mathematics

MDPI

Article

Member Tampering Attack on Burmester-Desmedt Group Key Exchange Protocol and Its Countermeasure

Da-Zhi Sun [1,*] and Yangguang Tian [2]

[1] Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
[2] Department of Computer Science, University of Surrey, Guildford GU2 7XH, Surrey, UK
* Correspondence: sundazhi@tju.edu.cn; Tel.: +86-22-2740-1091

Abstract: With the rapid development of cloud computing and mobile networks, more and more application scenarios require a secret group key for secure communication. Group Key Exchange (GKE) protocol provides a secret group key for three or more members. Burmester and Desmedt presented an influential GKE protocol, which has a broadcast version and a cyclic version. In this paper, we investigate the security weaknesses of the Burmester-Desmedt protocol. We report that both the broadcast version and the cyclic version of the Burmester-Desmedt protocol suffer member tampering attacks if the two members that belong to both group A and group B are corrupted. That is, two corrupted members can add some unknowing members of group A to group B and trick the legal members of group B to believe that these unknowing members share the secret group key with them after a protocol run. Furthermore, to defeat the member tampering attack, we propose digital signature-based improvements on the broadcast version and the cyclic version of the Burmester-Desmedt protocol. We hope our research results will encourage the development of more robust and effective GKE protocols that stand rigorous security analysis.

Keywords: group key exchange; member tampering attack; scalability; security protocol; security analysis

MSC: 68M12

Citation: Sun, D.-Z.; Tian, Y. Member Tampering Attack on Burmester-Desmedt Group Key Exchange Protocol and Its Countermeasure. *Mathematics* **2022**, *10*, 3685. https://doi.org/10.3390/math10193685

Academic Editors: Liehuang Zhu, Meng Li and Zijian Zhang

Received: 1 September 2022
Accepted: 5 October 2022
Published: 8 October 2022

1. Introduction

Group Key Exchange (GKE) protocol makes an extension for the traditional two-member key establishment [1]. The function of the GKE protocol is to provide a shared secret group key for three or more members. With the rapid development of cloud computing and mobile networks, more and more application scenarios require group communication. Naturally, the security of group communication depends on the security of the group key [2,3]. We can summarize the general requirements of the GKE protocols as follows.

(1) Members in various groups generate different shared secret group keys (also named group session keys).
(2) The group session key is dynamic and can be updated as needed.
(3) The protocol messages exchanged among the members are open and via the public channel.
(4) Each member calculates his group session key independently.

To realize group key generation, an obvious method is to deploy a Trusted Third Party (TTP) recognized by all members in a group communication system. Moreover, each member shares a secret key with the TTP in advance. When the group session key needs to be established, the TTP randomly generates a group session key, determines the group of members and encrypts the group session key by using the shared secret key with each member. Upon receiving the ciphertext of the group session key, each member in the group can decrypt it using his shared key and obtain the group session key. The disadvantage of this method is that there must be a TTP to provide group key generation

services for all members in real-time. That is, the online TTP means heavy computation and communication overhead and may become the technical bottleneck.

Another optional method that does not require a TTP is to choose a member of the group as the group controller. The group controller negotiates a secret Diffie-Hellman key [4] with each member of the group. Then, the group controller randomly generates a group session key, uses each Diffie-Hellman key to encrypt the group session key, and sends it to each member of the group. After receiving the ciphertext of the group session key, the members can respectively decrypt the group session key using its own Diffie-Hellman key. The disadvantage of this method is that the large overhead of computation and communication is a burden on the group controller, and the members of the group need to designate a group controller before the group key generation services.

In this paper, we investigate the GKE protocols based on generalizing Diffie-Hellman two-member key exchange [4], because such protocols can potentially avoid the TTP or the group controller. This is a desirable feature under decentralized computing environments such as ad hoc networks and Blockchain.

1.1. Basic Notion of Group Key Exchange

Let k be a security parameter and $U_1, U_2, \ldots, U_n$, $n = \text{poly}(k)$ (a polynomial in the security parameter k), be members that are allowed to take part in a specified GKE protocol P to generate a group session key K. We can define the GKE protocol as follows.

Definition 1. *Protocol P is called a GKE if: when a group of any t members has honestly executed the protocol P as specified, then each member U_i in the group will finally compute his own session key K_i such that $K = K_i$, where $3 \leq t \leq n$ and $i \in \{1,2, \ldots, n\}$.*

Assume that any GKE protocol P runs on public networks in which all members in a group can broadcast or send messages (bit strings) to other members in the presence of a (polynomially bounded) passive adversary E. Passive adversary E can eavesdrop on the transmitting messages and keep a log of the messages during past protocol runs. However, the passive adversary E does not modify the transmitting messages of protocol runs. A formal security definition as an example is presented to address the above passive adversary of the GKE protocol P [5] as follows.

Definition 2. *The GKE protocol P guarantees privacy if it is computationally infeasible for the passive adversary E to compute the group session key K. The GKE protocol P guarantees secrecy if the passive adversary E cannot distinguish the group session key K from a random bit string of the same length with probability better than $1/2 + \varepsilon$, where ε is negligible (in the security parameter k).*

1.2. Related Work

Some of the literature in the GKE domain mainly revolves around the exposition of new construction of the GKE protocols. An early investigation of GKE protocols by Ingemarsson, Tang, and Wong [6] described a number of so-called Diffie-Hellman generalizations based on the idea of symmetric functions. Steer et al. [7] gave other GKE protocols generalizing the Diffie-Hellman two-member key exchange. Their protocol is suitable for adding new group members; however, deleting members is relatively difficult. Latter, Steiner, Tsudik, and Waidner [8] presented three GKE protocols (named GDH.1, GDH.2, and GDH.3) to improve the communication and computation costs of the Ingemarsson-Tang-Wong GKE protocols. Kim, Perrig, and Tsudik [9], and Bresson and Manulis [10] considered tree-based Diffie-Hellman two-member key exchange and showed that their protocols are very suitable when the composition of the group is changed without restarting the whole protocol. Burmester and Desmedt [5] presented an influential GKE protocol by extending in a natural way of the Diffie-Hellman two-member key exchange. Harn and Lin [11], Harn and Hsu [12] proposed GKE protocols based on both the secret sharing scheme and the group Diffie-Hellman key exchange scheme. Moriya, Takashima, and Takagi [13] and Fan,

Xu, and Li [14] proposed the GKE protocols based on Commutative Supersingular Isogeny Diffie-Hellman (CSIDH), which are the post-quantum Diffie-Hellman type key exchange protocols from commutative group action.

Some of the literature in the GKE domain research on the security analysis of the GKE protocols. Katz and Yung [15] presented a compiler that transforms the GKE protocol secure against a passive adversary and proved the Burmester-Desmedt protocol secure against the passive adversary. Emmanuel, Chevassut, and Pointcheval [16] provided a formal security model and several security definitions, which are adapted to many cryptographic scenarios for authenticated Diffie-Hellman key exchange in a group. To prevent stronger attacks against honest but open members, Bresson, Manulis, and Schwenk [17] gave examples where m uncorrupted members accept each other but have different session keys. Gorantla, Boyd, and Nieto [18] specifically analyzed the security of the GKE protocol under the key-compromise impersonation attacks. Baouch et al. [19] introduced an active attack on the Burmester-Desmedt protocol, that is, the adversary can obtain a copy of the shared key, which is created in a collaborative manner with the legal members in a group. Yang et al. [20] presented a security model for generic dynamic authenticated GKE to cover more active attacks (such as leakage of the ephemeral secret key) than previous similar models. As another line of security analysis work on GKE, Cohn-Gordon et al. [21] considered the post-compromise security of group messaging, that is, an adversary who compromises a single group member can indefinitely read and inject messages and built a formal security model on the ideas of multi-stage key exchange by Fischlin and Günther [22]. Alwen et al. [23] further analyzed the TreeKEM protocol [24], which is proposed by the Internet Engineering Task Force (IETF) working group and is the core of the secure group messaging protocols on message-layer security. Alwen et al. [25] also defined optimal security notions for GKE and considered two settings, i.e., the passive setting and the active setting. In addition, Hougaard and Miyaji [26] recently proposed a GKE compiler using any two-member key exchange for which the shared key space is the subset of a group and whose security reduces to the Decisional Diffie-Hellman (DDH) problem. Poettering et al. [27] classified characteristics of a large selection of game based GKE models, including those proposed for GKE with post-compromise security, and observed a range of shortcomings in some of the studied models.

1.3. Our Motivations and Contributions

When a GKE protocol is running, the adversary may illegally add some dummy members to the group. Although these dummy members do not belong to the group, the legal members believe that they are in the group and share a group session key with them. We call it the member tampering attack. To the best of our knowledge, the member tampering problem of the GKE protocol has not been studied yet. It is widely known that the Burmester-Desmedt protocol has advantages over other existing GKE protocols in terms of computation cost, communication cost, and the number of rounds. Moreover, since broadcasting messages can be expensive in some communications networks, the Burmester-Desmedt protocol [1,5] has not only a broadcast version but also a cyclic version. Therefore, we investigate the member tampering attack on the broadcast version and cyclic version of the Burmester-Desmedt protocol. We report that both the broadcast version and cyclic version suffer the member tampering attack when two members are corrupted. That is, two corrupted members, who belong to two groups A and B, can collude to include some unknowing members of group A into the group B and cheat other legal members of the group B to believe that these unknowing members exist in their group and share a group session key after the protocol run.

2. Review of the Burmester-Desmedt Group Key Exchange Protocol

Burmester and Desmedt [5] employed a certain structure of the cyclic group to design their GKE protocol. Without loss of generality, we can write it as follows. Let G be a cyclic group with prime order q and let g be a fixed generator of G. Let $Z_p = Z/pZ$ be the integers

modulo p. The cyclic group requires choosing two primes p and q such that $q \mid p-1$ and a generator $g \in Z_p$ with the order q. Then, the cyclic group is formed by $G = \{g^x \bmod p, x \in Z\}$.

Assume that $U_1, U_2, \ldots, U_n$ are all legal members of the GKE system. Initially, a GKE center chooses the system parameters for the GKE protocol. That is, the GKE center determines a security parameter k and a constant $c \geq 1$, randomly generates a prime $p = \Theta(2^{ck})$ and an element $g \in Z_p$ of order $q = \Theta(2^k)$, where Θ denotes asymptotic tight bound, and publishes the parameters p, g, and q to build the cyclic group G. Let $U_1, U_2, \ldots, U_{t \leq n}$ be a group of members who want to generate a group session key K. To enable a general description, we always allow any index i but U_i and U_j are the same members when satisfying $i = j \bmod t$.

