Security weakness of Tseng's fault-tolerant conference-key agreement protocol[☆]

Sangho Lee*, Jong Kim, Sung Je Hong

Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Hyoja-dong, Nam-gu, Pohang, Korea

Abstract

A fault-tolerant conference-key agreement protocol establishes a shared key among participants of a conference even when some malicious participants disrupt key agreement processes. Recently, Tseng proposed a new fault-tolerant conference-key agreement protocol that only requires a constant message size and a small number of rounds. In this paper, we show that the Tseng's protocol cannot provide forward and backward confidentiality during a conference session for the proposed attack method. We also show that a simple countermeasure — re-randomizing short-term keys of some participants — to avoid the proposed attack can be broken by extending the proposed attack method.

Key words: Network-based conference, Fault-tolerant key agreement, Forward/Backward confidentiality, Security attack

1. Introduction

A network-based conference becomes common in the era of information technology because it can save time and money, and lead to people participate easily. However, network is a shared medium so that it is weak to security attacks such as eavesdropping and message forging. Thus, we have to protect network-based conferences when the conferences deal with private and important information.

A general method to assure confidentiality of a conference is encryption of messages with a shared key among participants. Two types of key

[☆]This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute for Information Technology Advancement) (IITA-2009-C1090-0901-0045).

*Corresponding author. Tel.: +82 54 279 2915. Fax: +82 54 279 1805

Email addresses: sangho2@postech.ac.kr (Sangho Lee), jkim@postech.ac.kr (Jong Kim), sjhong@postech.ac.kr (Sung Je Hong)

sharing schemes exist: key distribution schemes (Chang et al., 1992; Hirose and Ikeda, 1997; Hwang and Yang, 1995; Tseng, 2002) and key agreement schemes (Ateniese et al., 2002; Boyd and Nieto, 2003; Bresson et al., 2001, 2002; Burmester and Desmedt, 1994; Katz and Yung, 2003). In the key distribution scheme, a key distributor establishes a key and then securely distributes it to every participant. The advantages of this scheme are simplicity, and low computational and communication costs. However, it requires a key distributor which should be a trusted third party (TTP). In the key agreement scheme, every conference attendee participates to establish a shared key. This scheme is relatively complex and requires high computational and communication costs. However, it does not require a TTP and is more secure because the shared key is revealed to the participants only. This paper is focused on the analysis of a previously proposed fault-tolerant conference-key agreement scheme.

Simple key agreement schemes cannot assure key creations if malicious participants disrupt key agreement processes. To solve this problem, fault-tolerant conference-key agreement protocols are proposed to assure honest participants can acquire a conference-key no matter how many malicious participants exist (Shi et al., 2004; Tseng, 2005, 2007; Tzeng, 2002; Yi, 2004). Especially, Tseng's fault-tolerant conference-key agreement protocol provides fault-tolerance and forward secrecy with a constant message size and round efficiency (Tseng, 2007). However, in this paper, we propose an attack method to the Tseng's protocol to compromise following confidentiality during a conference session: forward confidentiality to a malicious participant, backward confidentiality to a joining participant, and forward confidentiality to a leaving participant. The proposed attack method requires a small number of computations, one exponentiation, two multiplications, and one division. We also show that a simple fix to eliminate the security weakness of Tseng's protocol is hard to find and not yet available by introducing a simple countermeasure against the proposed attack and then breaking that countermeasure with an extended attack method.

The remainder of this paper is organized as follows: In Section 2, we introduce the preliminaries of this paper, and in Section 3, we briefly review the Tseng's fault-tolerant conference-key agreement protocol. In Section 4, we introduce the security weakness of the Tseng's protocol, and in Section 5, we show a simple countermeasure to the proposed attack can be broken by extending the proposed attack. Finally, we conclude this paper in Section 6.

2. Preliminaries

We adopt the system model, definitions, and notations of Tseng's work (Tseng, 2007).

2.1. System model and assumptions

In the system, each user has a secret and public key pair. The system has a public directory that records public system parameters and each user's public key information. All users connect using a broadcast network, and they can broadcast messages to each other. In the broadcast network, communications among users cannot be altered, blocked, or delayed. Because the broadcast channel is public, anyone can sniff messages transmitted over the broadcast channel. Therefore, when users want to transmit secure messages to each other over the broadcast network, they must establish a conference-key to encrypt their communications.

