sensors

MDPI

Article

A Round-Efficient Authenticated Key Agreement Scheme Based on Extended Chaotic Maps for Group Cloud Meeting

Tsung-Hung Lin [1], Chen-Kun Tsung [1,*], Tian-Fu Lee [2] and Zeng-Bo Wang [1]

[1] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, No.57, Sec. 2, Zhongshan Rd., Taiping District, Taichung 41170, Taiwan; duke@ncut.edu.tw (T.-H.L.); scottzxk20@gmail.com (Z.-B.W.)

[2] Department of Medical Informatics, Tzu Chi University, No.701, Sec. 3, Zhongyang Rd., Hualien 97004, Taiwan; jackytflee@mail.tcu.edu.tw

* Correspondence: ckt@ncut.edu.tw; Tel.: +886-4-2392-4505

Received: 30 October 2017; Accepted: 30 November 2017; Published: 3 December 2017

Abstract: The security is a critical issue for business purposes. For example, the cloud meeting must consider strong security to maintain the communication privacy. Considering the scenario with cloud meeting, we apply extended chaotic map to present passwordless group authentication key agreement, termed as Passwordless Group Authentication Key Agreement (PL-GAKA). PL-GAKA improves the computation efficiency for the simple group password-based authenticated key agreement (SGPAKE) proposed by Lee et al. in terms of computing the session key. Since the extended chaotic map has equivalent security level to the Diffie–Hellman key exchange scheme applied by SGPAKE, the security of PL-GAKA is not sacrificed when improving the computation efficiency. Moreover, PL-GAKA is a passwordless scheme, so the password maintenance is not necessary. Short-term authentication is considered, hence the communication security is stronger than other protocols by dynamically generating session key in each cloud meeting. In our analysis, we first prove that each meeting member can get the correct information during the meeting. We analyze common security issues for the proposed PL-GAKA in terms of session key security, mutual authentication, perfect forward security, and data integrity. Moreover, we also demonstrate that communicating in PL-GAKA is secure when suffering replay attacks, impersonation attacks, privileged insider attacks, and stolen-verifier attacks. Eventually, an overall comparison is given to show the performance between PL-GAKA, SGPAKE and related solutions.

Keywords: cloud meeting; group authenticated; key agreement; extended chaotic maps

1. Introduction

Communicating over the Internet is a convenient application as the development of the Internet becomes popular. People can communicate with each other via cloud meeting is a common application. A lot of companies deploy cloud meeting equipment to realize a remote discussion. Some special industries also take into account the cloud meeting, but they focus on the information security. For example, personalized information must be under controlled in medical conferences, and business confidentiality can not be tapped in cloud meeting.

The cloud meeting has following properties:

1. Known members: The meeting members are known before meeting. Therefore, the organizer has a participant list in advance.

2. Difficult preset: Even if the organizer has a participant list, generating the meeting setting, e.g., passwords or meeting tokens, in advance is inappropriate. Dynamically generating meeting setting is the optimal solution [1–7] for the security consideration.
3. Over the Internet: This is the core requirement for realizing a cloud meeting.
4. Multi-member communication: The communication within a pair of members is tractable. When the number of participants increases, to ensure that each member can identify each other is difficult.

For the fourth property, the cloud meeting can be classified into three categories, and they are one-to-one, one-to-many, and many-to-many models as shown in Figure 1. The most popular application is one-to-many model. For example, the user uses a password to log in to a web service. In this model, participants have a security communication based on a centralized server [8]. The many-to-many model is similar to the one-to-many model, but the many-to-many model is decentralized [9].

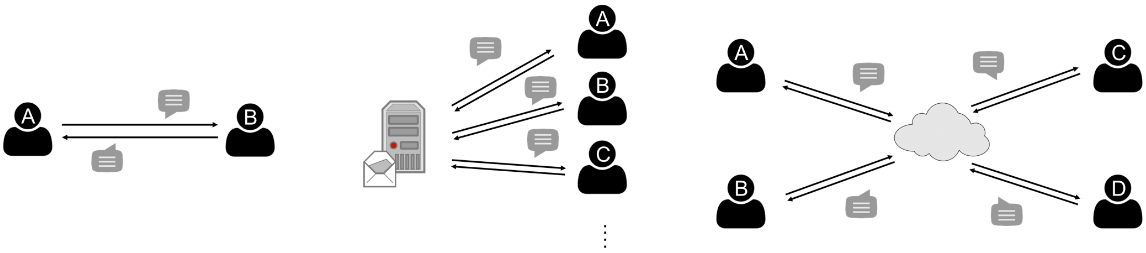

Figure 1. Three types of cloud meeting. (a) The left one is one-to-one communication; (b) the central one is one-to-many communication; and (c) the right one is many-to-many communication.

Both one-to-many and many-to-many models are popular in real world cloud meeting. For example, building a safety communication tunnel to avoid information loss is a possible solution [8]. As shown in case (b) of Figure 1, the server provides a safety communication tunnel for all connected members. The major advantage of the one-to-many model is the convenience. Although the many-to-many model does not suffer the attacks from hackers due to the decentralization, each member must have higher security equipment in the many-to-many model than in the one-to-many model. Therefore, we focus on the one-to-many model and propose a lightweight solution with security communication.

Before entities send messages with each other, they have to build up a secure communication. In the current secure communication technologies including Internet Protocol Security (IPSec) and https require a communication setup process with two steps: session key generation and message encryption/decryption. The goal of session key generation is to compute a session key for all communication members. Since the message that required by computing a session key is sent over the Internet, hiding the information applied to generate a session key is the major challenge. After all members have the same session key, they can use the session key to encrypt or decrypt messages in the second step. In this paper, we focus on the first step to design an efficient session key agreement scheme under the scenario drawn in case of Figure 1b.

Group authentication key agreement scheme is a possible solution in security cloud meeting. Each participant generates a session key to encrypt information, and it only can be used during this cloud meeting. Even if encrypted messages sent over the Internet are taken by man in the middle, they do not have enough information to get the original message. Diffie–Hellman key exchange is an appropriate technique to develop the group authentication key agreement scheme [8]. It guarantees high security for information exchange in a limited time period. A cloud meeting takes a few hours rather than several years, so Diffie–Hellman key exchange is secure for a cloud meeting.