2.1. Broadcast Version

The Burmester-Desmedt protocol is simplest to describe in the version that allows broadcast communications. Here, any message sent to more than one member means the broadcast.

Protocol 1: Broadcast version

Round 1. Each member U_i ($i = 1, 2, \ldots, t$) selects a random $r_i \in Z_q$ and computes and broadcasts $z_i = g^{r_i} \bmod p$.

Round 2. Each member U_i ($i = 1, 2, \ldots, t$) computes and broadcasts $X_i = (z_{i+1}/z_{i-1})^{r_i} \bmod p$.

Key Computation. Each member U_i ($i = 1, 2, \ldots, t$) computes his own session key:

$$K_i = (z_{i-1})^{tr_i} X_i^{t-1} X_{i+1}^{t-2} \ldots X_{i-2} \bmod p. \tag{1}$$

Figure 1 shows the communication and computation of broadcast version. If each member U_i ($i = 1, 2, \ldots, t$) honestly generates and broadcasts his z_i and X_i, all members will secretly share the same group session key $K = g^{r_1 r_t + r_1 r_2 + r_2 r_3 + \ldots + r_{t-2} r_{t-1} + r_{t-1} r_t} \bmod p$. That is to say, for every member U_i ($i = 1, 2, \ldots, t$), we have

$$\begin{gathered} K_i = (z_{i-1})^{tr_i} X_i^{t-1} X_{i+1}^{t-2} \ldots X_{i-2} \bmod p = \\ (z_{i-1})^{tr_i} \left(\frac{z_{i+1}}{z_{i-1}}\right)^{(t-1)r_i} \left(\frac{z_{i+2}}{z_i}\right)^{(t-2)r_{i+1}} \ldots \left(\frac{z_{i-1}}{z_{i-3}}\right)^{r_{i-2}} \bmod p = \\ (g^{r_{i-1}})^{tr_i} \left(\frac{g^{r_{i+1}}}{g^{r_{i-1}}}\right)^{(t-1)r_i} \left(\frac{g^{r_{i+2}}}{g^{r_i}}\right)^{(t-2)r_{i+1}} \ldots \left(\frac{g^{r_{i-1}}}{g^{r_{i-3}}}\right)^{r_{i-2}} \bmod p = \\ (g^{r_{i-1}})^{r_i} (g^{r_{i+1}})^{r_i} (g^{r_{i+2}})^{r_{i+1}} \ldots (g^{r_{i-3}})^{r_{i-2}} (g^{r_{i-1}})^{r_{i-2}} \bmod p = \\ g^{r_1 r_t + r_1 r_2 + r_2 r_3 + \ldots + r_{t-2} r_{t-1} + r_{t-1} r_t} \bmod p = K \end{gathered} \tag{2}$$

2.2. Cyclic Version

In the broadcast version, the protocol messages always need to be sent to all of the members. In some communications environments such as wired communications, the broadcast of messages to all involved members may be more costly than sending them to a single member. Therefore, the cyclic version can maintain the same Diffie-Hellman generalization function as the broadcast version but implement the point-to-point message transmission among the members.

Protocol 2: Cyclic version

Round 1. Each member U_i ($i = 1, 2, \ldots, t$) selects a random $r_i \in Z_q$, and then computes and sends $z_i = g^{r_i} \bmod p$ to the members U_{i-1} and U_{i-1}.

Round 2. Each member U_i ($i = 1, 2, \ldots, t$) computes $X_i = (z_{i+1}/z_{i-1})^{r_i} \bmod p$. Let $b_0 = c_0 = 1$. For $i = 1$ to t, the member U_i computes $b_i = b_{i-1} X_i \bmod p$ and $c_i = b_{i-1} c_{i-1} \bmod p$, and then sends b_i and c_i to the member U_{i+1}.

Round 3. Let $d_0 = c_t$ and $d = b_t$. For $i = 1$ to $t - 1$, the member U_i computes $d_i = d d_{i-1} X_i^{\ -t} \bmod p$, and then sends d_i and d to the member U_{i+1}.

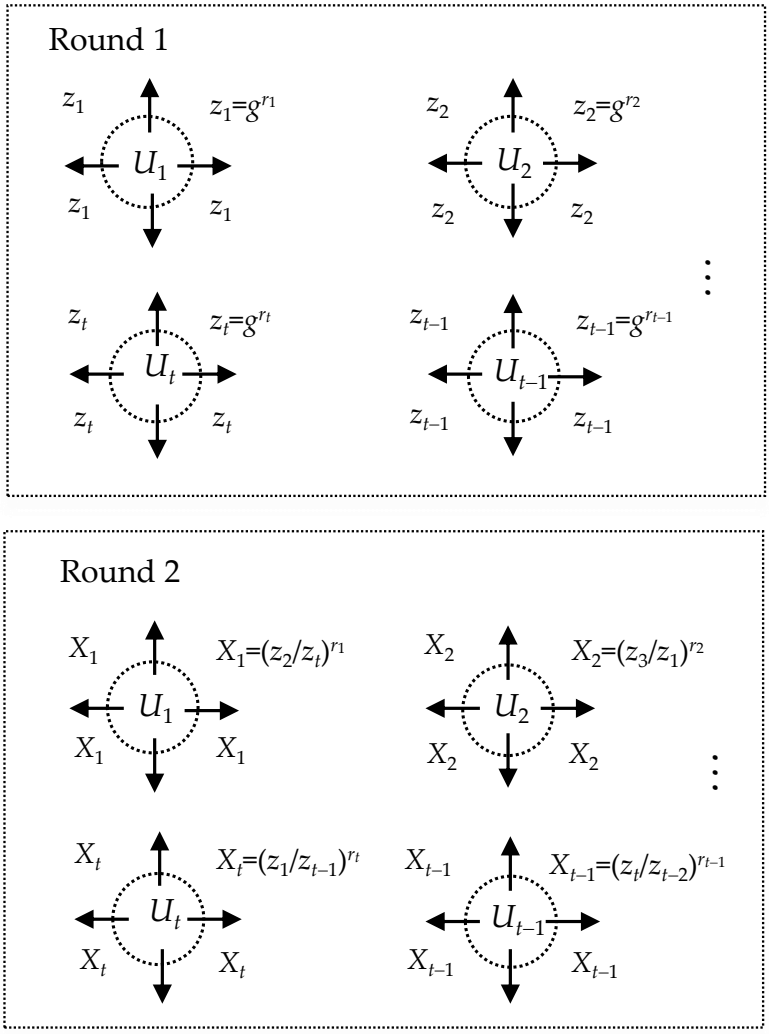


Figure 1. Communication and computation of broadcast protocol.

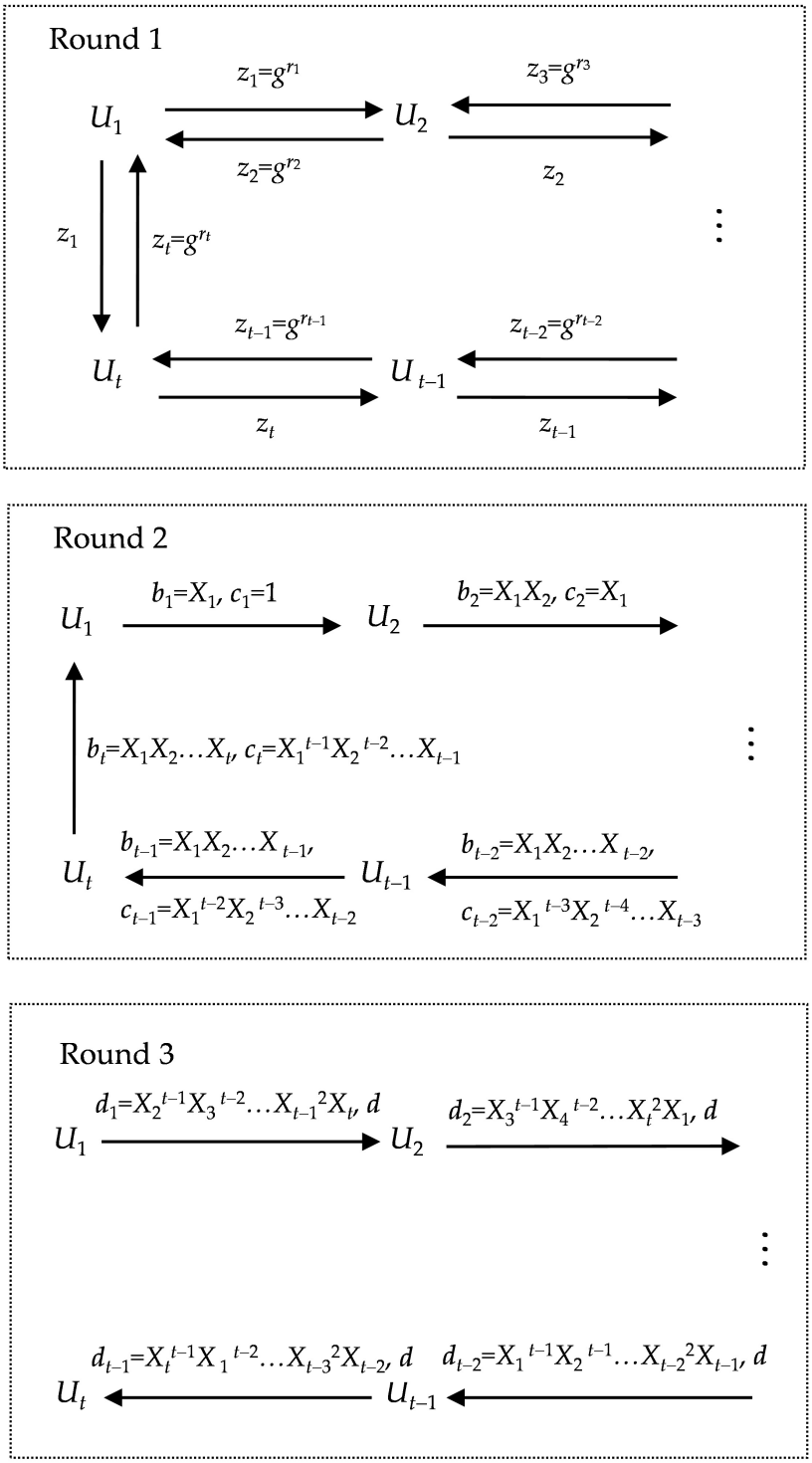
Key Computation. Each member U_i ($i = 1, 2, \ldots, t$) computes his own session key:

$$K_i = (z_{i-1})^{tr_i} d_{i-1} \bmod p. \tag{3}$$

It is easily to see that cyclic version does not require the broadcast channel. We know that in Round 2, $b_0 = c_0 = 1$, $b_1 = b_0X_1 = X_1 \bmod p$, $c_1 = b_0c_0 = 1$, $b_2 = b_1X_2 = X_1X_2 \bmod p$, $c_2 = b_1c_1 = X_1 \bmod p$, $b_3 = b_2X_3 = X_1X_2X_3 \bmod p$, $c_3 = b_2c_2 = {X_1}^2X_2 \bmod p$, $\ldots$, $b_t = b_{t-1}X_t = X_1X_2 \ldots X_t \bmod p$, $c_t = b_{t-1}c_{t-1} = X_1X_2 \ldots X_{t-2}X_{t-1}{X_1}^{t-2}{X_2}^{t-3} \ldots X_{t-2} = {X_1}^{t-1}{X_2}^{t-2} \ldots {X_{t-2}}^2X_{t-1} \bmod p$. Since $d_0 = c_t = {X_1}^{t-1}{X_2}^{t-2} \ldots X_{t-1} \bmod p$ and $d = b_t = X_1X_2 \ldots X_t \bmod p$ during Round 3, we get $d_1 = dd_0{X_1}^{-t} = X_1X_2X_3 \ldots X_{t-1}X_t{X_1}^{t-1}{X_2}^{t-2}{X_3}^{t-3} \ldots X_{t-1}{X_1}^{-t} = {X_2}^{t-1}{X_3}^{t-2} \ldots {X_{t-1}}^2X_t \bmod p$, $d_2 = dd_1{X_2}^{-t} = X_1X_2X_3X_4 \ldots X_t{X_2}^{t-1}{X_3}^{t-2}{X_4}^{t-3} \ldots X_t{X_2}^{-t} = {X_3}^{t-1}{X_4}^{t-2} \ldots {X_t}^2X_1 \bmod p$, $\ldots$, $d_{t-1} = dd_{t-2}{X_{t-1}}^{-t} = X_1X_2 \ldots X_{t-3}X_{t-2}X_{t-1}X_t{X_{t-1}}^{t-1}{X_t}^{t-2}{X_1}^{t-3}\ {X_2}^{t-4} \ldots X_{t-3}{X_{t-1}}^{-t} = {X_t}^{t-1}{X_1}^{t-2}{X_2}^{t-3} \ldots {X_{t-3}}^2X_{t-2} \bmod p$. We therefore know $d_{i-1} = {X_i}^{t-1}{X_{i+1}}^{t-2} \ldots X_{i-2} \bmod p$, where $i = 1, 2, \ldots, t$. Thus, for all $1 \le i \le t$, it follows that

$$\begin{gathered} K_i = (z_{i-1})^{tr_i} d_{i-1} \bmod p = (z_{i-1})^{tr_i} X_i^{t-1} X_{i+1}^{t-2} \ldots X_{i-2} \bmod p = \\ g^{r_1r_t + r_1r_2 + r_2r_3 + \ldots + r_{t-2}r_{t-1} + r_{t-1}r_t} \bmod p = K. \end{gathered} \tag{4}$$

Hence, we conclude that all members in cyclic version can finally share the same group session key as broadcast version. Figure 2 shows the communication and computation of cyclic version.



Shared session key: $K = K_i = (z_{i-1})^{tr_i} d_{i-1} \bmod p$, where $i = 1,2,\ldots,t$

Figure 2. Communication and computation of cyclic protocol.

2.3. Security Results

The security of the Burmester-Desmedt protocol depends on the Computational Diffie-Hellman (CDH) problem and the DDH problem. We describe the CDH problem and the DDH problem as follows.

Definition 3. *The CDH problem: given two primes p and g and two random numbers* $x = g^a \bmod p$, $y = g^b \bmod p$, *find* $g^{ab} \bmod p$.

Definition 4. *The DDH problem: given two primes p and g and three random numbers* $x = g^a \bmod p$, $y = g^b \bmod p$, *and* $z = g^c \bmod p$, *decide if* $ab = c$.

The Burmester-Desmedt protocol achieves the following security properties [5].

Fact 1. *Protocol 1 and Protocol 2 guarantee the privacy as in Definition 2 if, and only if, the CDH problem is intractable. Protocol 1 and Protocol 2 guarantee the secrecy as in Definition 2 if, and only if, the DDH problem is intractable.*

Fact 1 indicates that both Protocol 1 and Protocol 2 maintain a sound security promise when the CDH problem and the DDH problem are hard to solve.

3. Member Tampering Attacks on the Burmester-Desmedt Protocol

We know that any group $U_1, U_2, \ldots, U_t$ named as group A can execute a protocol run and negotiate a group session key K by using Equation (1) or Equation (3). We assume that members $U_{i'}$ and $U_{i''}$ are two malicious insiders in group A, that is, $i', i'' \in \{1, 2, \ldots, t\}$ and $i' \neq i''$. Without loss of generality, let $0 < i' < i'' < t+1$. Moreover, both members $U_{i'}$ and $U_{i''}$ simultaneously initiate an abnormal protocol run with another group $U_{a_1}, U_{a_2}, \ldots, U_{a_{t'}}$ named as the group B. Here, $0 < a_j < n + 1$ for $0 < j < t' + 1$.

3.1. Attacks on Broadcast Version

In the protocol run as shown in Figure 1, each member U_i ($i = 1, 2, \ldots, t$) of group A should compute and broadcast $z_i = g^{r_i} \bmod p$ and $X_i = (z_{i+1}/z_{i-1})^{r_i} \bmod p$ during Round 1 and Round 2 of Protocol 1. Finally, each member U_i ($i = 1, 2, \ldots, t$) is able to compute and share the same group session key $K = g^{r_1 r_t + r_1 r_2 + r_2 r_3 + \cdots + r_{t-2} r_{t-1} + r_{t-1} r_t} \bmod p$ by computing $(z_{i-1})^{t r_i} {X_i}^{t-1} {X_{i+1}}^{t-2} \cdots X_{i-2} \bmod p$. To realize the member tampering attack, two malicious insiders $U_{i'}$ and $U_{i''}$ of group A simultaneously start an abnormal protocol run with another group B. Let $i'' - i' > 1$. We assume that each member U_i ($i = i' + 1, i' + 2, \ldots, i'' - 1$) in the group A does not belong to the group B, that is, $a_j \notin \{i'+1, i'+2, \ldots, i'' - 1\}$ for all $j \in \{1, 2, \ldots, t'\}$. After the abnormal protocol run of the group B, each member U_{a_j} ($j = 1, 2, \ldots, t'$) believes that not only members U_{a_j} ($j = 1, 2, \ldots, t'$) but also members U_i ($i = i', i' + 1, \ldots, i''$) take part in negotiating a group session key. However, all members U_i ($i = i' + 1, i' + 2, \ldots, i'' - 1$) in group A actually do not involve themselves in the abnormal protocol run of group B. In the following, we always write $t' + i'' - i'$ as f to make the expression concise and easy to understand. We describe the abnormal protocol run as follows.

Round 1. Each member U_{a_j} ($j = 1, 2, \ldots, t'$) selects a random number $r'_j \in Z_q$, and computes and broadcasts $z'_j = g^{r'_j} \bmod p$. Each z_i generated by the member U_i ($i = i', i'+1, \ldots, i''$) also is broadcasted to group B.

Round 2. The malicious insider $U_{i''}$ computes and broadcasts $X'_{f+1} = (z'_1/z_{i''-1})^{r_{i''}} \bmod p$ and the member U_{a_1} computes and broadcasts $X'_1 = (z'_2/z_{i''})^{r'_1} \bmod p$. Each member U_{a_j} ($j = 2, \ldots, t' - 1$) computes and broadcasts $X'_j = (z'_{j+1}/z'_{j-1})^{r'_j} \bmod p$. The member $U_{a_{t'}}$ computes and broadcasts $X'_{t'} = (z_{i'}/z'_{t'-1})^{r_{t'}} \bmod p$ and the malicious insider $U_{i'}$ computes and broadcasts $X'_{t'+1} = (z_{i'-1}/z'_{t'})^{r_{i'}} \bmod p$. In addition, each X_i generated by the member U_i ($i = i' + 1, \ldots, i'' - 1$) is broadcasted to all members U_{a_j} ($j = 1, 2, \ldots, t'$).

Key Computation. Assume that $r'_i = r_{i+i'-t'-1}$ and $z'_i = z_{i+i'-t'-1}$, where $i = t'+1, t'+2, \ldots, t'+i''-i'+1$ and $X'_i = X_{i+i'-t'-1}$, where $i = t'+2, t'+3, \ldots, f$. Each member U_{a_j} ($j = 1, 2, \ldots, t'$) computes the group session key:

$$K'_j = \left(z'_{j-1}\right)^{(f+1)r'_j} X'^{f}_{j} X'^{f-1}_{j+1} \ldots X'_{j-2} \bmod p, \tag{5}$$

where the indexes j in both z'_j and X'_j are performed modulo $f+1$.