2.2. Definitions

Definition 1 (Malicious participant)**.** A malicious participant is an adversary who attends a conference-key agreement protocol to disrupt the establishment of the conference-key.

Definition 2 (Forward confidentiality)**.** A forward confidentiality is a security requirement that when a participant leaves from a conference session, the participant could not decrypt the future encrypted messages of the conference session.

Definition 3 (Backward confidentiality)**.** A backward confidentiality is a security requirement that when a new participant joins into a conference session, the participant could not decrypt the past encrypted messages of the conference session.

2.3. Notations

- U_i: an ith participant.
- U: a set of participants, $U = \{U_1, U_2, \ldots, U_n\}$ when n participants exist.
- q: a large prime number.
- p: a large prime number such that $p = 2q + 1$.
- G_q: a subgroup of quadratic residues in $\mathbb{Z}_p^*$, that is $G_q = \left\{i^2 | i \in \mathbb{Z}_p^*\right\}$.
- g: a generator for the subgroup G_q.
- H: a one-way function from $\mathbb{Z}$ to $\mathbb{Z}_q$.
- x_i: U_i's secret key which is a random value in $\mathbb{Z}_p^*$.
- y_i: U_i's public key such that $y_i = g^{x_i} \bmod p$.
- M: a message which includes the valid time period to deter the replay attack.

The indices of U_i are taken in a cycle if there is no member joining or leaving in the member set. So, $U_{n+1} = U_1$ and $U_0 = U_n$.

3. Tseng's fault-tolerant conference-key agreement protocol

This section briefly reviews the Tseng's fault-tolerant conference-key agreement protocol. Details about the protocol can be found in (Tseng, 2007).

3.1. Conference-key agreement protocol

The Tseng's protocol consists of three-step conference-key agreement and the fault detection and correction phase.

Step 1. Short-term key distribution. Each participant U_i selects two short-term secret keys k_i and v_i in $\mathbb{Z}_q^*$ randomly, and computes $w_i = g^{k_i} \bmod p$, $A_i = g^{v_i} \bmod p$, and $B_i = v_i^{-1}(H(w_i, M) - A_i x_i) \bmod q$. Then each U_i broadcasts (w_i, A_i, B_i, M) to other participants.

Step 2. Sub-key distribution. After receiving all (w_j,A_j,B_j,M), $(1 \leq j \leq n, j \neq i)$, each U_i checks $g^{H(w_j,M)} \stackrel{?}{=} y_j^{A_j} A_j^{B_j} \bmod p$ to validate that each w_j is really issued by U_j. Each U_i also checks $2 \leq w_j \leq p-1$ and ${w_j}^q \bmod p \stackrel{?}{=} 1$ to validate that w_j is a generator of G_q. If above checks do not hold, U_i broadcasts "$V_{ij} = failure$". Else U_i selects a random number $r_i \in \mathbb{Z}_q^*$ and computes $z_i = (w_{i+1}/w_{i-1})^{k_i} \bmod p$, $\alpha_i = g^{r_i} \bmod p$, $\beta_i = (w_{i+1}/w_{i-1})^{r_i} \bmod p$, and $\delta_i = r_i + H(z_i, \alpha_i, \beta_i, M)k_i \bmod q$. Then each U_i broadcasts $(z_i, \alpha_i, \beta_i, \delta_i)$ to other participants.

Step 3. Sub-key verification and conference-key computation. After receiving all $(z_j, \alpha_j, \beta_j, \delta_j)$, each U_i checks $g^{\delta_j} \stackrel{?}{=} \alpha_j w_j^C \bmod p$ and $(w_{j+1}/w_{j-1})^{\delta_i} \stackrel{?}{=} \beta_j z_j^C \bmod p$, where $(1 \leq j \leq n, j \neq i)$ and $C = H(z_j, \alpha_j, \beta_j, M)$. If two checks do not hold, U_i broadcasts "$V_{ij} = failure$". Else U_i can use k_i selected in Step 2 to compute the conference-key K by $K = w_{i-1}^{nk_i} z_i^{n-1} z_{i+1}^{n-2} \cdots z_{i-2} \bmod p$ $= g^{k_1k_2+k_2k_3+\cdots+k_nk_1} \bmod p$.

