a public channel. Then, we assume that $\mathcal{A}$ may obtain all transmitted messages $(GID_i, ID_{member}, T_{S_i}(x), C_i, T_{r_i}(x), MAC_i, MAC_i', X_i)$ for $i = 1, 2, \ldots, n$, where $X_i = B_{i-1} \oplus B_i$, $B_i = H(GID_i \oplus ID_{member} \oplus SK_{i,i+1} \oplus T_{S_i}T_{S_{i+1}})$, and $SK_{i,i+1} = T_{r_i}T_{r_{i+1}}(x)$.

Here, we show that our scheme prevents $\mathcal{A}$ from obtaining any information about the group session key $GSK_i = H(B_1||B_2|| \cdots ||B_n)$. Under the CDDH problem, we prove that two tuples $T_1 = \langle T_{S_1}(x), \ldots, T_{S_n}(x), T_{r_1}(x), \ldots, T_{r_n}(x), X_1, \ldots, X_n, GSK_i\rangle$ and $T_2 = \langle T_{S_1}(x), \ldots, T_{S_n}(x), T_{r_1}(x), \ldots, T_{r_n}(x), X_1, \ldots, X_n, R_1\rangle$ are computationally indistinguishable, where $R_1 \in \mathbb{Z}_p$. Using the contradiction proof, let us assume that the adversary $\mathcal{A}$ can efficiently distinguish between T_1 and T_2 within polynomial-time. Then, we can construct an algorithm $\mathcal{A}'$ that can efficiently distinguish a chaotic-based decision Diffie-Hellman (CDDH) problem $\langle T_a(x), T_b(x), T_{ab}(x) \mod p\rangle$ and $\langle T_a(x), T_b(x), R_2 \in \mathbb{Z}_p\rangle$ for some a and $b \in [1, p+1]$.

Without loss of generality, we set $T_{S_1}(x) = T_a(x)$ and $T_{S_2}(x) = T_b(x)$ as the input of $\mathcal{A}'$ and execute the following steps:

1) $\mathcal{A}'$ selects $t_i \in [1, p+1]$ and sets $T_{S_i} = T_{t_i}(x) \mod p$ for $i = 3, 4, \ldots, n$.
2) $\mathcal{A}'$ selects $v_i \in [1, p+1]$ and sets $T_{r_i} = T_{v_i}(x) \mod p$ for $i = 1, 2, \ldots, n$.
3) $\mathcal{A}'$ computes $SK_{i,i+1} = T_{v_i}T_{v_{i+1}}(x)$ for $i = 1, 2, \ldots, n$.
4) $\mathcal{A}'$ computes $B_1 = H(GID_i \oplus ID_{member} \oplus SK_{1,2} \oplus R_2)$ and $B_i = H(GID_i \oplus ID_{member} \oplus SK_{i,i+1} \oplus T_{t_i}T_{t_{i+1}}(x))$ for $i = 2, 3, \ldots, n$.
5) $\mathcal{A}'$ computes $X_i = B_{i-1} \oplus B_i$ for $i = 1, 2, \ldots, n$.

Finally, $\mathcal{A}'$ constructs all the values of $\langle T_{S_1}(x), \ldots, T_{S_n}(x), T_{r_1}(x), \ldots, T_{r_n}(x), X_1, \ldots, X_n, R_1 = H(H(GID_i \oplus ID_{member} \oplus SK_{1,2} \oplus R_2)||B_2|| \cdots ||B_n)\rangle$ and sends them to $\mathcal{A}$. $\mathcal{A}$ can determine whether GSK_i is equal to R_1. If it is true, it means that $R_2 = T_{ab}(x) \mod p$. Thus, the assumption that $\mathcal{A}'$ can run $\mathcal{A}$ as a subroutine to efficiently distinguish between the two tuples $(T_a(x), T_b(x), T_{ab}(x) \mod p)$ and $(T_a(x), T_b(x), R_2 \in \mathbb{Z}_p)$, is a contradiction. ■

C. SECURITY OF MEMBER JOINING/WITHDRAWAL PHASE

Forward secrecy is an important security property for the group key agreement scheme. Forward secrecy is defined as the feature that a newly joined group member cannot compute the previous group session keys. Using a similar argument, backward secrecy is defined as the feature that a withdrawn member cannot compute further group session keys. In this subsection, we show that our scheme provides forward and backward secrecies.

Theorem 2: Under the Chaotic-based computational Diffie-Hellman (CCDH) problem, the proposed group key agreement scheme provides forward secrecy for members joining and backward secrecy for member withdrawing from the group.