However, Diffie–Hellman key exchange applies modular exponentiation to compute single-use session key, so it requires a lot of computation cost before the information exchange. In the cloud meeting, the schedules of many people may be rush, so they need an efficient solution for minimizing the setup time.

Another efficient key agreement protocol is extended chaotic map-based approaches [8,10]. These kind of schemes apply Chebyshev polynomials to provide the property, which is equivalent to the semigroup property of chaotic map [10–14]. The details are shown in Section 2 Preliminaries. Therefore, extended chaotic map-based approaches are efficient in computing session keys [15]. However, there is no group authentication key agreement scheme that applies the extended chaotic map in the one-to-many model [10].

There are some key agreement protocols that can be applied in case of Figure 1a. For example, Abdalla and Pointcheval provide a password-based approach for a pair of users [15]. Dutta and Barua extend the results of Abdalla and Pointcheval from one-to-one communication to the many-to-many model, and the shared password has been enhanced [16]. Kim et al. focus on the members join/leave a group without the assistance from a central server [3]. Boyd and Nieto address the efficiency of the key agreement protocol in terms of the number of rounds to generate a session key, and the proposed solution can be done in one round [17]. However, the solution still needs to be improved for the forward security issue.

For the group authentication, Lee et al. present a simple group password-based authenticated key agreement (SGPAKE) [8]. SGPAKE considers modular exponentiation, but the cost of generating session keys is not acceptable in cloud meeting. Therefore, we apply the extended chaotic map to propose the passwordless group authentication key agreement, termed by PL-GAKA. PL-GAKA is an extended chaotic map-based approach, so it improves the computation efficiency of SGPAKE. Since PL-GAKA is passwordless, meeting members do not need other password maintenance.

In our analysis, we first prove that each member can compute correct session key and they have security communication. Then, we refer to [8,18–22] to measure the security of PL-GAKA in terms of session key security, mutual authentication, perfect forward security, data integrity, and man-in-the-middle attack. Moreover, we also demonstrate that the proposed solution is safe when suffering replay attacks, impersonation attacks, privileged insider attacks, and stolen-verifier attacks.

The structure of this paper is as follows: the background knowledge is present in Section 2. The proposed PL-GAKA is illustrated in Section 3. In Section 4, we analyze the correctness, security, and the overall comparison. The conclusion and future works are illustrated in Section 5.

2. Preliminaries

In this section, we will show that the security of Diffie–Hellman key exchange, and how the chaotic map-based approaches can reduce the computation cost without sacrificing the security of key agreement. In the following context, we first give an example to show the way of computing a session key over the Internet. Then, we introduce the Diffie-Hellman problem, which is the major property to guarantee the communication security. Eventually, we show an alternative technique named by the chaotic map to reduce the computation cost.

Diffie–Hellman key exchange is a famous scheme in terms of security communications. Considering the following scenario of generating a session key before starting a safety communication: Alice and Bob would like to create a security communication within G rounds. Firstly, Alice selects a big prime p and a primitive root g. Then, Alice generates a secrete value a for this communication with Bob:

Step 1 Alice obtains the message $A = g^a \mod p$ and sends g, p, A to Bob over Internet.

Step 2 Bob also computes a secret value b for the communication with Alice. Bob computes the message $B = g^b \mod p$ and sends B to Alice. Moreover, Bob uses g, p, A and b to compute the session key $K = A^b \mod p = g^{ab} \mod p$.

Step 3 Alice can compute the session key $K = B^a \mod p = g^{ba} \mod p$ from B. Then, both Alice and Bob have the same session key and they can start to communicate with each other.

In step 3, Alice and Bob get the session key K, and then they can communicate with each other via encrypting/decrypting messages by K.

During the steps above, Alice and Bob focus on computing K in an open environment. Only Alice and Bob can derive correct K even if eavesdroppers capture the messages sent from Alice or Bob. The core idea of the safety in terms of generating K is the Diffie–Hellman problem and that is shown in the following definition.

Definition 1. *Diffie–Hellman problem [23]: Given appropriate settings of G and g, eavesdroppers obtain g^{ab} by solving the Diffie–Hellman problem.*

Solving Difie-Hellman problem is hard [23,24], and this is the reason that Diffie–Hellman key exchange provides high security. However, Diffie–Hellman key exchange requires heavy computation cost due to the modular exponentiation consideration. Designing a key ageerment approach with lower computation cost is a research direction.

Since Alice computes $K = B^a \mod p = g^{ba} \mod p$ and Bob computes $K = A^b \mod p = g^{ab} \mod p$, they derive the same K. Therefore, Alice and Bob can generate the session key via Internet. Chebyshev polynomials have similar properties as shown in the following definition.

Definition 2. *Semigroup property [25]: We have $T_r(T_s(x)) = T_{rs}(x)$ for different r and s, where* $\forall -1 \leq x \leq 1$.

The core idea of semigroup is similar to g^{ab} in the Diffie–Hellman problem. Semigroup implies that there is not a specific order for r and s. This property comes from Chebyshev polynomials, which is defined as $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$, where $T_0(x) = 1$, $T_1(x) = x$, $n \in \mathbb{Z}^+$, and $x \in \mathbb{R}$.

However, $-1 \leq x \leq 1$ is not enough provide high security in terms of the diversity of x, and Zhang extends the mapping range from $[-1, 1]$ to $(-\infty, \infty)$ [10]. The Extended Chebyshev polynomials are shown in Definition 3. The security can be improved dramatically. In other words, the scheme with semigroup property has similar security to that of the Diffie–Hellman key exchange.

Definition 3. *Extended Chebyshev polynomials: Given $x \in \mathbb{R}$, we have $T_r(T_s(x)) \mod p = T_{sr}(x) \mod p = T_s(T_r(x)) \mod p$ for different r and s.*

In other words, we can apply chaotic map functions to design a key agreement approach with lower computation costs than that required by Diffie–Hellman key exchange protocols. The chaotic map-based key agreement approaches have similar security to that of the Diffie–Hellman problem.

3. Proposed Solution

SGPAKE has three processes including registration, authentication, and password modification. PL-GAKA is a passwordless scheme, so password modification is not necessary. The processes of registration and authentication are illustrated in the following subsections. Moreover, the symbol system applied in this paper is shown in Table 1.