Fault detection and correction. In Step 2, each U_i validates all w_j to detect and remove a malicious participant U_j who sends wrong w_j to other participants. Only the pre-position and post-position participants of U_j, U_{j-1} and U_{j+1}, need to re-compute $(z_{j-1},\alpha_{j-1},\beta_{j-1},\delta_{j-1})$ using (w_{j-2}, w_{j+1}) and $(z_{j+1},\alpha_{j+1},\beta_{j+1},\delta_{j+1})$ using (w_{j-1}, w_{j+2}), respectively. In Step 3, if any malicious participant U_j sends a wrong $(z_j, \alpha_j, \beta_j, \delta_j)$, U_j will be decided as malicious and be deleted from the participant set U. U_{j-1} and U_{j+1} re-compute $(z_{j-1}, \alpha_{j-1}, \beta_{j-1}, \delta_{j-1})$ and $(z_{j+1}, \alpha_{j+1}, \beta_{j+1}, \delta_{j+1})$. The remaining honest participants then check whether $(z_{j-1}, \alpha_{j-1}, \beta_{j-1}, \delta_{j-1})$ and $(z_{j+1}, \alpha_{j+1}, \beta_{j+1}, \delta_{j+1})$ are correct or not. The final participant set U' be $U' = \left\{U_1', U_2', \ldots, U_m'\right\}$, where $m \leq n$. At last, each U_i' computes the conference-key K by $K = {w'_{i+1}}^{mk_i'} {z'_i}^{m-1} {z'_{i+1}}^{m-2} \cdots {z'}_{i-2} \bmod p$ $= g^{k_1k_2+k_2k_3+\cdots k_nk_1} \bmod p$.

3.2. Joining/leaving procedures

The Tseng's protocol has joining/leaving procedures to an ongoing conference session.

Participant joining procedure. If a participant U_{n+1} wants to join a conference session, the conference-key K must be updated to ensure *backward confidentiality* against U_{n+1}. The joining procedure is

1. U_{n+1} broadcasts $(w_{n+1}, A_{n+1}, B_{n+1})$.
2. After receiving $(w_{n+1}, A_{n+1}, B_{n+1})$, other participants validate it. Only the pre-position and post-position participants of U_{n+1}, U_n and U_1, send (w_n, A_n, B_n) and (w_1, A_1, B_1) to the U_{n+1}, respectively. Later, U_n uses (w_{n-1}, w_{n+1}) to re-compute $(z_n, \alpha_n, \beta_n, \delta_n)$ and U_1 uses (w_{n+1}, w_2) to re-compute $(z_1, \alpha_1, \beta_1, \delta_1)$. Meanwhile, other participants U_i, $(2 \leq i \leq n-1)$, also send the previous message $(z_i, \alpha_i, \beta_i, \delta_i)$ to U_{n+1}. Finally, U_{n+1} uses (w_n, w_1) to compute and broadcast $(z_{n+1}, \alpha_{n+1}, \beta_{n+1}, \delta_{n+1})$ to all participants.
3. All participants can compute the new conference-key as in Step 3 of the conference-key agreement protocol.

Participant leaving procedure. If a participant U_j wants to leave a conference session, the conference-key K must be updated to ensure *forward confidentiality* against U_j. The leaving procedure is

1. Only the pre-position and post-position participants of U_j, U_{j-1} and U_{j+1}, use (w_{j-2}, w_{j+1}) and (w_{j-1}, w_{j+2}) to re-compute and broadcast $(z_{j-1}, \alpha_{j-1}, \beta_{j-1}, \delta_{j-1})$ and $(z_{j+1}, \alpha_{j+1}, \beta_{j+1}, \delta_{j+1})$, respectively.
2. All participants can compute the new conference-key as in Step 3 of the conference-key agreement protocol.

4. Security weakness of Tseng's protocol

In this section, we suggest an attack method that can compromise the following confidentiality of the Tseng's fault-tolerant conference-key agreement protocol during a conference session: the forward confidentiality to a malicious participant, the backward confidentiality to a joining participant, and the forward confidentiality to a leaving participant. The proposed attack method only requires a small number of computations (one exponentiation, two multiplications, and one division). So, it is applicable.