Proof: Assume that a new member M_{n+1} joins the group and has obtained a previous transcript $\langle GID_i, ID_{member}, T_{S_1}(x), \ldots, T_{S_n}(x), T_{r_1}(x), \ldots, T_{r_n}(x), X_1, \ldots, X_n\rangle$, where $X_i = B_{i-1} \oplus B_i$. Now, we show that M_{n+1} cannot compute the previous group session key $GSK_i = H(B_1||B_2|| \cdots ||B_n)$.

Under the CDDH problem, M_{n+1} cannot compute $SK_{j,j+1} = T_{r_j}T_{r_{j+1}}(x) = T_{r_jr_{j+1}}(x)$ and $T_{S_j}T_{S_{j+1}}(x) = T_{S_jS_{j+1}}(x)$ for some $j \in \{1, 2, \ldots, n\}$. Thus, M_{n+1} can never recover $B_j = H(GID_i \oplus ID_{member} \oplus SK_{j,j+1} \oplus T_{S_j}T_{S_{j+1}}(x))$. Note that if M_{n+1} can recover B_j, then all B_i can be recovered by $X_1, \ldots, X_n$. In other words, our proposed group key agreement scheme provides forward secrecy for members joining a group.

Without loss of generality, we assume that the member M_n leaves the group. Now, we show that M_n cannot compute further group session keys $GSK_i' = H(B_1'||B_2'|| \cdots ||B_{n-1}')$ even if M_n eavesdrops on the further transmitted messages $\langle GID_i', ID_{member}', T_{S_1}(x), \ldots, T_{S_{n-1}}(x), T_{r_1}(x), \ldots, T_{r_{n-1}}(x), X_1', \ldots, X_{n-1}'\rangle$. Using a similar argument mentioned above, we can state that M_n cannot compute GSK_i'. It means that our scheme provides backward secrecy for member leaving the group. ■

VI. SIMULATION VERIFICATION WITH PROVERIF

Proverif is an automatic cryptographic protocol verifier, which is widely used for specifying and analyzing the security of authenticated key agreement protocols [5], [6], [19]. In this section, we utilize Proverif to further analyze the security and validity of the proposed protocol. The whole simulation contains the following procedures:

- First, a public channel *ch* is defined for the communications. SK_{ij} and SK_{ji} are the session keys generated by the users. Then comes the functions, rules and queries (Fig. 4).

```
(* channel *)
free ch:channel. (* public channel *)

(* shared keys *)
free SKij:bitstring [private].
free SKji:bitstring [private].

(* constants *)
free si:bitstring [private]. (* the ith member's secret key *)
free sj:bitstring [private]. (* the jth member's secret key *)
free Pi:bitstring. (* the ith member's public key *)
free Pj:bitstring. (* the jth member's public key *)
const x:bitstring.
const IDi:bitstring. (* the ith member's information *)
const GIDi:bitstring.
const IDj:bitstring. (* the jth member's information *)
const GIDj:bitstring.

(* functions & reductions & equations *)
fun h(bitstring):bitstring. (* hash function *)
fun senc(bitstring,bitstring):bitstring. (* symmetric encryption *)
reduc forall m:bitstring, key:bitstring; sdec(senc(m,key),key)=m.
fun con(bitstring,bitstring):bitstring. (* concatenation operation *)
reduc forall m:bitstring, n:bitstring; getfirst(con(m,n))=m.
fun che(bitstring,bitstring):bitstring. (* Chebyshev *)

(* queries *)
query attacker(SKij).
query attacker(SKji).
```

FIGURE 4. Functions, rules and queries.

- The process of user S_i. (Fig. 5).

```
(* --------- Si's process --------- *)
let ProcessSi(Pi:bitstring,Pj:bitstring) =
    new ri:bitstring;  (* == start of step1 == *)
    let Ki = che(ri,Pj) in
    let Ti = che(ri,x) in
    let Ci = senc(con(con(con(GIDi,IDi),IDj),Ti),Ki) in
    let MACi = h(con(con(con(con(con(GIDi,IDi),IDj),Ci),h(che(si,Pj))),Ti)) in

    out(ch,(Ci,Ti,MACi));  (* == end of step1 == *)

    in(ch,(Cj:bitstring,Tj:bitstring,MACj:bitstring)); (* == start of step3 == *)
    let Kj' = che(ri,Pj) in
    let SKij = che(ri,Tj) in
    let t1 = sdec(Cj,SKij) in
    (***** t2 = con(GIDi,IDi),IDj) *)
    let t2 = getfirst(t1) in
    let MACj' = h(con(con(con(con(t2,Cj),h(che(si,Pj))),Tj),SKij)) in
    if MACj' = MACj then
    let MACi2 = h(con(con(t2,h(che(si,Pj))),SKij)) in
    out(ch,(MACi2)); (* == end of step3 == *)
    0.
```

FIGURE 5. The process of user S_i.