Table 1. The symbol system applied by the proposed solution.

Symbol	Definition
U_i	i-th user
GWN	The trusty authentication server
$h(.)$	One-way hash function

Table 1. *Cont.*

Symbol	Definition
K_G	The private key generated by GWN
UID_i	The id of U_i
p	A large prime number
T_r	Chaotic map
T	The timestamp
x	A variable within $(-\infty, \infty)$
K_{GS_i}	The identity of GWN for U_i
$Auth_{GS_i}$	The authentication information applied by U_i for verifying GWN
$Auth_{i1}$	The authentication information applied by U_j, $\forall j \neq i$, for verifying U_i
$Auth_{i2}$	The authentication information applied by GWN for verifying U_i
sk_i	The factor of generating session key for U_i
S_n	The list of participants
SK	The session key

3.1. Registration

The purpose of registration is to construct a list of potential meeting members for GWN. Each meeting member U_i provides the identity UID_i to GWN. GWN uses UID_i to generate the encrypted shared secret information K_{GS_i}, and then U_i are available to join a cloud meeting.

The major consideration is the security, and we have the following issues. The first issue is how GWN confirms U_i, and the second one is how to ensure the safety of the entire process. Since PL-GAKA is passwordless, UID_i is important information for verifying U_i. The whole registration can be completed in an offline and face-to-face process, and the secure solution can be applied to determine the user characteristics, e.g., smart card [18]. We focus on providing the communication security during the cloud meeting, and meeting members can be pre-defined before meeting. Therefore, the offline registration process is available for cloud meeting to ensure each member is verified. The registration processes are illustrated in Figure 2, and details are listed as follows:

Step 1 The user U_i registers his/her identity UID_i in GWN.
Step 2 GWN uses the private key k_G to compute $K_{GS_i} = h(UID_i \| k_G)$ and then sends K_{GS_i} to U_i via the secure channel.
Step 3 U_i saves K_{GS_i} for further authentications.

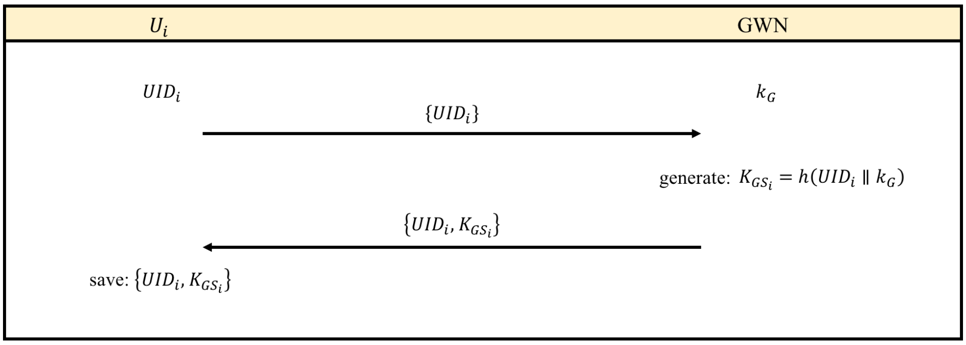
U_i
GWN
UID_i
k_G
$\{UID_i\}$
generate: $K_{GS_i} = h(UID_i \parallel k_G)$
$\{UID_i, K_{GS_i}\}$
save: $\{UID_i, K_{GS_i}\}$

Figure 2. The registration process.

3.2. Authentication

The communication security depends on the stable member. All members must know each other. When a member joins the meeting, the authentication process is launched to ensure that all members know each other including GWN.

The authentication process spreads four messages. In the beginning, each U_i sends the encrypted identity message M_1 to GWN. GWN verifies M_1 and sends the message M_2 including the list of meeting members and the encrypted server information back. After receiving M_2, U_i broadcasts M_3 including the information required by cross authentication. Then, U_i generates and broadcasts authentication information M_4. Eventually, each member authenticates each other and computes the session key for the encryption in the following meeting. We consider the timestamp in each message to guarantee that the process sequence can be tracked. Thus, when receiving a message, verifying the timestamp is the first task.

Consider n registered members who would like to participate in a cloud meeting. The proposed authentication process is illustrated in Figure 3, and the details are shown as follows:

Step 1 Each user U_i generates a random number a_i and computes $R_i = T_{a_i}(X) \mod p$. After considering the timestamp T_1, we have $R_i \oplus h(K_{GS_i} \| T_1)$. Then, the encrypted identity message $M_1 = \{UID_i, R_i \oplus h(K_{GS_i} \| T_1), T_1\}$ is organized and sent to GWN.

Step 2 As receiving M_1, GWN verifies T_1 firstly. GWN calculates $h(K_{GS_i} \| T_1)$ and obtains $R_i \oplus h(K_{GS_i} \| T_1)$ by Exclusive-OR (XOR) operation. Then, R_i is derived by $(R_i \oplus h(K_{GS_i} \| T_1)) \oplus h(K_{GS_i} \| T_1)$. According to semigroup property from Definition 3, we have $X_i = T_{b_i}(R_i) \mod p = T_{a_i b_i}(X) \mod p$ and $X'_i = T_{b_i}(R_{i+1}) \mod p = T_{a_{i+1} b_i}(X) \mod p$. Then, we use $Y_i = X_{i-1} \oplus h(K_{GS_i} \| R_i \| T_2 \| 0)$ and $Y'_i = X'_i \oplus h(K_{GS_i} \| R_i \| T_2 \| 1)$ to generate the authentication information $Auth_{GS_i} = h(K_{GS_i} \| R_i \| Y_i \| Y'_i)$ for verifying GWN. Eventually, the message $M_2 = \{S_n, Y_i, Y'_i, Auth_{GS_i}, T_2\}$, where $S_n = \{U_1, U_2, \ldots, U_n\}$, including meeting member list and GWN authentication information is sent to U_i.