Claim 1. *Let U_j be a malicious participant to be excluded during a conference session. U_j is the participant who has already executed Step 2 fully before the exclusion and therefore has a conference key K. Even when pre-position and post-position participants of U_j update their keying materials to establish a new conference-key K' for the forward confidentiality against U_j in Step 3, U_j can obtain K' with a small number of computations by sniffing the updated keying materials.*

Proof. Let K be the conference-key which was made when U_j acts as an honest participant. To establish an updated conference-key $K^{'}$, U_{j-1} and U_{j+1} re-compute $z_{j-1}^{'}$ and $z_{j+1}^{'}$ respectively and broadcast them to other participants. [1] Since $K = g^{k_1k_2+\cdots+k_{j-1}k_j+\ k_jk_{j+1}+\cdots+k_nk_1} \bmod p$ and $K^{'} = g^{k_1k_2+\cdots+k_{j-1}k_{j+1}+\cdots+k_nk_1} \bmod p$,

$$K^{'}/K = g^{k_{j-1}k_{j+1}-k_{j-1}k_j-k_jk_{j+1}} \bmod p. \tag{1}$$

Since $z_{j-1} = (w_j/w_{j-2})^{k_{j-1}} = g^{k_{j-1}k_j-k_{j-2}k_{j-1}} \bmod p$ and $z_{j-1}^{'} = (w_{j+1}/w_{j-2})^{k_{j-1}}$ $= g^{k_{j-1}k_{j+1}-k_{j-2}k_{j-1}} \bmod p$,

$$z_{j-1}^{'}/z_{j-1} = g^{k_{j-1}k_{j+1}-k_{j-1}k_j} \bmod p. \tag{2}$$

By using Eqs. (1) and (2),

$$(K^{'}/K)(z_{j-1}/z_{j-1}^{'}) = g^{-k_jk_{j+1}} \bmod p = w_{j+1}^{-k_j} \bmod p. \tag{3}$$

According to Eq. (3), a malicious participant U_j can obtain $K^{'}$ by

$$K^{'} = K(z_{j-1}^{'}/z_{j-1})w_{j+1}^{-k_j} \bmod p. \tag{4}$$

This result shows that if a malicious participant has kept keying materials received from other participants and sniffs newly created $z_{j-1}^{'}$, he/she can establish the new conference-key $K^{'}$. Eq. (4) only requires one exponentiation, one division, and two multiplications when K was already computed. Therefore, a malicious participant U_j can obtain a new conference-key with a small number of computations by sniffing updated keying materials. □

Claim 2. *Let U_{n+1} be a participant who wants to join an ongoing conference session. Even when pre-position and post-position participants of U_{n+1} update their keying materials to establish a new conference-key $K^{'}$ and to hide an old conference-key K to U_{n+1} for the backward confidentiality, U_{n+1} can obtain K if it sniffed old keying materials previously.*

Proof. The old conference-key which was used before U_{n+1} joining is $K = g^{k_1k_2+k_2k_3+\cdots+k_nk_1} \bmod p$ and the new conference-key after U_{n+1} joining is $K^{'} = g^{k_1k_2+k_2k_3+\cdots+k_nk_{n+1}+k_{n+1}k_1} \bmod p$. In this condition, if U_{n+1} is leaving, then K can be considered as a new conference-key, again. Thus, computing K from $K^{'}$ to compromise the backward confidentiality is same to the procedure to compromise the forward confidentiality. Therefore, according to Claim 1, a joining participant U_{n+1} can obtain the old conference-key K by $K = K^{'}(z_n/z_n^{'})w_1^{-k_{n+1}} \bmod p$. □

[1] From here, we exclude A, B in Step 1 and α, β, δ in Step 2 for simplicity because they are not used to establish the updated conference-key.

Claim 3. *Let U_j be a participant who leaves an ongoing conference session. Even when pre-position and post-position participants of U_j update their keying materials to establish a new conference-key $K^{'}$ for the forward confidentiality against U_j, U_j can obtain $K^{'}$ by sniffing updated keying materials.*

Proof. Actually, the leaving process of the Tseng's protocol is same to the key regeneration process when a malicious participant is excluded from a conference session. Therefore, according to Claim 1, a leaving participant U_j can obtain the new conference-key $K^{'}$ by $K^{'} = K(z^{'}_{j-1}/z_{j-1})w^{-k_j}_{j+1} \bmod p$. □

Analysis of fault-tolerance features. The desired features of a fault-tolerant conference-key agreement protocol are 1) an assurance of a key establishment even when a malicious participant attempts to disrupt key agreement processes and 2) a protection of the established key from the malicious participant. A malicious participant will have a DoS type effect when it sent incorrect values in key agreement processes. However, a fault-tolerant conference-key agreement protocol will assure that the honest parties will reach to an agreement in establishing a conference-key. In this section, we presented Claims that the Tseng's protocol can reveal the established key. So, the feature 2) is no longer guaranteed in the Tseng's protocol.