The member $U_{i'}$ can share the group session key by calculating

$$K'_{i'} = \left(z'_{t'}\right)^{(f+1)r_{i'}} X'^{f}_{t'+1} X'^{f-1}_{t'+2} \ldots X'_{t'-1} \bmod p. \tag{6}$$

The member $U_{i''}$ also can obtain the group session key by calculating

$$K'_{i''} = \left(z'_{f}\right)^{(f+1)r_{i''}} X'^{f}_{f+1} X'^{f-1}_{1} \ldots X'_{f-1} \bmod p. \tag{7}$$

According to Equation (5), we have

$$\begin{gathered} K'_j = \left(z'_{j-1}\right)^{(f+1)r'_j} X'^{f}_{j} X'^{f-1}_{j+1} \ldots X'_{j-2} \bmod p = \\ \left(z'_{j-1}\right)^{(f+1)r'_j} \left(\frac{z'_{j+1}}{z'_{j-1}}\right)^{fr'_j} \left(\frac{z'_{j+2}}{z'_{j}}\right)^{(f-1)r'_{j+1}} \ldots \left(\frac{z'_{j-1}}{z'_{j-3}}\right)^{r'_{j-2}} \bmod p, \end{gathered} \tag{8}$$

$$\begin{aligned} K'_j &= \left(g^{r'_{j-1}}\right)^{(f+1)r'_j} \left(\frac{g^{r'_{j+1}}}{g^{r'_{j-1}}}\right)^{fr'_j} \left(\frac{g^{r'_{j+2}}}{g^{r'_{j}}}\right)^{(f-1)r'_{j+1}} \ldots \left(\frac{g^{r'_{j-1}}}{g^{r'_{j-3}}}\right)^{r'_{j-2}} \bmod p \\ &= \left(g^{r'_{j-1}}\right)^{r'_j} \left(g^{r'_{j+1}}\right)^{r'_j} \left(g^{r'_{j+2}}\right)^{r'_{j+1}} \ldots \left(g^{r'_{j-3}}\right)^{r'_{j-2}} \left(g^{r'_{j-1}}\right)^{r'_{j-2}} \bmod p, \end{aligned} \tag{9}$$

$$\begin{gathered} K'_j = g^{r'_1 r'_{f+1} + r'_1 r'_2 + r'_2 r'_3 + \ldots + r'_{f-1} r'_f + r'_f r'_{f+1}} \bmod p = \\ g^{r'_1 r_{i''} + r'_1 r'_2 + r'_2 r'_3 + \ldots + r_{i''-2} r_{i''-1} + r_{i''-1} r_{i''}} \bmod p. \end{gathered} \tag{10}$$

Since $r_{i'} = r'_{t'+1}$ and $r_{i''} = r'_{f+1}$, we know that Equations (6) and (7) satisfy Equation (5) when $j = t'+1$ and $j = f+1$. Hence, all members $U_{i'}$, U_{a_j} ($j = 1, 2, \ldots, t'$), $U_{i''}$ should share the same group session key $K' = K'_{i'} = K'_{i''} = K'_j = g^{r'_1 r_{t''} + r'_1 r'_2 + r'_2 r'_3 + \ldots + r_{i''-2} r_{i''-1} + r_{i''-1} r_{i''}} \bmod p$. Moreover, all members U_{a_j} ($j = 1, 2, \ldots, t'$) believe that the members U_i ($i = i'+1, i'+2, \ldots, i''-1$) also exist in their group. Figure 3 shows member tampering attacks, which exploits two parallel protocol runs of group A and group B. In fact, the malicious insiders $U_{i'}$ and $U_{i''}$ also can use above attack to add the unknowing members U_i ($i = 1, 2, \ldots, i'-1, i''+1, i''+2, \ldots, t$) of the group A into the group B.

3.2. Attack on Cyclic Version

Consider the protocol run of group A. As shown in Figure 2, each member U_i ($i = 1, 2, \ldots, t$) computes $z_i = g^{r_i} \bmod p$ and sends it to the members U_{i-1} and U_{i+1} during Round 1 of Protocol 2. Each member U_i ($i = 1, 2, \ldots, t$) computes $X_i = (z_{i+1}/z_{i-1})^{r_i} \bmod p$, $b_i = b_{i-1} X_i \bmod p$, $c_i = b_{i-1} c_{i-1} \bmod p$, where $b_0 = c_0 = 1$, and then sends b_i and c_i to the member U_{i+1} during Round 2 of Protocol 2. In Round 3 of Protocol 2, each member U_i ($i = 1, 2, \ldots, t-1$) computes $d_i = d d_{i-1} X_i^{-t} \bmod p$, where $d = b_t$ and $d_0 = c_t$, and then sends d_i and d to the next member U_{i+1}. Finally, each member U_i ($i = 1, 2, \ldots, t$) computes and secretly shares the group session key $K = K_i = (z_{i-1})^{tr_i} d_{i-1} \bmod p$.

Meanwhile, the malicious insiders $U_{i'}$ and $U_{i''}$ of group A also simultaneously start another abnormal protocol run with the group $U_{a_1}, U_{a_2}, \ldots, U_{a_{t'}}$, i.e., group B. Here, we assume that $i''-i' < t-1$ and $a_j \notin \{1, 2, \ldots, i'-1, i''+1, i''+2, \ldots, t\}$ for any $j \in \{1, 2, \ldots, t'\}$. At the end of the abnormal protocol run of group B, each member U_{a_j} ($j = 1, 2, \ldots, t'$) mistakenly believes that the members U_i, where $i = 1, 2, \ldots, i'-1, i''+1, i''+2, \ldots, t$, also are in their group and share a group session key with them, although any of these members

do not participate in in the abnormal protocol run. We assume that the malicious insider $U_{i''}$ eavesdrops on all b_i, where $i = 1, 2, \ldots, i' - 2, i'' + 1, i'' + 2, \ldots, t$. Since the malicious insider $U_{i''}$ knows $b_0 = 1$, he can further compute all $X_i = b_i / b_{i-1} \bmod p$, where $i = 1, 2, \ldots, i' - 1, i'' + 1, i'' + 2, \ldots, t$. Figure 4 depicts the member tampering attack on the cyclic version. For simplicity, we write $t + i' + t' - i''$ as h in the following. We further describe the abnormal protocol run of group B as follows.

Round 1. The member $U_{i'}$ sends $z_{i'}$ to member U_{a_1}. The member U_{a_1} selects a random number $r'_1 \in Z_q$, computes $z'_1 = g^{r'_1} \bmod p$ and sends z'_1 to the members $U_{i'}$ and U_{a_2}. Each member U_{a_j} ($j = 2, 3, \ldots, t' - 1$) selects a random number $r'_j \in Z_q$, computes $z'_j = g^{r'_j} \bmod p$, and sends z'_j to the members U_{a_j-1} and U_{a_j+1}. The member $U_{a_{t'}}$ selects a random number $r'_{t'} \in Z_q$ and computes $z'_{t'} = g^{r'_{t'}} \bmod p$, and then sends $z'_{t'}$ to the members $U_{a_{t'}-1}$ and $U_{i''}$. The member $U_{i''}$ sends his $z_{i''}$ to the member $U_{a_{t'}}$.

Round 2. The member $U_{i'}$ computes and updates his new $X'_{h+1} = (z'_1 / z_{i'-1})^{r_{i'}} \bmod p$. The member $U_{i'}$ then computes $b' = b_{i'-1} X'_{h+1} \bmod p$ and $c' = c_{i'} = b_{i'-1} c_{i'-1} \bmod p$, and further sends b' and c' to the member U_{a_1}. The member U_{a_1} computes $X'_1 = (z'_2 / z_{i'})^{r'_1} \bmod p$. Moreover, member U_{a_1} computes $b'_1 = b' X'_1 \bmod p$ and $c'_1 = b' c' \bmod p$, and then sends b'_1 and c'_1 to the next member U_{a_2}. Each member U_{a_j} ($j = 2, 3, \ldots, t' - 1$) computes $X'_j = (z'_{j+1} / z'_{j-1})^{r'_j} \bmod p$. For $j = 2$ to $t' - 1$, the member U_{a_j} computes $b'_j = b'_{j-1} X'_j \bmod p$ and $c'_j = b'_{j-1} c'_{j-1} \bmod p$, and then sends b'_j and c'_j to U_{a_j+1}. Upon receiving the member $U_{a_{t'}-1}$'s $b'_{t'-1}$ and $c'_{t'-1}$, the member $U_{a_{t'}}$ computes $X'_{t'} = (z_{i''} / z'_{t'-1})^{r' t'} \bmod p$, $b'_{t'} = b'_{t'-1} X'_{t'} \bmod p$, $c'_{t'} = b'_{t'-1} c'_{t'-1} \bmod p$, and then sends $b'_{t'}$ and $c'_{t'}$ to the member $U_{i''}$. Upon receiving the member $U_{a_{t'}}$'s $b'_{t'}$ and $c'_{t'}$, the member $U_{i''}$ calculates his $X'_{t'+1} = (z_{i''-1} / z'_{t'})^{r_{i''}} \bmod p$, $b'_{t'+1} = b'_{t'} X'_{t'+1} \bmod p$, $c'_{t'+1} = b'_{t'} c'_{t'} \bmod p$. For $i = i'' + 1$ to t, the member $U_{i''}$ further computes $b'_{i+t'-i''+1} = b'_{i+t'-i''} X_i \bmod p$ and $c'_{i+t'-i''+1} = b'_{i+t'-i''} c'_{i+t'-i''} \bmod p$.

Round 3. Let $d'_0 = c'_{t+t'-i''+1}$. For $i = 1$ to $i' - 1$, the member $U_{i''}$ computes $d'_i = b'_{t+t'-i''+1} d'_{i-1} X_i^{-(h+1)} \bmod p$. And then, the member $U_{i''}$ sends $d'_{i'-1}$ and $b'_{t+t'-i''+1}$ to the member $U_{i'}$. The member $U_{i'}$ computes $d'_{i'} = b'_{t+t'-i''+1} d'_{i'-1} X'_{h+1}{}^{-(h+1)} \bmod p$ and sends $d'_{i'}$ and $b'_{t+t'-i''+1}$ to the member U_{a_1}. For $j = 1$ to $t' - 1$, the member U_{a_j} computes $d'_{j+i'} = b'_{t+t'-i''+1} d'_{j+i'-1} X'_j{}^{-(h+1)} \bmod p$, and then sends $d_{j+i'}$ and $b'_{t+t'-i''+1}$ to the member U_{a_j+1}. Upon receiving the member U_{a_j-1}'s $d'_{i'+t'-1}$ and $b'_{t+t'-i''+1}$, the member $U_{a_{t'}}$ computes $d'_{i'+t'} = b'_{t+t'-i''+1} d'_{i'+t'-1} X'_{t'}{}^{-(h+1)} \bmod p$, and then sends $d'_{i'+t}$ and $b'_{t+t'-i''+1}$ to the member $U_{i''}$.