Step 3 Any other member $U_i, \forall i \neq j$, receives M_2 and verifies T_2 and GWN by $Auth_{GS_i}$. Next, X_{i-1} and X'_i are derived by $Y_i \oplus h(K_{GS_i} \| R_i \| T_2 \| 0)$ and $Y'_i \oplus h(K_{GS_i} \| R_i \| T_2 \| 1)$, respectively. Thus, we can compute $V_i = T_{a_i}(X_{i-1}) \mod p = T_{a_{i-1} b_{i-1} a_i}(X) \mod p$, and $V_{i-1} = T_{a_i}(X'_i) \mod p = T_{a_{i+1} b_i a_i}(X) \mod p$. Then, the factor of the session key can be derived $W_i = V_i / V_{i-1} = (T_{a_{i-1} b_{i-1} a_i}(X) \mod p)/(T_{a_{i+1} b_i a_i}(X) \mod p)$. Finally, the message $M_3 = \{UID_i, W_i, T_3\}$ is broadcasted to all users.

Step 4 U_i verifies T_3 after receiving M_3, and then derives the session key sk_i by the following process:

$$\begin{aligned} sk_i &= (V_i)^n \times (W_{i+1})^{n-1} \times (W_{i+2})^{n-2} \times \ldots \times (W_{i-1}) \\ &= V_1 \times V_2 \times \ldots \times V_n. \end{aligned}$$

The authentication information $Auth_{i1} = h(S_n \| sk_i \| UID_i \| T_3)$ applied by other members, and the authentication information $Auth_{i2} = h(K_{GS_i} \| S_n \| V_i \| T_3)$ applied by GWN can be derived. U_i broadcasts authentication information $M_4 = \{Auth_{i1}, Auth_{i2}\}$ to other users.

Step 5 After receiving M_4, any other member $U_j, \forall i \neq j$, can authenticate U_i by $Auth_{i1}$, and GWN can authenticate U_i by $Auth_{i2}$. Eventually, the session key of this meeting can be generated $SK = h(S_n, sk_i)$.

When each participant obtains SK, they can start to communicate with each other via encrypting/decrypting messages by SK.

In PL-GAKA, we apply a chaotic map to reduce the computation cost from SGPAKE. The process of key agreement can be finished early, and the meeting members can build a safety communication. For the security, each participant applies a semigroup property shown in Definition 3 to compute the factor of session key as shown in Step 2. The messages required by the process of key agreement can be sent via the Internet. In summary, the proposed PL-GAKA requires low computation cost but provides similar security level to the Diffie–Hellman problem in a convenient cloud meeting.

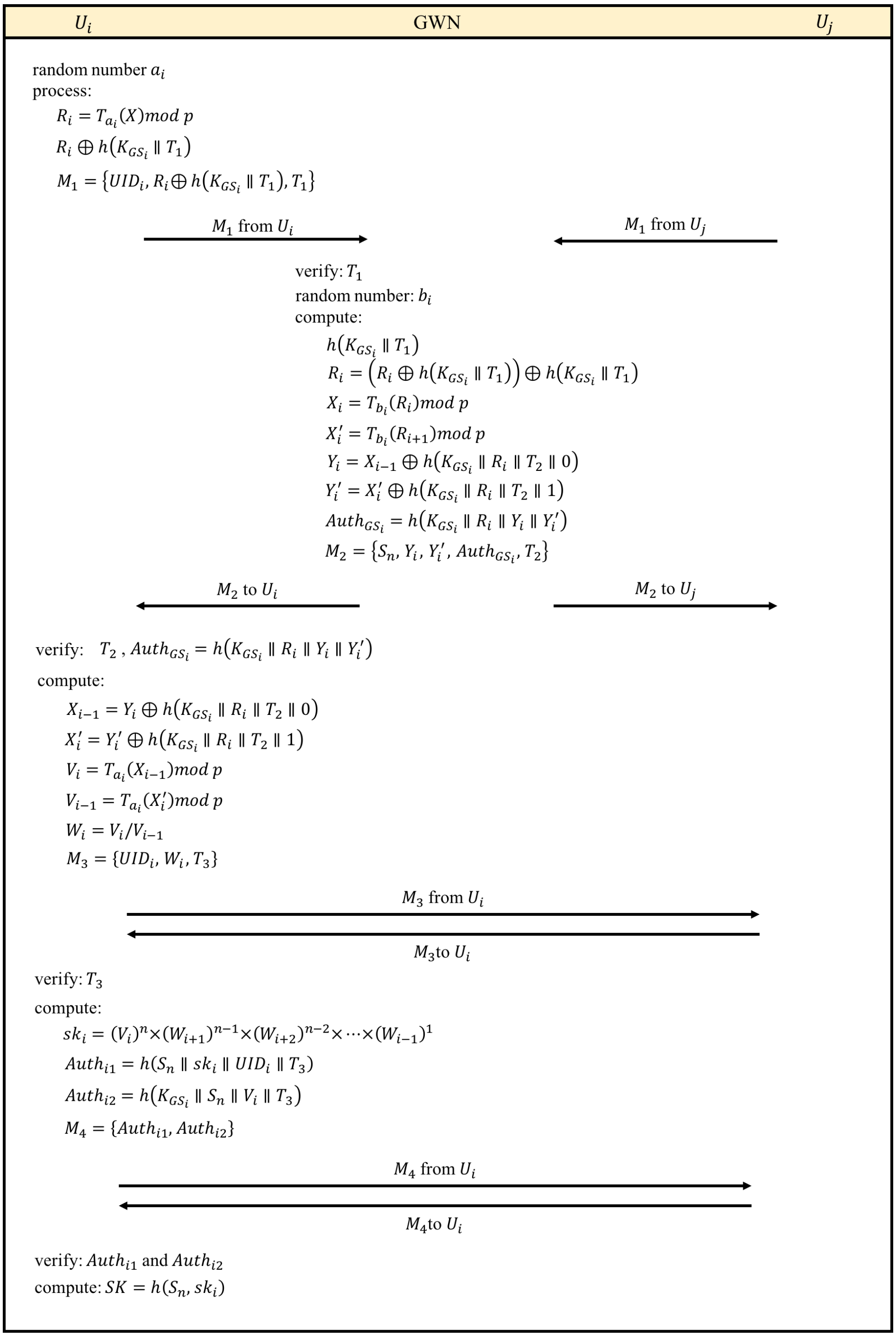
U_i
GWN
U_j
random number a_i
process:
$R_i = T_{a_i}(X) mod\ p$
$R_i \oplus h(K_{GS_i} \parallel T_1)$
$M_1 = \{UID_i, R_i \oplus h(K_{GS_i} \parallel T_1), T_1\}$
M_1 from U_i
M_1 from U_j
verify: T_1
random number: b_i
compute:
$h(K_{GS_i} \parallel T_1)$
$R_i = \left(R_i \oplus h(K_{GS_i} \parallel T_1)\right) \oplus h(K_{GS_i} \parallel T_1)$
$X_i = T_{b_i}(R_i) mod\ p$
$X_i' = T_{b_i}(R_{i+1}) mod\ p$
$Y_i = X_{i-1} \oplus h(K_{GS_i} \parallel R_i \parallel T_2 \parallel 0)$
$Y_i' = X_i' \oplus h(K_{GS_i} \parallel R_i \parallel T_2 \parallel 1)$
$Auth_{GS_i} = h(K_{GS_i} \parallel R_i \parallel Y_i \parallel Y_i')$
$M_2 = \{S_n, Y_i, Y_i', Auth_{GS_i}, T_2\}$
M_2 to U_i
M_2 to U_j
verify: T_2, $Auth_{GS_i} = h(K_{GS_i} \parallel R_i \parallel Y_i \parallel Y_i')$
compute:
$X_{i-1} = Y_i \oplus h(K_{GS_i} \parallel R_i \parallel T_2 \parallel 0)$
$X_i' = Y_i' \oplus h(K_{GS_i} \parallel R_i \parallel T_2 \parallel 1)$
$V_i = T_{a_i}(X_{i-1}) mod\ p$
$V_{i-1} = T_{a_i}(X_i') mod\ p$
$W_i = V_i / V_{i-1}$
$M_3 = \{UID_i, W_i, T_3\}$
M_3 from U_i
$M_3$to U_i
verify: T_3
compute:
$sk_i = (V_i)^n \times (W_{i+1})^{n-1} \times (W_{i+2})^{n-2} \times \cdots \times (W_{i-1})^1$
$Auth_{i1} = h(S_n \parallel sk_i \parallel UID_i \parallel T_3)$
$Auth_{i2} = h(K_{GS_i} \parallel S_n \parallel V_i \parallel T_3)$
$M_4 = \{Auth_{i1}, Auth_{i2}\}$
M_4 from U_i
$M_4$to U_i
verify: $Auth_{i1}$ and $Auth_{i2}$
compute: $SK = h(S_n, sk_i)$