5. Simple countermeasure and extended attack

A simple countermeasure to the attack method described in Section 4 is re-randomizing the short-term keys of some participants when a malicious participant is being excluded, a new participant is joining, or a participant is leaving. This countermeasure can solve the proposed attack method in Section 4. However, in this section, we propose an extended attack method to break the simple countermeasure. We show that to avoid the extended attack method, almost every honest participant should re-randomizes its short-term key. It means that the extended attack method makes the simple countermeasure the same as restarting the conference-key agreement protocol from the beginning.

5.1. Short-term key re-randomization

1. When faults, joining events, or leaving events are occurred, t participants, $U_{i_1}, U_{i_2}, \ldots, U_{i_t}$, re-select two short-term secret keys $(k^{'}_{i_j}, v^{'}_{i_j})$ in $\mathbb{Z}^*_q$ randomly $(1 \leq j \leq t)$, and then compute and broadcast $(w^{'}_{i_j}, A^{'}_{i_j}, B^{'}_{i_j})$.
2. After receiving all $(w^{'}_{i_j}, A^{'}_{i_j}, B^{'}_{i_j})$, $(i_j \neq h)$, every participant U_h in U validates $w^{'}_{i_j}$ and then computes and broadcasts $(z^{'}_h, \alpha^{'}_h, \beta^{'}_h, \delta^{'}_h)$ if U_{h-1} or U_{h+1} is a malicious participant, a joining participant, a leaving participant, or the one in $\{U_{i_1}, U_{i_2}, \ldots, U_{i_t}\}$.

3. All participants compute the new conference-key as in Step 3 of the conference-key agreement protocol in Section 3.

5.2. Attacking short-term key re-randomization

Lemma 1. *Let U_j be a malicious participant. Even when t participants ($t \leq n-3$) who are not pre-position and post-position participants of U_j re-randomize their secret short-term keys to establish a new conference-key $K^{'}$ for the forward confidentiality against U_j, U_j can obtain $K^{'}$.*

Proof. Let $U_{i_1}, U_{i_2}, \ldots, U_{i_t}$ be the participants who are not pre-position and post-position participants of U_j. They re-randomize their secret keys k to $k^{'}$. Without loss of generality, we assume that the first index larger than j is i_1 and following larger indices are $i_2, \ldots, i_t$ with round to smaller indices. Here, $K^{'}$ is

$$K^{'} = g^{k_1k_2+\cdots+k_{j-1}k_{j+1}+\cdots+k_{i_1-1}k^{'}_{i_1}+k^{'}_{i_1}k_{i_1+1}+\cdots+k_{i_t-1}k^{'}_{i_t}+k^{'}_{i_t}k_{i_t+1}+\cdots+k_nk_1} \bmod p.$$

Then, a malicious participant U_j can compute $\hat{K}_j$ that is an *intermediate-key* adopting the new keying materials of t participants and their pre-position and post-position participants, and reusing the old keying materials of the other participants including U_{j-1} and U_{j+1} by

$$\begin{aligned}\hat{K}_j =& w_{j-1}^n z_j^{n-1} z_{j+1}^{n-2} \cdots z_{i_1-2}^{n-(i_1-2-(j-1))} z^{'}_{i_1-1}{}^{n-(i_1-1-(j-1))} z^{'}_{i_1}{}^{n-(i_1-(j-1))} \\ & z^{'}_{i_1+1}{}^{n-(i_1+1-(j-1))} z_{i_1+2}^{n-(i_1+2-(j-1))} \cdots z_{i_t-2}^{n-(i_t-2-(j-1))} z^{'}_{i_t-1}{}^{n-(i_t-1-(j-1))} \\ & z^{'}_{i_t}{}^{n-(i_t-(j-1))} z^{'}_{i_t+1}{}^{n-(i_t+1-(j-1))} z_{i_t+2}^{n-(i_t+2-(j-1))} \cdots z_{j-3}^2 z_{j-2} \bmod p \\ =& g^{k_1k_2+\cdots+k_{j-1}k_j+k_jk_{j+1}+\cdots+k_{i_1-1}k^{'}_{i_1}+k^{'}_{i_1}k_{i_1+1}+\cdots+k_{i_t-1}k^{'}_{i_t}+k^{'}_{i_t}k_{i_t+1}+\cdots+k_nk_1} \\ & \bmod p.\end{aligned}$$