Key Computation. Let $z'_0 = z_{i'}$. Each member U_{a_j} ($j = 1, 2, \ldots, t'$) computes the group session key:

$$K\prime_j = \left(z\prime_{j-1}\right)^{(h+1) r'_j} d\prime_{j+i'-1} \bmod p. \tag{11}$$

The member $U_{i'}$ shares the group session key by calculating

$$K'_{i'} = (z_{i'-1})^{(h+1) r_{i'}} d'_{i'-1} \bmod p. \tag{12}$$

The member $U_{i''}$ also obtains the group session key by computing

$$K'_{i''} = \left(z'_{t'}\right)^{(h+1) r_{i''}} d'_{i'+t'} \bmod p. \tag{13}$$

In the following, we analyze and confirm the group session key shared among the members $U_{i'}$, U_{a_j} ($j = 1, 2, \ldots, t'$), $U_{i''}$. During Round 1, the members $U_{i'}$ and $U_{i''}$ reuse their $z_{i'}$ and $z_{i''}$ generated in the protocol run of the group A and send $z_{i'}$ to the member U_{a_1} and $z_{i''}$ to the member $U_{a_{t'}}$. During Round 2, the member $U_{i'}$ computes his new $X'_{h+1} = (z'_1 / z_{i'-1})^{r_{i'}} \bmod p$ instead of the old $X_{i'} = (z_{i'+1} / z_{i'-1})^{r_{i'}} \bmod p$ in the protocol run of the group A, and uses this X'_{h+1} to compute $b' = b_{i'-1} X'_{h+1} \bmod p$. Then, the member U_{a_1} computes $X'_1 = (z'_2 / z_{i'})^{r'_1} \bmod p$ and $b'_1 = b' X'_1 \bmod p$ and the member U_{a_j} computes $X'_j = (z'_{j+1} / z'_{j-1})^{r'_j} \bmod p$ and $b'_j = b'_{j-1} X'_j \bmod p$ for $j = 2$ to t'. After calculating $X'_{t'+1} = (z_{i''-1} / z'_{t'})^{r_{i''}} \bmod p$ and $b'_{t'+1} = b'_{t'} X'_{t'+1} \bmod p$, the member $U_{i''}$ can sequentially compute each member U_i's $b'_{i+t'-i''+1} = b'_{i+t'-i''} X_i \bmod p$, where $i = i'' + 1, i'' + 2, \ldots, t$. The

reason is that the member $U_{i''}$ had eavesdropped on all b_i for $i = i'' + 1, i'' + 2, \ldots, t$ in the protocol run of the group A and can derive the member U_i's X_i by computing b_i/b_{i-1} mod p. Therefore, we know that

$$b' = b_{i'-1}X'_{h+1} = X_1X_2 \ldots X_{i'-1}X'_{h+1} \bmod p, \tag{14}$$

$$\begin{aligned} b\prime_{t'} = b\prime_{t'-1}X'_{t'} &= b\prime_{t'-2}X'_{t'-1}X'_{t'} = b\prime X\prime_1 \ldots X'_{t'-1}X\prime_{t'} \\ &= X_1X_2 \ldots X_{i'-1}X'_{h+1}X'_1 \ldots X'_{t'-1}X'_{t'} \bmod p, \end{aligned} \tag{15}$$

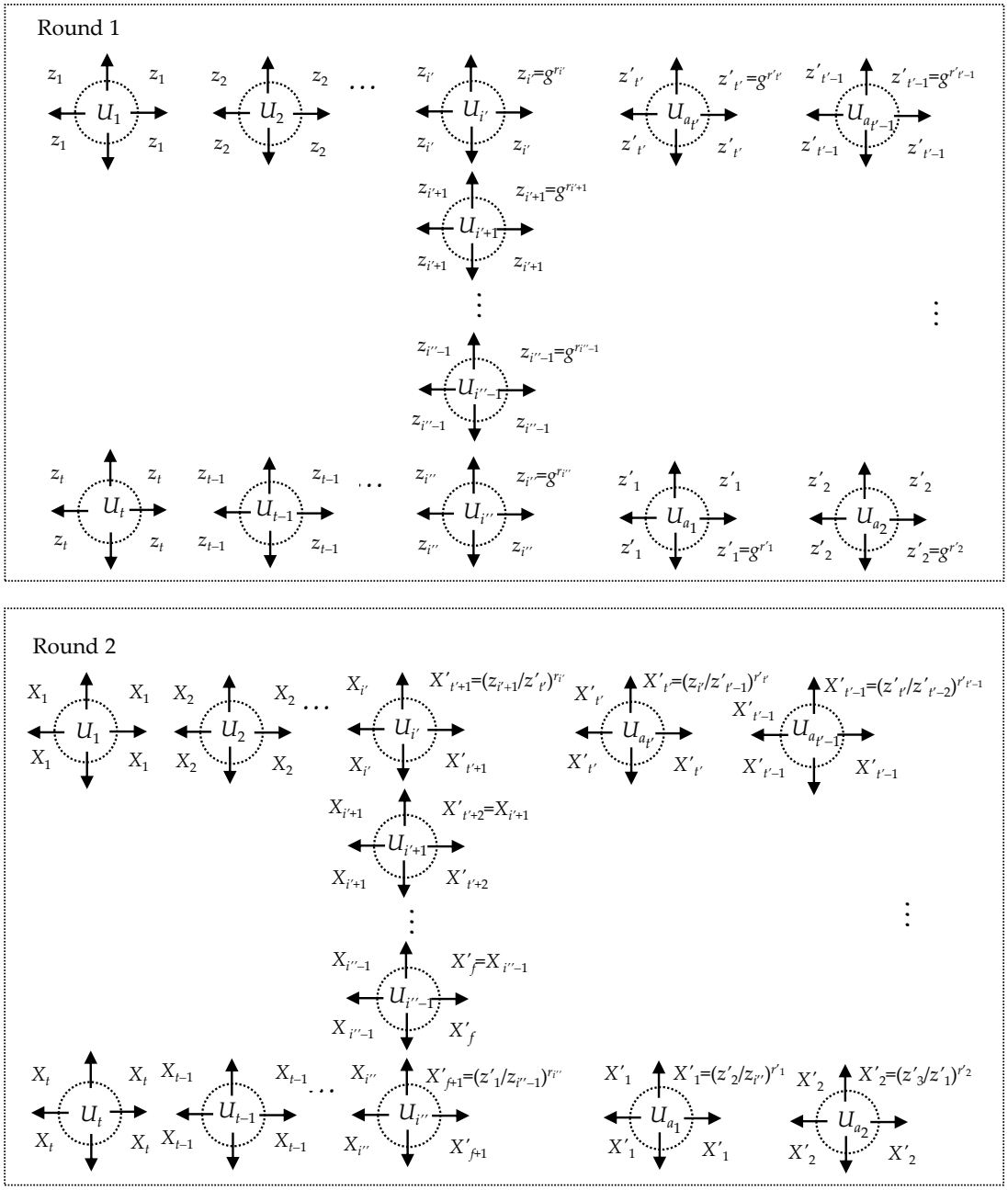


Figure 3. Member tampering attack on the broadcast protocol.

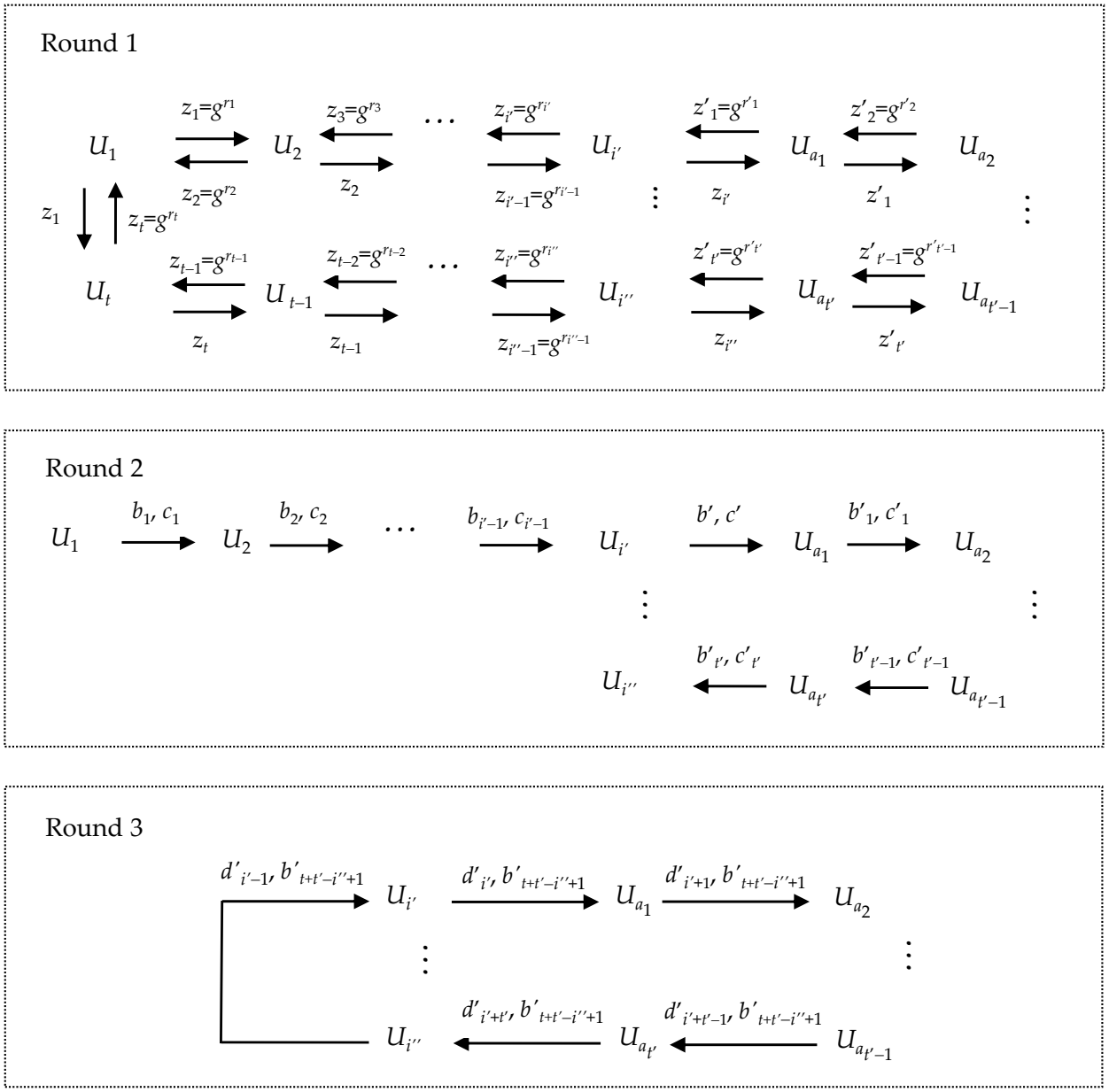


Figure 4. Member tampering attack on the cyclic protocol.