Figure 3. The authentication process.

4. Performance Analysis

We analyze the proposed solution in terms of the correctness, the security and the overall comparison with related solutions. For the security verification, we refer to [8,18–22] to evaluate session key security, mutual authentication, perfect forward security, and data integrity. Moreover, we also demonstrate that the proposed solution is safe when suffering replay attacks, impersonation attacks, privileged insider attacks, and stolen-verifier attacks.

4.1. Correctness

If each U_i computes sk_i correctly, it implies that all members have security communications in the cloud meeting. Therefore, we trace the process of generating sk_i, and the resuls are correct:

$$
\begin{aligned}
sk_i &= (T_{a_{i-1}b_{i-1}a_i}(x) \mod p)^n & \times\left(\frac{T_{a_ib_ia_{i+1}}(x) \mod p}{T_{a_{i-1}b_{i-1}a_i}(x) \mod p}\right)^{n-1} \times \left(\frac{T_{a_{i+1}b_{i+1}a_{i+2}}(x) \mod p}{T_{a_ib_ia_{i+1}}(x) \mod p}\right)^{n-2} \\
& & \times \ldots \times \left(\frac{T_{a_{i+n-1}b_{i+n-1}a_{i+n}}(x) \mod p}{T_{a_{i+n-2}b_{i+n-2}a_{i+n-1}}(x) \mod p}\right) \\
&= (T_{a_{i-1}b_{i-1}a_i}(x) \mod p) & \times(T_{a_ib_ia_{i+1}}(x) \mod p) \times (T_{a_{i+1}b_{i+1}a_{i+2}}(x) \mod p) \\
& & \times \ldots \times (T_{a_{i+n-1}b_{i+n-1}a_{i+n}}(x) \mod p) \\
&= (T_{a_1b_1a_2}(x) \mod p) & \times(T_{a_2b_2a_3}(x) \mod p) \times (T_{a_3b_3a_4}(x) \mod p) \\
& & \times \ldots \times (T_{a_nb_na_{n+1}}(x) \mod p).
\end{aligned}
$$

4.2. Security Analysis

4.2.1. Session Key Security

U_i uses the session key to encrypt the information sending over Internet. Therefore, if the session key is secure, it means that the communication in the cloud meeting is also security. The proposed solution has the Diffie–Hellman problem. Even if attackers capture $T_{a_i}(x)$ or $T_{b_i}(x)$, they still can not generate authentication information. Moreover, we consider random value a_i and b_i, so it is difficult for attackers to compute sk_i and $SK = h(S_n, sk_i)$. Therefore, the session key is security in PL-GAKA.

4.2.2. Mutual Authentication

In the authentication process, the authentication information is used to verify members and GWN. In PL-GAKA, each member uses $Auth_{GS_i}$ and $Auth_{i1}$ to verify GWN and other members while GWN uses $Auth_{i2}$ to verify participants. Even if attackers can capture the identity and K_{GS_i}, respectively, and then generate $Auth_{GS_i}$ and M_4, each member must be authenticated by all other members and GWN. Therefore, the PL-GAKA is secure under the multi-authentication consideration.

4.2.3. Perfect Forward Security

Considering a situation in which attackers have the ability to capture the session key, they can use the session key to decrypt the information sending during cloud meetings. For example, a web user uses a username and a password to log in to a web service. If someone knows the username and the password, he/she can log in to the same web service and use it.

PL-GAKA does not take username and password into account for each meeting member. In each meeting, we use $sk_i = (V_i)^n \times (W_{i+1})^{n-1} \times (W_{i+2})^{n-2} \times \ldots \times (W_{n-1})$ to compute the session key $SK = h(S_n, sk_i)$. In other words, even if the session key is captured by attackers, the cloud meeting is still secure during the cloud meeting.

4.2.4. Data Integrity

When the information is modified by attackers, we say that the protocol has data integrity if each member can recognize the correctness of the received data. In PL-GAKA, if $R_i \oplus h(K_{GS_i} \| T_1)$ in M_1 is tampered with, GWN can use $h(K_{GS_i} \| T_1)$ to capture R_i. If W_i in M_3 is tampered with, other members will derive an unmatched sk_i. Therefore, the proposed protocol satisfies data integrity.