Because $K^{'}/\hat{K}_j = g^{k_{j-1}k_{j+1}-k_{j-1}k_j-k_jk_{j+1}} \bmod p$, U_j can obtain $K^{'}$ by $K^{'} = \hat{K}_j(z^{'}_{j-1}/z_{j-1})w_{j+1}^{-k_j} \bmod p$ according to Claim 1. Therefore, a malicious participant U_j can obtain the new conference-key even when t participants who are not pre-position and post-position participants of U_j re-randomize their secret short-term keys. □

Lemma 2. *Let U_j be a malicious participant. Even when t-consecutive participants ($t \leq n-3$) including pre-position and/or post-position participants of U_j re-randomize their secret short-term keys to establish a new conference-key $K^{'}$ for the forward confidentiality against U_j, U_j can obtain $K^{'}$.*

Proof. If U_{j-1} re-randomizes its secret key, U_j can compute $K^{'}/K$ as

$$\begin{aligned}K^{'}/K &= g^{k_{j-2}k^{'}_{j-1}+k^{'}_{j-1}k_{j+1}-k_{j-2}k_{j-1}-k_{j-1}k_j-k_jk_{j+1}} \bmod p \\ &= (z^{'}_{j-2}/z_{j-2})^2(z^{'}_{j-1}/z_{j-1})w_{j+1}^{-k_j} \bmod p.\end{aligned}$$

If U_{j-2} and U_{j-1} re-randomize their secret keys, U_j can compute $K^{'}/K$ as

$$\begin{aligned} K^{'}/K &= g^{k_{j-3}k^{'}_{j-2}+k^{'}_{j-2}k^{'}_{j-1}+k^{'}_{j-1}k_{j+1}-k_{j-3}k_{j-2}-k_{j-2}k_{j-1}-k_{j-1}k_j-k_jk_{j+1}} \bmod p \\ &= (z^{'}_{j-3}/z_{j-3})^3(z^{'}_{j-2}/z_{j-2})^2(z^{'}_{j-1}/z_{j-1})w_{j+1}^{-k_j} \bmod p. \end{aligned}$$

According to above relations, if $U_{j-t}, \ldots, U_{j-1}$ re-randomize their secret keys, U_j can obtain $K^{'}$ by

$$K^{'} = K(z^{'}_{j-t-1}/z_{j-t-1})^{t+1}(z^{'}_{j-t}/z_{j-t})^t \cdots (z^{'}_{j-1}/z_{j-1})w_{j+1}^{-k_j} \bmod p. \quad (5)$$

Similarly, if $U_{j+1}, \ldots, U_{j+t}$ re-randomize their secret keys, U_j can obtain $K^{'}$ by

$$\begin{aligned} K^{'} =& K(z^{'}_{j-1}/z_{j-1})^{t+1}(z^{'}_{j+1}/z_{j+1})^t \cdots (z^{'}_{j+t}/z_{j+t})(w_{j-1}^{k_j}w_{j+1}^{-k_j})^t w_{j+1}^{-k_j} \\ & \bmod p. \end{aligned} \quad (6)$$

According to Eqs. (5) and (6), if $U_l, \ldots, U_{l+t}$ re-randomize their secret keys ($l < j < l+t$), U_j can obtain $K^{'}$ by

$$\begin{aligned} K^{'} =& K(z^{'}_{l-1}/z_{l-1})^{t+1} \cdots (z^{'}_{j-1}/z_{j-1})^{t+1-(j-l)}(z^{'}_{j+1}/z_{j+1})^{t-(j-l)} \cdots \\ & (z^{'}_{l+t}/z_{l+t})(w_{j-1}^{k_j}w_{j+1}^{-k_j})^{t-(j-l)}w_{j+1}^{-k_j} \bmod p. \end{aligned} \quad (7)$$