and

$$\begin{gathered} b'_{t+t'-i''+1} = b'_{t+t'-i''}X_t = b'_{t+t'-i''-1}X_{t-1} = \ldots = \\ b'_{t'+1}X_{i''+1}X_{i''+2}\ldots X_{t-1}X_t = b'_{t'}X'_{t'+1}X_{i''+1}X_{i''+2}\ldots X_{t-1}X_t = \\ X_1X_2\ldots X_{i'-1}X'_{h+1}X'_1\ldots X'_{t'-1}X'_{t'}X'_{t'+1}X_{i''+1}X_{i''+2}\ldots X_{t-1}X_t \bmod p. \end{gathered} \tag{16}$$

According to the analysis in Section 2.2, we have $c' = c_{i'} = b_{i'-1}c_{i'-1} = X_1X_2 \ldots X_{i'-2}X_{i'-1}{X_1}^{i'-2}{X_2}^{i'-3}\ldots X_{i'-2} = {X_1}^{i'-1}{X_2}^{i'-2}\ldots {X_{i'-2}}^2X_{i'-1} \bmod p$. Since $b' = b_{i'-1}X'_{h+1} \bmod p$ in Round 2 of the group B, ${c'}_1 = b'c' = b_{i'-1}X'_{h+1}c' = b_{i'-1}X'_{h+1}{X_1}^{i'-1}{X_2}^{i'-2}\ldots {X_{i'-2}}^2X_{i'-1} = X_1X_2\ldots X_{i'-2}X_{i'-1}X'_{h+1}{X_1}^{i'-1}{X_2}^{i'-2}\ldots {X_{i'-2}}^2X_{i'-1} = {X_1}^{i'}{X_2}^{i'-1}\ldots$

$X_{i'-2}{}^{3}X_{i'-1}{}^{2}X'_{h+1} \bmod p$. We further get $c'_j = b'_{j-1}c'_{j-1} = b'_{j-1}b'_{j-2}\,c'_{j-2} = \ldots = b'_{j-1}b'_{j-2} \ldots b'_1c'_1 \bmod p$ for $j = 2$ to t'. Therefore,

$$\begin{aligned} &c'_{t'} = b'_{t'-1}b'_{t'-2}\ldots b'_1c'_1 = X'_{t'-1}b'_{t'-2}b'_{t'-2}\ldots b'_1c'_1 = \ldots \\ &= X'_{t'-1}X'^{2}_{t'-2}\ldots X'^{t'-2}_{2}b'^{t'-1}_{1}c'_1 = X'_{t'-1}X'^{2}_{t'-2}\ldots X'^{t'-2}_{2}X'^{t'-1}_{1}b'^{t'-1}c'_1 \\ &= X'_{t'-1}X'^{2}_{t'-2}\ldots X'^{t'-2}_{2}X'^{t'-1}_{1}X_1^{t'-1}X_2^{t'-1}\ldots X_{i'-1}^{t'-1}X'^{t'-1}_{h+1}X_1^{i'}X_2^{i'-1}\ldots X_{i'-1}^{2}X'_{h+1} \bmod p, \end{aligned} \tag{17}$$

$$c'_{t'} = X'_{t'-1}X'^{2}_{t'-2}\ldots X'^{t'-2}_{2}X'^{t'-1}_{1}X'^{t'}_{h+1}X_{i'-1}^{t'+1}\ldots X_2^{t'+i'-2}X_1^{t'+i'-1} \bmod p. \tag{18}$$

Since the member $U_{i''}$ in Round 2 of the group B computes $b'_{t'+1} = b'_{t'}X'_{t'+1} \bmod p$ and $c'_{t'+1} = b'_{t'}c'_{t'} \bmod p$, and then calculates $b'_{i+t'-i''+1} = b'_{i+t'-i''}X_i \bmod p$ and $c'_{i+t'-i''+1} = b'_{i+t'-i''}c'_{i+t'-i''} \bmod p$ for $i = i'' + 1$ to t, we have

$$\begin{aligned} &c'_{t+t'-i''+1} = b'_{t+t'-i''}c'_{t+t'-i''} = b'_{t+t'-i''}b'_{t+t'-i''-1}c'_{t+t'-i''-1} = \ldots = \\ &b'_{t+t'-i''}b'_{t+t'-i''-1}\ldots b'_{t'+1}c'_{t'+1} = b'_{t+t'-i''}b'_{t+t'-i''-1}\ldots b'_{t'+1}b'_{t'}c'_{t'}, \end{aligned} \tag{19}$$

$$\begin{aligned} c'_{t+t'-i''+1} &= X_{t-1}X_{t-2}^{2}\ldots X_{i''+1}^{t-i''-1}X'^{t-i''}_{t'+1}b'^{t-i''+1}_{t'}c'_{t'} \\ &= X_{t-1}X_{t-2}^{2}\ldots X_{i''+1}^{t-i''-1}X'^{t-i''}_{t'+1}X_1^{t-i''+1}X_2^{t-i''+1}\ldots \\ &X_{i'-1}^{t-i''+1}X'^{t-i''+1}_{h+1}X'^{t-i''+1}_{1}X'^{t-i''+1}_{2}\ldots X'^{t-i''+1}_{t'-2}X'^{t-i''+1}_{t'-1}X'^{t-i''+1}_{t'}X'_{t'-1} \\ &X'^{2}_{t'-2}\ldots X'^{t'-2}_{2}X'^{t'-1}_{1}X'^{t'}_{h+1}X_{i'-1}^{t'+1}\ldots X_2^{t'+i'-2}X_1^{t'+i'-1} \bmod p, \end{aligned} \tag{20}$$

$$\begin{aligned} c'_{t+t'-i''+1} &= X_1^{h}X_2^{h-1}\ldots X_{i'-1}^{t+t'-i''+2}X'^{t+t'-i''+1}_{h+1}X'^{t+t'-i''}_{1}X'^{t+t'-i''-1}_{2} \\ &\ldots X'^{t-i''+3}_{t'-2}X'^{t-i''+2}_{t'-1}X'^{t-i''+1}_{t'}X'^{t-i''}_{t'+1}X_{i''+1}^{t-i''-1}\ldots X_{t-2}^{2}X_{t-1} \bmod p. \end{aligned} \tag{21}$$

In Round 3 of the group B, we know $d'_0 = c'_{t+t'-i''+1}$ and $d'_i = b'_{t+t'-i''+1}d'_{i-1}X_i{}^{-(h+1)} \bmod p$ or $i = 1$ to $i' - 1$. That is, $d'_{i'-1} = b'_{t+t'-i''+1}d'_{i'-2}X_{i'-1}{}^{-(h+1)} = b'_{t+t'-i''+1}{}^{2}d'_{i'-3}X_{i'-1}{}^{-(h+1)}X_{i'-2}{}^{-(h+1)} = \ldots = b'_{t+t'-i''+1}{}^{i'-1}X_{i'-1}{}^{-(h+1)}X_{i'-2}{}^{-(h+1)} \ldots X_1{}^{-(h+1)}d'_0 = b'_{t+t'-i''+1}{}^{i'-1}X_{i'-1}{}^{-(h+1)}X_{i'-2}{}^{-(h+1)} \ldots X_2{}^{-(h+1)}X_1{}^{-(h+1)}c'_{t+t'-i''+1} \bmod p$. Now we have

$$\begin{aligned} &d'_{i'-1} = b'^{i'-1}_{t+i'-i''+1}X_{i'-1}^{-(h+1)}X_{i'-2}^{-(h+1)}\ldots X_2^{-(h+1)}X_1^{-(h+1)}c'_{t+t'-i''-1} = \\ &(X_1X_2\ldots X_{i'-2}X_{i'-1}X'_{h+1}X'_1X'_2\ldots X'_{t'-2}X'_{t'-1}X'_{t'}X'_{t'+1}X_{i''+1}\ldots X_{t-2}X_{t-1}X_t)^{i'-1} \\ &X_{i'-1}^{-(h+1)}X_{i'-2}^{-(h+1)}\ldots X_2^{-(h+1)}X_1^{-(h+1)}X_1^{h}X_2^{h-1}\ldots X'^{t+t'-i''-1}_{2}\ldots X'^{t+t'-i''+3}_{t'-2}X'^{t-i''+2}_{t'-1} \\ &X'^{t-i''+1}_{t'}X'^{t-i''}_{t'+1}X_{i''+1}^{t-i''-1}\ldots X_{t-2}^{2}X_{t-1} \bmod p, \end{aligned} \tag{22}$$

$$\begin{aligned} d\prime_{i'-1} &= X'^{h}_{h+1}X'^{h-1}_{1}X'^{h-2}_{2}\ldots X'^{t+i'-i''+2}_{t'-2}X'^{t+i'-i''+1}_{t'-1}X'^{t+i'-i''}_{t'} \\ &X'^{t+i'-i''-1}_{t'+1}X_{i''+1}^{t+i'-i''-2}\ldots X_{t-2}^{i'+1}X_{t-1}^{i'}X_t^{i'-1}X_1^{i'-2}X_2^{i'-3}\ldots X_{i'-2} \bmod p. \end{aligned} \tag{23}$$