4.2.5. Replay Attack

Attackers can eavesdrop on the packets sending over Internet to capture the communication information. Then, attackers send the captured information again to be an authenticated user. This is the replay attack. If the mechanism can not detect replay attack, someone can counterfeit an authentication member.

In the proposed solution, we consider the timestamp for each message. If attackers counterfeit an authentication member and resend the message again, the timestamp can be used to capture the irrationality. Thus, the replay attack is useless in PL-GAKA.

4.2.6. Impersonation Attack

Impersonation attack means that illegal users impersonate legal ones and pass the authentication process with the stolen authenticated message to enter the system.

In the proposed group authenticated key agreement mechanism, the attacker can not obtain the authenticated message of K_{GS_i} because K_{GS_i} is encrypted. Without K_{GS_i}, the attacker can not impersonate U_i or GWN. Therefore, PL-GAKA can defend impersonation attacks.

4.2.7. Privileged-Insider Attack

Privileged-insider attack means that an authentication member impersonates other legal users with his/her own authenticated message. U_i in PL-GAKA gets K_{GS_i} from GWN in a safety tunnel in the registration process. Since different members will have various K_{GS_i}, no member can use his/her own K_{GS_i} to impersonate the other one. Hence, this mechanism can defend privileged-insider attack.

4.2.8. Stolen-Verifier Attack

Some protocol considers static verification data, which is saved in the server for authenticating members. Attackers steal the verification data from authentication servers, so that the attackers are authenticated by the verification data. Each member in the proposed solution is verified by other members and GWN, so verification data is not necessary. Therefore, the stolen-verifier attack is useless for the PL-GAKA.

4.2.9. Shared Device

Sharing a communication device, e.g., cell phone or tablet, is a common behavior between friends. In our scenario, if the encryption and decryption protocols are implemented in the specific communication device, the sharing device may be a security issue. PL-GAKA requires users to provide the identity as shown in several processes, such as generating M_1 and M_3. If a sharing device is used in PL-GAKA, the impersonator still can not join the cloud meeting due to the lack of identity. Therefore, sharing a device does not work in PL-GAKA.

4.2.10. Man-in-the-Middle Attack

During the key generation process, man-in-the-middle attack means that there is an attacker who builds a pair of connections with a specific sender and receiver. In other words, all messages sent from sender to receiver will be relayed by the attacker, and the attacker can access all the information of sender and receiver.

Man-in-the-middle attack is useless in the PL-GAKA, and we have the following properties to prove this claim. First, each member uses his/her unique UID_i in the registration and authentication processes. Thus, generating UID_i is an essential requirement. Second, each member must register in the GWN by the UID_i. The attacker has to be verified by GWN. Third, S_n is considered in Step 2 of authentication process. In other words, each meeting member must be verified by each other. Putting the above together, PL-GAKA avoids a man-in-the-middle attack.

4.3. Security Analysis via BAN Logic

We apply Burrows-Abadi-Needham (BAN) logic to verify the security of PL-GAKA in a formal analysis. PL-GAKA consists of registration and authentication phrases. Since registration phrase can be processed in a safety tunnel, we focus on the analysis in terms of the authentication phrase.

PL-GAKA is a group key authentication scheme, and some cloud meeting members will exchange messages between each member and GWN. To simplify the communication model, we generalize a meeting communication to the model with GWN and two members u_i and u_j. There are some concurrent processes in the authentication of PL-GAKA. For example, each member sends the identity message to GWN that all members send M_1 to GWN, and we consider a simple case that u_i and u_j send M_1 to GWN simultaneously. Moreover, M_3 and M_4 will be broadcasted to all members, and we consider the case that u_i sends M_3 to u_j while u_j sends M_4 to u_i. Therefore, we can generalize the communication model to a simple one, as shown in Figure 4.

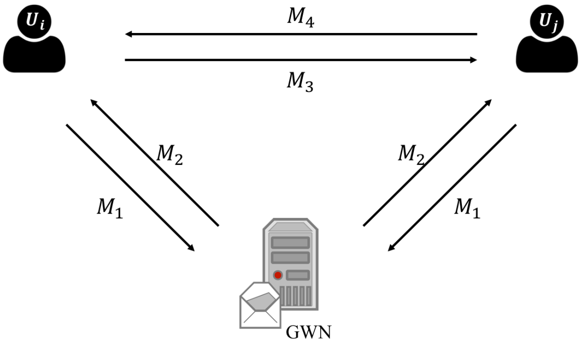

Figure 4. The message delivery structure in the authentication process of PL-GAKA.

After registering in GWN, each u_i has the initial state including UID_i, K_{GS_i}, and a timestamp generator. According to Figure 4, we have the following processes. Note that both u_i and u_j sends M_1 to GWN while GWN responses M_2 to u_i and u_j, and we just focus on the notation on the communication between u_i and GWN.

P_1 $said(u_i, M_1)$: u_i sends M_1.
P_2 $sees(GWN, M_1)$: GWN receives M_1.
P_3 $said(GWN, M_2)$: GWN sends M_2.
P_4 $sees(u_i, M_2)$: u_i receives M_2.
P_5 $said(u_i, M_3)$: u_i sends M_3.
P_6 $sees(u_j, M_3)$: u_j receives M_3.
P_7 $said(u_j, M_4)$: u_j sends M_4.
P_8 $sees(u_i, M_4)$: u_i receives M_4.

Here, we have the following assumptions:

A_1 $bel(GWN, cont(u_i, M_1))$: GWN believes that he/she has the ability to confirm M_1 sent from u_i.
A_2 $bel(GWN, goodinfo(u_i, M_1, GWN))$: GWN believes that M_1 sent from u_i to GWN is confirmed.
A_3 $bel(u_i, cont(GWN, M_2))$: u_i believes that he/she has the ability to confirm M_2 sent from GWN.
A_4 $bel(u_i, goodinfo(GWN, M_2, u_i))$: u_i believes that M_2 sent from GWN to u_i is confirmed.
A_5 $bel(u_j, cont(u_i, M_3))$: u_j believes that he/she has the ability to confirm M_3 sent from u_i.
A_6 $bel(u_j, goodinfo(u_i, M_3, u_j))$: u_j believes that M_3 sent from u_i to u_j is confirmed.
A_7 $bel(u_i, cont(u_j, M_4))$: u_i believes that he/she has the ability to confirm M_4 sent from u_j.
A_8 $bel(u_i, goodinfo(u_j, M_4, u_i))$: u_i believes that M_4 sent from u_j to u_i is confirmed.
A_9 $bel(GWN, fresh(T_1))$: GWN believes that T_1 is fresh.