Eqs. (5), (6), and (7) only hold when $t \leq n-3$. Therefore, a malicious participant U_j can obtain a new conference-key even when t-consecutive participants ($t \leq n-3$) including pre-position and/or post-position participants of U_j re-randomize their secret keys. □

Theorem 1. *At least $n-2$ participants should re-randomize their secret short-term keys to assure forward and backward confidentiality.*

Proof. To prove this theorem, we use Lemmas 1 and 2. At first, we apply an intermediate-key method of Lemma 1 to remove re-randomizations of participants who are not in t-consecutive participants including pre-position and/or post-position participants of a malicious participant U_j. Then, we apply Lemma 2 to remove re-randomizations of the t-consecutive participants. Finally, U_j can obtain a new conference-key. Moreover, according to Claims 1, 2, and 3, we can apply an attack method to compromise the forward confidentiality to compromise the backward confidentiality also. Therefore, at least $n-2$ participants should re-randomize their secret short-term keys to assure the forward and backward confidentiality. □

6. Conclusions

In this paper, we proposed an attack method that compromise forward and backward confidentiality of the Tseng's fault-tolerant conference-key

agreement protocol. The proposed attack only requires additional one exponentiation, two multiplications, and one division. Because the cost of the attack is very low, it can be easily implemented. We also introduced a simple countermeasure, a short-term key re-randomization method. However, as we prove, the short-term key re-randomization can be broken by extending the proposed attack method. Therefore, a more efficient countermeasure is needed.

References

Ateniese, G., Steiner, M., Tsudik, G., 2002. New multiparty authentication services and key agreement protocols. IEEE J. Sel. Areas Commun. 18 (4), 628–639.

Boyd, C., Nieto, J. M. G., 2003. Round-optimal contributory conference key agreement. In: Proc. Public-Key Cryptography. Vol. 2567 of LNCS. pp. 161–174.

Bresson, E., Chevassult, O., Pointcheval, D., 2002. Dynamic group diffie-hellman key exchange under standard assumptions. In: Proc. Advances in Cryptology - Eurocrypt'02. Vol. 2332 of LNCS. pp. 321–336.

Bresson, E., Chevassult, O., Pointcheval, D., Quisquater, J.-J., 2001. Provably authenticated group diffie-hellman key exchange. In: Proc. 8th Annual ACM Conf. Computer and Communications Security. pp. 255–264.

Burmester, M. V., Desmedt, Y., 1994. A secure efficient conference key distribution system. In: Proc. Advances in Cryptology - Eurocrypt'94. Vol. 950 of LNCS. pp. 275–286.

Chang, C.-C., Wu, T.-C., Chen, C. P., 1992. The design of a conference key distribution system. In: Proc. Advances in Cryptology - Auscrypt'92. Vol. 718 of LNCS. pp. 457–466.

Hirose, S., Ikeda, K., 1997. A conference distribution system for the start configuration based on the discrete logarithm problem. Inform. Process. Lett. 62 (4), 189–192.

Hwang, M.-S., Yang, W.-P., 1995. Conference key distribution schemes for secure digital mobile communications. IEEE J. Sel. Areas Commun 13 (2), 416–420.

Katz, J., Yung, M., 2003. Scalable protocols for authenticated group key exchange. In: Proc. Advances in Cryptology - Crypto'03. Vol. 2729 of LNCS. pp. 110–125.

Shi, T., Guo, Y., Ma, J., 2004. A fault-tolerant and secure multi-conference-key agreement protocol. In: Proc. Int. Conf. Communications, Circuits and Systems. Vol. 1. pp. 18–21.

Tseng, Y.-M., 2002. Cryptanalysis and improvement of key distribution system for vsat satellite communication. Informatica 13 (3), 369–376.

Tseng, Y.-M., 2005. An improved conference-key agreement protocol with forward secrecy. Informatica 16 (2), 275–284.

Tseng, Y.-M., 2007. A communication-efficient and fault-tolerant conference-key agreement protocol with forward secrecy. J. Syst. Softw. 80, 1091–1101.

Tzeng, W.-G., 2002. A secure fault-tolerant conference-key agreement protocol. IEEE Trans. Comput. 51 (4), 373–399.

Yi, X., 2004. Identity-based fault-tolerant conference key agreement. IEEE Trans. Dependable Secur. Comput. 1 (3), 170–178.