Owing to $d'_{i'} = b'_{t+t'-i''+1}d'_{i'-1}X'_{h+1}{}^{-(h+1)} \bmod p$, we get

$$\begin{aligned} &d\prime_{i'} = X_1X_2\ldots X_{i'-2}X_{i'-1}X'_{h+1}X'_1X'_2\ldots X'_{t'-2}X'_{t'-1}X'_{t'}X'_{t'+1}X_{i''+1}\ldots \\ &X_{t-2}X_{t-1}X_tX'^{h}_{h+1}X'^{h-1}_{1}X'^{h-2}_{2}\ldots X'^{t+i'-i''+2}_{t'-2}X'^{t+i'-i''+1}_{t'-1}X'^{t+i'-i''}_{t'} \\ &X'^{t+i'-i''-1}_{t'+1}X_{i''+1}^{t+i'-i''-2}\ldots X_{t-2}^{i'+1}X_{t-1}^{i'}X_t^{i'-1}X_1^{i'-2}X_2^{i'-3}\ldots X_{i'-2}X'^{-(h+1)}_{h+1} \bmod p, \end{aligned} \tag{24}$$

$$\begin{aligned} d\prime_{i'} &= X'^{h}_{1}X'^{h-1}_{2}\ldots X'^{t+i'-i''+3}_{t'-2}X'^{t+i'-i''+2}_{t'-1}X'^{t+i'-i''+1}_{t'}X'^{t+i'-i''}_{t'+1}X_{i''+1}^{t+i'-i''-1} \\ &\ldots X_{t-2}^{i'+2}X_{t-1}^{i'+1}X_t^{i'}X_1^{i'-1}X_2^{i'-2}\ldots X_{i'-2}^{2}X_{i'-1} \bmod p. \end{aligned} \tag{25}$$

Since $d'_{j+i'} = b'_{t+t'-i''+1}d'_{j+i'-1}X'_j{}^{-(h+1)} \bmod p$ for $j = 1$ to $t' - 1$ and $d'_{t'+i'} = b'_{t+t'-i''+1}d'_{t'+i'-1}X'_t{}^{-(h+1)} \bmod p$, we get

$$\begin{aligned} d'_{j+i'} &= b'^{2}_{t+i'-i''+1}d'_{j+i'-2}X'^{-(h+1)}_{j} = \ldots = b'^{j}_{t+i'-i''+1}d'_{i'}X'^{-(h+1)}_{j} \\ &X'^{-(h+1)}_{j-1}X'^{-(h+1)}_{j-2}\ldots X'^{-(h+1)}_{1} \bmod p, \end{aligned} \tag{26}$$

$$
\begin{aligned}
&d'_{j+i'}\\
&=\left(X_1X_2\ldots X_{i'-1}X'_{h+1}X'_1\ldots X'_{j-2}X'_{j-1}X'_jX'_{j+1}X'_{j+2}\ldots X'_{t'}X'_{t'+1}X_{i''+1}\ldots X_{t-2}X_{t-1}X_t\right)^j\\
&X'^h_1\ldots X'^{h-j+3}_{j-2}X'^{h-j+2}_{j-1}X'^{h-j+1}_{j}X'^{h-j}_{j+1}X'^{h-j-1}_{j+2}\ldots X'^{t+i'-i''+1}_{t'}X'^{t+i'-i''}_{t'+1}X^{t+i'-i''-1}_{i''+1}\ldots\\
&X^{i'+2}_{t-2}X^{i'+1}_{t-1}X^{i'}_tX^{i'-1}_1X^{i'-2}_2\ldots X_{i'-1}X'^{-(h+1)}_{j}X'^{-(h+1)}_{j-1}X'^{-(h+1)}_{j-2}\ldots X'^{-(h+1)}_1\bmod p,
\end{aligned}
\tag{27}
$$

$$
\begin{aligned}
d'_{j+i'}&=X'^h_{j+1}X'^{h-1}_{j+2}\ldots X'^{j+t+i'-i''+1}_{t'}X'^{j+t+i'-i''}_{t'+1}X^{j+t+i'-i''-1}_{i''+1}\ldots\\
&X^{j+i'+2}_{t-2}X^{j+i'+1}_{t-1}X^{j+i'}_tX^{j+i'-1}_1X^{j+i'-2}_2\ldots X^{j+1}_{i'-1}X'^j_{h+1}X'^{j-1}_1\ldots X'^2_{j-2}X'_{j-1}\bmod p
\end{aligned}
\tag{28}
$$

and

$$
\begin{aligned}
&d'_{i'+t'}=b'_{t+i'-i''+1}d'_{i'+t'-1}X'^{-(h+1)}_{t'}=X_1X_2\cdots X_{i'-1}X'_{h+1}X'_1\cdots X'_{t'-3}\\
&X'_{t'-2}X'_{t'-1}X'_{t'}X'_{t'+1}X_{i''+1}\cdots X_{t-2}X_{t-1}X_tX'^h_{t'}X'^{h-1}_{t'+1}X^{h-2}_{i''+1}\cdots X^{i'+t'+1}_{t-2}X^{i'+t'}_{t-1}\\
&X^{i'+t'-1}_tX^{i'+t'-2}_1X^{i'+t'-3}_2\cdots X^{t'}_{i'-1}X'^{t'-1}_{h+1}X'^{t'-2}_1\cdots X'^2_{t'-3}X'_{t'-2}X'^{-(h+1)}_{t'}\bmod p,
\end{aligned}
\tag{29}
$$

$$
\begin{aligned}
&d'_{i'+t'}=X'^h_{t'+1}X^{h-1}_{i''+1}\cdots X^{i'+t'+2}_{t-2}X^{i'+t'+1}_{t-1}X^{i'+t'}_tX^{i'+t'-1}_1X^{i'+t'-2}_2\ldots\\
&X'^h_{t'+1}X^{h-1}_{i''+1}\cdots X^{i'+t'+2}_{t-2}X^{i'+t'+1}_{t-1}X^{i'+t'}_tX^{i'+t'-1}_1X^{i'+t'-2}_2\ldots\\
&X^{t'+1}_{i'-1}X'^{t'}_{h+1}X'^{t'-1}_1\cdots X'^3_{t'-3}X'^2_{t'-2}X'_{t'-1}\bmod p.
\end{aligned}
\tag{30}
$$

Let $X'_{l+t+t'-i''+1}=X_l \bmod p$ for $l=1$ to $i'-1$ and $X'_{l+t'-i''+1}=X_l \bmod p$ for $l=i''+1$ to t. Moreover, let $n'=h+1$ and any index j but X'_i and X'_j are the same value when satisfying $i=j \bmod n'$. According to Equations (11), (25), and (28), we have

$$
\begin{aligned}
K'_j&=\left(z'_{j-1}\right)^{(h+1)r'_j}d'_{j+i'-1}\\
&=\left(z'_{j-1}\right)^{(h+1)r'_j}X'^h_{j+1}X'^{h-1}_{j+2}\ldots X'^{j+t+i'-i''+1}_{t'}X'^{j+t+i'-i''}_{t'+1}\\
&X'^{j+t+i'-i''-1}_{t'+2}\ldots X'^{j+i'+2}_{t+t'-i''-1}X'^{j+i'+1}_{t+t'-i''}X'^{j+i'}_{t+t'-i''+1}X'^{j+i'-1}_{t+t'-i''+2}X'^{j+i'-2}_{t+t'-i''+3}\cdots\\
&X'^{j+1}_hX'^j_{h+1}X'^{j-1}_1\ldots X'^2_{j-2}X'_{j-1}\bmod p
\end{aligned}
\tag{31}
$$

for $j=1,2,\ldots,t'$. We write $z'_h=z_{i'-1}$, $r'_{h+1}=r_{i'}$, $r'_{t'+1}=r_{i''}$. By Equations (23) and (30), and Equations (12) and (13) become

$$
\begin{aligned}
&K'_{i'}=(z_{i'-1})^{(h+1)r_{i'}}d'_{i'-1}=(z'_h)^{(h+1)r'_{h+1}}X'^h_{h+1}X'^{h-1}_1X'^{h-2}_2\ldots\\
&X'^{t+i'-i''+2}_{t'-2}X'^{t+i'-i''+1}_{t'-1}X'^{t+i'-i''}_{t'}X'^{t+i'-i''-1}_{t'+1}X'^{t+i'-i''-2}_{t'+2}\ldots X^{t'+1}_{t+t'-i''-1}\\
&X^{t'}_{t+t'-i''}X^{t'-1}_{t+t'-i''+1}X^{t'-2}_{t+t'-i''+2}X^{t'-3}_{t+t'-i''+3}\ldots X'_{h-1}\bmod p
\end{aligned}
\tag{32}
$$

and

$$
\begin{aligned}
&K'_{i''}=(z'_{t'})^{(h+1)r_{i''}}d'_{i'+t'}=(z'_{t'})^{(h+1)r'_{t'+1}}X'^h_{t'+1}X'^{h-1}_{t'+2}\cdots X'^{i'+t'+2}_{t+t'-i''-1}X'^{i'+t'+1}_{t+t'-i''}\\
&X'^{i'+t'}_{t+t'-i''+1}X'^{i'+t'-1}_{t+t'-i''+2}X'^{i'+t'-2}_{t+t'-i''+3}\cdots X'^{t'+1}_hX'^{t'}_{h+1}X'^{t'-1}_1\cdots X'^3_{t'-3}X'^2_{t'-2}X'_{t'-1}\bmod p.
\end{aligned}
\tag{33}
$$

According to Equations (31)–(33), we know Equations (11)–(13) are strictly consistent with Equation (1). This means that h+1 members, i.e., the members U_{a_j} ($j=1,2,\ldots,t'$) and U_i ($i=1,2,\ldots,i',i'',i''+1,\ldots,t$), run cyclic version of the protocol. Finally, we can concluded that each member U_{a_j} ($j=1,2,\ldots,t'$) believes the members U_i ($i=1,2,\ldots,i',i'',i''+1,\ldots,t$) of group A are in their group and share the secret key, i.e., $K'=g^{r_1r_2+r_2r_3+\ldots+r_{i'}r'_1+r'_1r'_2+\ldots+r'_{t'}r_{i''}+r_{i''}r_{i''+1}+\ldots+r_tr_1}\bmod p$. However, the fact is that the members U_i ($i=1,2,\ldots,i'-1,i''+1,i''+2,\ldots,t$) do not take part in the abnormal protocol run with the group B. It needs to point out that the member $U_{i''}$ can select the part of the members U_i ($i=i''+1,i''+2,\ldots,t$) to add in the group B or ignore them all. Assume that $S\subset\{i''+1,i''+2,\ldots,t\}$ and the member $U_{i''}$ wants to include the members $U_{i\in S}$ into group B. The member $U_{i''}$ only modifies the computation of the $b'_{i+t'-i''+1}$ and $c'_{i+t'-i''+1}$ in

Round 2. That is, for $i = i'' + 1$ to t, the member $U_{i''}$ computes $b'_{i+t'-i''+1} = b'_{i+t'-i''} X_i \bmod p$ and $c'_{i+t'-i''+1} = b'_{i+t'-i''} c'_{i+t'-i''} \bmod p$, when $i \in S$.