- The process of user S_j (Fig. 6).
- The main execution. (Fig. 7).
- The result of the proposed protocol. From the results, we can conclude that any two users S_i and S_j can securely generate a common session key SK_{ij} (Fig. 8). Consequently, all group members can calculate a group key *GSK*.

VII. PERFORMANCE ANALYSIS AND COMPARISONS

In this section, we provide a performance analysis of our proposed scheme. First, we define some cryptographic notations and list the computational costs of our proposed system

```
(* --------- Sj's process --------- *)
let ProcessSj(Pi:bitstring,Pj:bitstring) =
    in(ch,(Ci:bitstring,Ti:bitstring,MACi:bitstring)); (* == start of step2 == *)
    let Ki' = che(sj,Ti) in
    (***** t1 = con(con(con(GIDi,IDi),IDj),Ti) *)
    let t1 = sdec(Ci,Ki') in
    (***** t2 = con(con(GIDi,IDi),IDj) *)
    let t2 = getfirst(t1) in
    let MACi' = h(con(con(con(t2,Ci),che(sj,Pi)),Ci)) in
    if MACi' = MACi then
    new rj:bitstring;
    let Kj = che(rj,Pi) in
    let Tj = che(rj,x) in
    let SKji = che(rj,Ti) in
    let Cj = senc(con(t2,Tj),SKji) in
    let MACj = h(con(con(con(con(t2,Cj),h(che(sj,Pi))),Tj),SKji)) in
    out(ch,(Cj,Tj,MACj));  (* == end of step2 == *)

    in(ch,(MACi2:bitstring)); (* == start of step4 == *)
    let MACi2' = h(con(con(t2,h(che(sj,Pi))),SKji)) in
    if MACi2' = MACi2 then
    0. (* == endof step4 == *)
```

FIGURE 6. The process of user S_j.

```
(* --------- Main --------- *)
process
    let Pi = che(si,x) in
    let Pj = che(sj,x) in
    (!ProcessSi(Pi,Pj) | !ProcessSj(Pi,Pj))
```

FIGURE 7. The main execution.

```
-- Query not attacker(SKji[])
Completing...
Starting query not attacker(SKji[])
RESULT not attacker(SKji[]) is true.

-- Query not attacker(SKij[])
Completing...
Starting query not attacker(SKij[])
RESULT not attacker(SKij[]) is true.
```

FIGURE 8. The result of the proposed protocol.

in Table 3. According to the experimental results performed in [16] with specifications as CPU: 2.4 GHz Intel core i5, RAM: 4.0 GB, using a GNU with multiple precision library and OpenSSL library. The average time of execution for one T_{Hash}, one T_{Che} and one T_{Sym} are 0.02 ms, 32.9 ms and 0.042 ms, respectively. As seen in Table 3, the computation tasks for each member includes 10 Chebyshev polynomial operations, 4 symmetric encryption/decryption operations and 11 one-way hashing operations. Among these, the Chebyshev polynomial operations are undoubtedly the most time-consuming task. Compared with the solutions of RSA and ECC, the solution provided by extended chaotic maps, not only offers faster computations and smaller key sizes but also reduces the memory usage and bandwidth consumption. The proposed solution does not use time consuming modular exponential computations and scalar multiplications on elliptic curves. Therefore, we can conclude that the intensity of the real-time computation in our proposed scheme is quite acceptable for OSN systems.

In Table 4, we present a number of security requirements as well as functionality criteria to compare our proposed scheme with related existing schemes [17], [25], [33], [41]. From Table 4 we can observe that none of the schemes can

TABLE 3. The computation cost of our proposed scheme.

Phase	Predecessor and successor authentication	Group key agreement
Computation cost	$10T_{Che}$+$4T_{Sym}$+$8T_{Hash}$	$3T_{Hash}$
Execution time	329.328ms	0.06ms
Communication round	3 rounds	1 round

T_{Hash}: Time needed to execute a one-way hash function.
T_{Sym}: Time needed to execute a symmetric en/decryption computation.
T_{Che}: Time needed to execute a Chebyshev polynomial computation.

TABLE 4. Comparisons of our proposed scheme with related group key establishment schemes.