A_{10} $bel(u_i, fresh(T_2))$: u_i believes that T_2 is fresh.
A_{11} $bel(u_j, fresh(T_3))$: u_j believes that T_3 is fresh.

Thus, we have the following goals:

G_1 $bel(UID_i, R_i \oplus h(K_{GS_i} \| T_1), fresh(T_1))$. $GWN \rightarrow u_i$: M_1 sent from u_i to GWN is correct and fresh.
G_2 $bel(S_n, Y_i, Y_i', Auth_{GS_i}, fresh(T_2))$. $u_i \rightarrow GWN$: M_2 sent from GWN to u_i is correct and fresh.
G_3 $bel(UID_i, W_i, fresh(T_3))$. $u_j \rightarrow u_i$: M_3 sent from u_i to u_j is correct and fresh.
G_4 $bel(Auth_{i1}, Auth_{i2})$. $u_i \rightarrow u_j$: M_4 sent from u_j to u_i is correct and fresh.

From the believe connection, each goal can be achieved:

G_1: From P_1, P_2, A_1, A_2, and A_9, M_1 is correct and fresh.
G_2: From P_3, P_4, A_3, A_4, and A_{10}, M_2 is correct and fresh.
G_3: From P_5, P_6, A_5, A_6, and A_{11}, M_3 is correct and fresh.
G_4: From P_7, P_8, A_7, and A_8, M_4 is correct and fresh.

Since each goal can be achieved, PL-GAKA provides a secure session key generation.

4.4. Security Comparison

The overall comparison between PL-GAKA and related approaches are shown in Table 2. We refer to [8] for considering the following protocols:

- Protocol #1 proposed by Abdalla and Pointcheval is a group password-based key agreement [15].
- Protocol #2 proposed by Dutta and Barua is a group password-based authentication key agreement [16].
- Protocol #3 proposed by Kim et al. is a group key agreement [3].
- Protocol #4 proposed by Boyd and Nieto is a group key agreement [17].
- Protocol #5 proposed by Lee et al. is a group password-based authentication key agreement [8].

Table 2. The overall comparison between the proposed solution and related approaches.

Protocol	Protocol #1	Protocol #2	Protocol #3	Protocol #4	Protocol #5	PL-GAKA
Public Key	No	No	Yes	Yes	No	No
Private Key	shared password	shared password	PKI-based	PKI-based	Yes	No
Asymmetric Encryption	No	No	No	Yes	No	No
Symmetric Encryption	Yes	Yes	No	No	Yes	No
Signature Verification	No	No	Yes	Yes	No	No
Mutual Authentication	No	Yes	No	No	Yes	Yes
Perfect Forward Security	Yes	No	Yes	No	Yes	Yes

PKI: Public Key Infrastructure.

For the security consideration, PL-GAKA takes into account the extended chaotic map to improve the computation efficiency from SGPAKE. Although the extended chaotic map does not provide the Diffie–Hellman problem, we still can derive an equivalent security level by extended Chebyshev polynomials. Therefore, solving the message generated by the extended chaotic map requires similar computing resource to that in the Diffie–Hellman problem. Therefore, the security level gap between PL-GAKA and SGPAKE is small.

4.5. Efficiency Comparison

The results of the efficiency comparison between SGPAKE and PL-GAKA are illustrated in Table 3. Since this paper focuses on the cloud meeting and improves SGPAKE in the cloud meeting, we compare

PL-GAKA with SGPAKE. For the Exponentiation evaluation, SGPAKE requires $4(2^a)$ because of two modular exponential computations for generating session keys. According to the properties of cloud meetings, the participant list can be determined before PL-GAKA starts, so the heavy work can be well prepared, and the computation cost can be finished from an offline computation.

Table 3. The efficiency comparison between SGPAKE and PL-GAKA.

Protocol	SGPAKE	PL-GAKA
Password Maintenance	Yes	No
Exponentiation	Yes	No
Key Calculation	Modular Exponentiation	Extented Choatic Map

For the efficiency of the session key calculation process, PL-GAKA considers the extended chaotic map, which is a lightweight calculation compared with the modular exponential computation. Thus, PL-GAKA requires less computation time to generate a session key than that of SGPAKE. On the other hand, the meeting member does not require a password to verify the identity in PL-GAKA, so the password maintenance mechanism is not necessary in Pl-GAKA, but it is required in SGPAKE. Putting the above together, PL-GAKA is more efficient than SGPAKE in terms of key generation and the user maintenance.

5. Conclusions

Group authentication key agreement is necessary for providing security communications, and a cloud meeting is a typical and popular application. Lee et al. present SGPAKE to realize the secure group communication. However, SGPAKE is a Diffie–Hellman key exchange scheme, and the heavy computation cost is an implementation issue. We consider SGPAKE and apply the extended chaotic map to propose a password-less group authentication key agreement named PL-GAKA. Since an extended chaotic map provides properties that are similar to semigroup in chaotic map, the security of PL-GAKA is equivalent to that of SGPAKE. PL-GAKA is a password-less protocol, so each user does not worry about the password maintenance. Moreover, the session key is dynamic in each cloud meeting. In other words, PL-GAKA considers short-term authentication, and it provides stronger security than other long-term authentication protocols. In the future, we will focus on the progress on improving the registration security of the meeting members coming from various companies, and consider sharing devices.

When a cloud meeting takes place, only the registered users can be invited to join the meeting. In the real world applications, the registration can be finished when a new staff member is reported to the company, and the entire process can be done in a secure procedure. It means that the meeting members must be employed in the same company in PL-GAKA. In other words, the registration process must be improved for staff members from different companies that do not have consistent registration processes.

Acknowledgments: The authors would like to thank to reviewers for their valuable suggestions to improve the manuscript.