Scheme → Functionality ↓	Zhao et al.'s [41] (2010)	Lou et al.'s [25] (2013)	Vijayakumar's [33] (2014)	Jaiwsal et al.'s [17] (2016)	The proposed scheme
F1	Yes	Yes	No	Yes	Yes
F2	No	Yes	Yes	Yes	Yes
F3	No	No	Yes	No	Yes
F4	Yes	Yes	Yes	Yes	Yes
F5	No	Yes	No	Yes	Yes
F6	No	Yes	Yes	No	Yes
F7	Yes	Yes	Yes	No	Yes
F8	No	No	No	No	Yes
F9	No	No	No	No	Yes
F10	No	No	No	No	Yes

F1: No need for an online key center
F2: No need for a synchronized timestamp
F3: No need for a group chairman
F4: Prevention of member masquerading attack
F5: Provision of mutual authentication
F6: Provision of group key update
F7: Provision of privacy preserving
F8: Provision of fairness in group key establishment
F9: Provision of group identification
F10: Provision of formal security proof

provide formal security proof, fairness in group key establishment and group identification properties in OSN systems. In contrast with related schemes, if a social member M_i joins numerous different online social groups, the proposed scheme uses GID_i to identify the source of the received message and performs group identification. On the other hand, in our proposed scheme, the generation of the final group session key GSK_i contains equal contributions from n social members in $M = \{M_1, M_2, \ldots, M_n\}$ without the help of a chairman of group or an online key center, thus ensuring fairness in the group key agreement scheme. From Tables 3 and 4, we can demonstrate that the proposed scheme is efficient and robust for OSN systems in terms of computational overhead and security strength.

VIII. CONCLUSIONS

In this paper, we presented a provably secure group key agreement scheme for OSN environments. Using the proposed scheme, a social group in a social networking platform generates a dynamic group session key by exchanging some authentication parameters among it's validated social members. The proposed scheme is based on the extended chaotic maps-based cryptographic model and is energy-efficient and supports anonymous interactions with high security and low computational costs. To prove the strength of it's security, we verified our scheme using the BAN logic analysis. The results of the analysis confirm that the proposed scheme is robust against both passive and active attacks. Therefore, we can conclude that the overall security, functionality and efficiency of our scheme makes it suitable for use in web-based OSNs.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

[1] H. K. Aslan and G. F. Elkabbany, "Design of a high performance implementation of a tree-based multicast key distribution protocol," *Int. J. Netw. Secur.*, vol. 15, no. 2, pp. 80–96, 2013.

[2] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.

[3] *BeeTalk*. Accessed: Aug. 16, 2016. [Online]. Available: http://beetalk.tw/

[4] P. Bergamo, P. D'Arco, A. De Santis, and L. Kocarev, "Security of public-key cryptosystems based on Chebyshev polynomials," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 7, pp. 1382–1393, Jul. 2005.

[5] C.-M. Chen, K.-H. Wang, K.-H. Yeh, B. Xiang, and T.-Y. Wu, "Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications," *J. Ambient Intell. Humanized Comput.*, pp. 1–10, Sep. 2018, doi: 10.1007/s12652-018-1029-3.

[6] S. A. Chaudhry, H. Naqvi, M. Sher, M. S. Farash, and M. U. Hassan, "An improved and provably secure privacy preserving authentication protocol for SIP," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 2, pp. 1–15, 2017.

[7] C. M. Chen, L. Xu, K.-H. Wang, S. Liu, and T.-Y. Wu, "Cryptanalysis and improvements on three-party-authenticated key agreement protocols based on chaotic maps," *J. Internet Technol.*, vol. 19, no. 3, pp. 679–687, 2018.

[8] R. Dutta and R. Barua, "Password-based encrypted group key agreement," *Int. J. Netw. Secur.*, vol. 3, no. 1, pp. 23–34, 2006.

[9] *Facebook*. Accessed: Aug. 16, 2016. [Online]. Available: https://zh-tw.facebook.com/

[10] *Google+*. Accessed: Aug. 16, 2016. [Online]. Available: https://plus.google.com/

[11] S. Han and E. Chang, "Chaotic map based key agreement with/out clock synchronization," *Chaos Solitons Fractals*, vol. 39, no. 3, pp. 1283–1289, 2009.

[12] D. He, Y. Chen, and J. Chen, "Cryptanalysis and improvement of an extended chaotic maps-based key agreement protocol," *Nonlinear Dyn.*, vol. 69, no. 3, pp. 1149–1157, 2012.

[13] S. K. H. Islam and G. P. Biswas, "A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks," *Ann. Telecommun.*, vol. 67, nos. 11–12, pp. 547–558, 2012.