Author Contributions: Tsung-Hung Lin, Tian-Fu Lee, and Zeng-Bo Wang analyze the requirements and design the mechanism; Chen-Kun Tsung and Tsung-Hung Lin analyze the proposed mechanism; Chen-Kun Tsung and Tsung-Hung Lin revises the paper and reports the suggestion reply.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GWN	Trust Authentication Server
SGPAKE	Simple Group Password-based Authenticated Key Agreement
PKI	Public Key Infrastructure
PL-GAKA	Passwordless Group Authentication Key Agreement

References

1. Feng, Y.; Li, B.; Li, B. Airlift: Video Conferencing as a Cloud Service Using Inter-datacenter Networks. In Proceedings of the IEEE International Conference on Network Protocols, Austin, TX, USA, 30 October–2 November 2012; pp. 1–11.
2. Glitho, R.H. Cloud-based Multimedia Conferencing: Business Model, Research Agenda, State-of-the-Art. In Proceedings of the IEEE 13th Conference on Commerce and Enterprise Computing, Luxembourg, 5–7 September 2011; pp. 226–230.
3. Kim, H.J.; Lee, S.M.; Lee, D.H. Constant-round Authenticated Group Key Exchange for Dynamic Groups. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, 5–9 December 2004; pp. 245–259.
4. Li, J.; Guo, R.; Zhang, X. Study on Service-oriented Cloud Conferencing. In Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology, Chengdu, China, 9–11 July 2010; Volume 6, pp. 21–25.
5. Lee, T.-F.; Hwang, T. Improvement of the Round-optimal Conference Key Agreement Protocol of Boyd and Nieto. In Proceedings of the 16th Information Security Conference, Taipei, Taiwan, 8 June 2006; pp. 98–102.
6. Lee, T.-F.; Wen, H.-A.; Hwang, T. A weil Pairing-based Roundefficient and Fault-tolerant Group Key Agreement Protocol for Sensor Networks. In *Sensor Network Operations*; IEEE Press: Piscataway, NJ, USA, 2006; pp. 571–579.
7. Lee, T.-F.; Wen, H.-A.; Jin, Y.-C.; Chen, C.-S. Password-based Group Key Agreement with Server's Public Key for Hypergraphs. In Proceedings of the Symposium on Applications of Information, Management and Communication Technology, Kaohsiung, Taiwan; 13 June 2008.
8. Lee, T.F.; Chang, I.P.; Wang, C.C. Simple Group Password-based Authenticated Key Agreements for the Integrated EPR Information System. *J. Med. Syst.* **2013**, *37*, 1–6.
9. Zhu, H.F. Secure Chaotic Maps-based Group Key Agreement Scheme with Privacy Preserving. *Int. J. Netw. Secur.* **2016**, *18*, 1001–1009.
10. Zhang, L. Cryptanalysis of the Public Key Encryption based on Multiple Chaotic Systems. *Chaos Solitons Fractals* **2008**, *37*, 669–674.
11. Guo, C.; Chang, C.C. Chaotic Maps-based Password-authenticated Key Agreement using Smart Cards. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 1433–1440.
12. Mishkovski, I.; Kocarev, L. *Chaos-Based Cryptography: Theory, Algorithms and Applications*; Springer: Berlin, Germany, 2011; pp. 53–54.
13. Zhu, H.F.; Zhu, D.; Zhang, Y. Using Chaotic Maps to Construct Anonymous Multi-receiver Scheme Based on BAN Logic. *J. Inf. Hiding Multimed. Signal Process.* **2016**, *7*, 685–696.
14. Zhu, H.F.; Zhang, Y.; Xia, Y.; Li, H. Password-Authenticated Key Exchange Scheme Using Chaotic Maps towards a New Architecture in Standard Model. *Int. J. Netw. Secur.* **2016**, *18*, 326–334.
15. Abdalla, M.; Pointcheval, D. Simple Password-based Authenticated Key Protocols. In *Topics in Cryptology - CT-RSA 2005. LNCS*; Springer-Verlag: Berlin, Germany, 2005; Volume 3376, pp. 191–208.
16. Dutta, R.; Barua, R. Password-based Encrypted Group Key Agreement. *Int. J. Inf. Secur.* **2006**, *3*, 23–34.
17. Boyd, C.; Nieto, J.M.G. Round-optimal Contributory Conference Key Agreement. In *Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 3, pp. 161–174.
18. Zhang, L.; Tang, S.; Cai, Z. Efficient and Flexible Password Authenticated Key Agreement for Voice over Internet Protocol Session Initiation Protocol using Smart Card. *Int. J. Commun. Syst.* **2014**, *27*, 2691–2702.
19. Cheng, Z.Y.; Liu, Y.; Chang, C.C.; Chang, S.C. A Practical Secure Chaos-Based Group Key Agreement Protocol Suitable for Distributed Network Environment. *Int. J. Innov. Comput. Inf. Control* **2013**, *9*, 1935–1949.
20. Bresson, E.; Chevassut, O.; Pointcheval, D. Provably Authenticated Group Diffie–Hellman Key Exchange – the Dynamic Case. *Adv. Cryptol. ASIACRYPT* **2001**, *2248*, 290–309.

21. Bresson, E.; Chevassut, O.; Pointcheval, D. Group Diffie–Hellman Key Exchange Secure Against Dictionary Attacks. *Adv. Cryptol. ASIACRYPT* **2002**, 603–610, doi:10.1007/3-540-36178-2_31.
22. Bresson, E.; Chevassut, O.; Pointcheval, D. Dynamic Group Diffie–Hellman Key Exchange under Standard Assumptions. *Adv. Cryptol. EUROCRYPT* **2002**, 321–336, doi:10.1007/3-540-46035-7_21.
23. Boneh, D. The Decision Diffie–Hellman Problem. In Proceedings of the International Algorithmic Number Theory Symposium, Portland, OR, USA, 21–25 June 1998; pp. 48–63.
24. Barbulescu, R.; Gaudry, P.; Joux, A.; Thomé, E. A Heuristic Quasi-Polynomial Algorithm for Discrete Logarithm in Finite Fields of Small Characteristic. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, 11–15 May 2014; pp. 1–16.
25. Wang, X.; Zhao, J. An Improved Key Agreement Protocol based on Chaos. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 4052–4057.