[14] S. K. H. Islam and A. Singh, "Provably secure one-round certificateless authenticated group key agreement protocol for secure communications," *Wireless Pers. Commun.*, vol. 85, no. 3, pp. 879–898, 2015.

[15] S. K. H. Islam, M. S. Obaidat, P. Vijayakumar, E. Abdulhay, F. Li, and M. K. C. Reddy, "A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs," *Future Gener. Comput. Syst.*, vol. 84, pp. 216–227, Jul. 2018.

[16] A. Jabbari and J. Bagherzadeh, "A revised key agreement protocol based on chaotic maps," *Nonlinear Dyn.*, vol. 78, no. 1, pp. 669–680, 2014.

[17] P. Jaiswal and S. Tripathi, "An authenticated group key transfer protocol using elliptic curve cryptography," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 4, pp. 857–864, 2017, doi: 10.1007/s12083-016-043407.

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

[19] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for Internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[20] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[21] C.-C. Lee, C. T. Li, and C. W. Hsu, "A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps," *Nonlinear Dyn.*, vol. 73, no. 1, pp. 125–132, Jul. 2013.

[22] C.-C. Lee and C.-W. Hsu, "A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps," *Nonlinear Dyn.*, vol. 71, nos. 1–2, pp. 201–211, 2013.

[23] *LINE*. Accessed: Aug. 16, 2016. [Online]. Available: https://line.me/zh-hant/

[24] 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 Dyn.*, vol. 74, no. 4, pp. 1133–1143, 2013.

[25] D.-C. Lou, K.-C. Liu, and H.-F. Huang, "Efficient mobile conference scheme for wireless communication," *Informatica*, vol. 24, no. 1, pp. 59–70, 2013.

[26] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*. Boca Raton, FL, USA: CRC Press, 2003.

[27] *Paktor*. Accessed: Aug. 16, 2016. [Online]. Available: https://gopaktor.com/

[28] A. T. Sherman and D. A. McGrew, "Key establishment in large dynamic groups using one-way function trees," *IEEE Trans. Softw. Eng.*, vol. 29, no. 5, pp. 444–458, May 2003.

[29] M. Stanek, "A note on security protocol for multicast communications," *Int. J. Netw. Secur.*, vol. 14, no. 1, pp. 59–60, 2012.

[30] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 8, pp. 769–780, Aug. 2000.

[31] *Twitter*. Accessed: Aug. 16, 2016. [Online]. Available: https://twitter.com/?lang=zh-tw

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

[33] P. Vijayakumar, S. Bose, and A. Kannan, "Chinese remainder theorem based centralised group key management for secure multicast communication," *IET Inf. Secur.*, vol. 8, no. 3, pp. 179–187, May 2014.

[34] P. Vijayakumar, R. Naresh, L. J. Deborah, and S. K. H. Islam, "An efficient group key agreement protocol for secure P2P communication," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 3952–3965, 2016.

[35] S.-Y. Wang and C.-S. Laih, "Merging: An efficient solution for a time-bound hierarchical key assignment scheme," *IEEE Trans. Depend. Sec. Comput.*, vol. 3, no. 1, pp. 91–100, Jan. 2006.

[36] *WeChat*. Accessed: Aug. 16, 2016. [Online]. Available: https://http://www.wechat.com/zh_TW/

[37] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. Netw.*, vol. 8, no. 1, pp. 16–30, Feb. 2000.

[38] D. Xiao, F. Y. Shih, and X. Liao, "A chaos-based hash function with both modification detection and localization capabilities," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 15, no. 9, pp. 2254–2261, 2010.

[39] L. Xu and C. Huang, "Computation-efficient multicast key distribution," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 5, pp. 577–587, May 2008.

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

[41] J. Zhao, D. Gu, and Y. Li, "An efficient fault-tolerant group key agreement protocol," *Comput. Commun.*, vol. 33, no. 7, pp. 890–895, 2010.

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

[43] H. Zhu, "Secure chaotic maps-based group key agreement scheme with privacy preserving," *Int. J. Netw. Secur.*, vol. 18, no. 2, 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, 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.

TSU-YANG WU received the Ph.D. degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2010. He is currently an Associate Professor with the Fujian University of Technology, China. His research interests include applied cryptography, pairing-based cryptography, and information security.

CHIEN-MING CHEN is currently an Associate Professor with the College of Computer Science and Technology, Shandong University of Science and Technology, Shandong, China. His current research interests include network security, mobile internet, wireless sensor network, and cryptography.