Further comment. The member tampering attacks are possibly exploited to cheat the members of the group and threaten group communication security. One threat scenario is patient consultation in telemedicine. When the medical experts and the patient establish a group communication by using the Burmester-Desmedt protocol, two medical experts can add some experts who have not participated in the consultation to the group as needed. This behavior can improve the authority of the consultation. Another threat scenario is the online auction system. If the online auction system runs the Burmester-Desmedt protocol, two members in the auction group can include the auction notaries, who do not exist in the auction group.

4. Countermeasure and Open Problem

The reason why the Burmester-Desmedt protocol suffers is that the members have not explicitly verified the identity of other members in the group. Therefore, the malicious insiders in the group can reuse messages of another protocol run to cheat other members into believing the non-existent members. If each member can confirm the identity of other members in the group during the protocol run, it is difficult for malicious insiders to add those dummy members.

Let $\mathrm{Sign}_{sk_{Ui}}()$ and $\mathrm{Vrfy}_{pk_{Ui}}()$ ($i = 1, 2, \ldots, n$) denote the member U_i's signature function using private key sk_{Ui} and verification function using public key pk_{Ui}. We require the signature scheme that is existentially unforgeable under an adaptive chosen-message attack. Here, the existentially unforgeable property and the game of adaptive chosen-message attack are the standard notions of security for digital signature schemes. Let UI_i ($i = 1, 2, \ldots, n$) be the member U_i's recognizable identities (in lexicographic order) wishing to establish the group session key. Let $\|$ be the concatenation operator. In the following, we provide the improvements on the Burmester-Desmedt protocol to defeat the member tampering attack.

In broadcast version as Figure 1, we require each member U_i ($i = 1, 2, \ldots, t$) to receiving all z_j and X_j during Round 1 and Round 2, where $j = 1, 2, \ldots, t$ and $j \neq i$. More importantly, each member U_i simultaneously computes and broadcasts $S_i = \mathrm{Sign}_{sk_{Ui}}(z_1 \| X_1 \| UI_1 \| z_2 \| X_2 \| UI_2 \| \ldots \| z_t \| X_t \| UI_t)$ in Round 2. Moreover, upon receiving all messages from other members, each member U_i also should verify all S_j by computing $\mathrm{Vrfy}_{pk_{Uj}}(S_j)$, where $j = 1, 2, \ldots, t$ and $j \neq i$. If any verification operation fails, the member U_i should terminate the protocol run. When an adversary wants to illegally include any member U_j into a certain group and cheat other members of the group, he must forge the member U_j's signature S_j during the protocol run. This signature signs recognizable identities of all group members and the fresh random values generated by all group members. However, the security of the signature algorithm [28] prevents the possibility of forging the member U_j's signature without his private key sk_{Ui}.

In cyclic version as Figure 2, we assume that each member U_i ($i = 1, 2, \ldots, t$) knows all other members U_j, where $j = 1, 2, \ldots, t$ and $j \neq i$, potentially are in the group before the protocol run starts. Figure 5 shows our improvement of cyclic version. Our Round 1 and Round 2 are fully same as cyclic version described in Figure 2. During Round 3, we require each member U_i ($i = 1, 2, \ldots, t$) simultaneously computes the signature $SC_i = \mathrm{Sign}_{sk_{Ui}}(K_i \| UI_1 \| UI_2 \| \ldots \| UI_t)$, and then each member U_i ($i = 1, 2, \ldots, t-1$) sends the signatures $SC_1, SC_2, \ldots, SC_i$ to the next member U_{i+1} and the member U_t sends the signatures SC_2, $SC_3, \ldots, SC_t$ to the member U_1. After Round 3, we additionally demand a new round named Round 4. In Round 4, each member U_i ($i = 1, 2, \ldots, t-2$) sends the signatures SC_{i+2}, $SC_{i+3}, \ldots, SC_t$ to the member U_{i+1}. In Round 3 and Round 4, each member U_i ($i = 1, 2, \ldots, t$) should verify all signatures SC_j by computing $\mathrm{Vrfy}_{pk_{Uj}}(SC_j)$, where $j = 1, 2, \ldots, t$ and $j \neq i$. If any verification operation fails, the member U_i should immediately terminate the protocol run. For reasons similar to the improved broadcast version, the

improved cyclic version also can resist the member tampering attack under the malicious active insider or outsider, when the signature algorithm is secure.

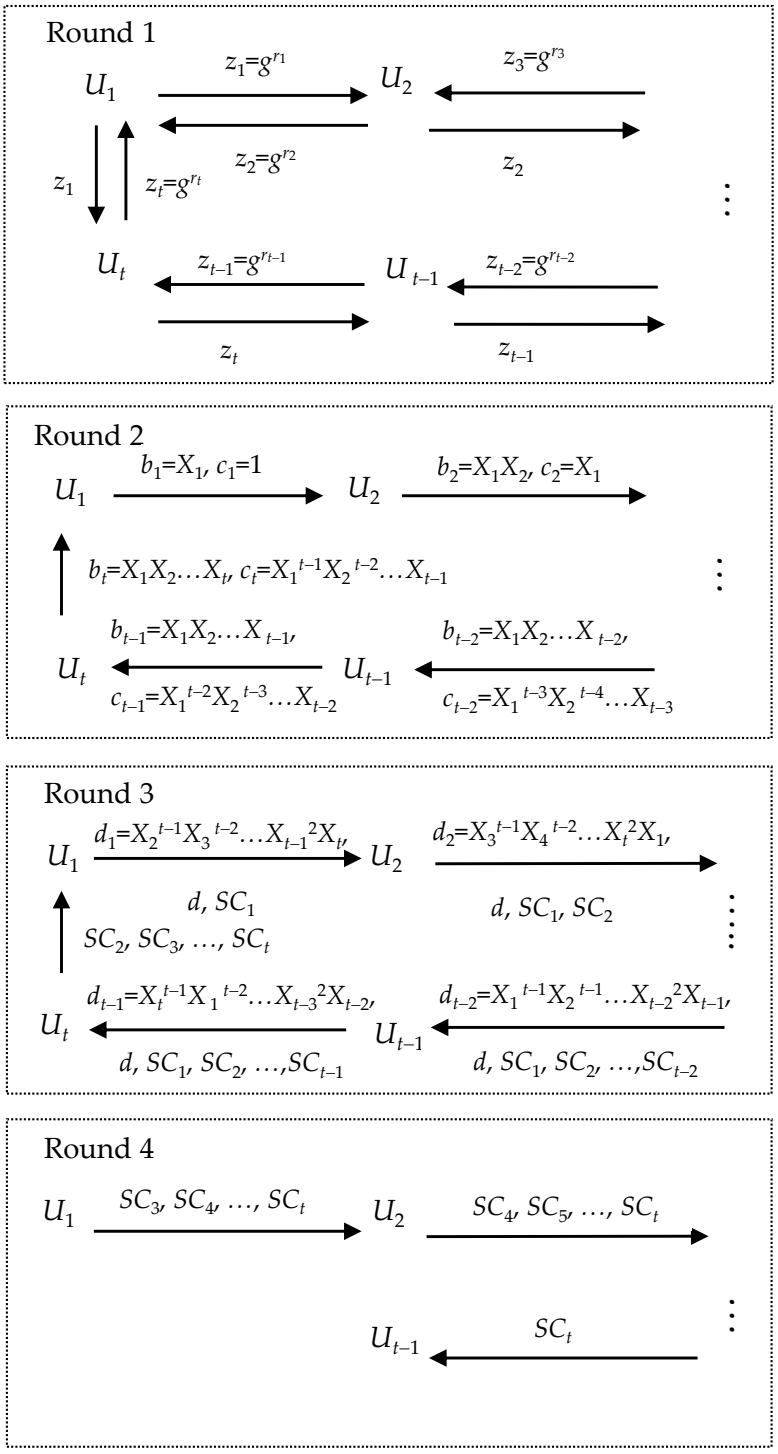


Figure 5. Improvement on cyclic protocol.

However, we argue that the above signature-based improvements on the Burmester-Desmedt protocol have at least two disadvantages, though they can resist member tampering attacks. First, the signature mechanism means that the group system needs to be equipped with Public Key Infrastructure (PKI) and issue digital certificates to ensure the authenticity of the members' public keys. In addition, the certificate chain may also increase the complexity of the group system. Second, the signature mechanism requires the

members of the group to generate, verify, transmit, and receive a considerable number of signatures. For example, each member in the improved broadcast version needs to generate and broadcast 1 signature and receive and verify the $t-1$ signature, if there are t members in the group. Moreover, the members of the group must run an extra round in an improved cyclic version. Obviously, these two disadvantages indicate a significant increase in the implementation overheads of the group system, when the Burmester-Desmedt protocol adopts the signature mechanism. Hence, the signature-based improvements on the Burmester-Desmedt protocol are not suitable for resource-constrained security environments, such as Bluetooth security [29,30]. The group systems based on Bluetooth low-energy devices are very popular in wireless personal area networks and Internet-of-Things (IoT). However, to maintain the low energy feature, Bluetooth security currently does not support the signature mechanism. Hence, an interesting open problem is whether one could enhance the Burmester-Desmedt protocol to prevent member tampering attacks without relying on resource-intensive public key algorithms, e.g., the signature and the public encryption. A related approach might be to rely on the cryptographic hash techniques to develop a more intricate hybrid structure for the Burmester-Desmedt protocol. This requires in-depth studies, including protocol design, efficiency evaluation, security analysis, etc. We leave it as future work.

5. Conclusions

One advantage of the Burmester-Desmedt protocol is scalability, that is, any $t > 2$ members can establish a group session key after the protocol run. The scalability is very fit to ad hoc networks, where the number of members is often changing. However, we have shown that two malicious insiders in both group A and the group B can exploit its scalability to add the unknowing members of group A into the group B and make other members of the group B mistakenly believe that their group session key is shared with those unknowing members. Hence, owing to our member tampering attacks, the number of members is a security parameter and needs to be protected in the appropriate way when the scalable GKE protocol is deployed.

We proposed the signature-based improvement on the Burmester-Desmedt protocols, which is secure against our member tampering attacks. We compared our improvement with Katz-Yung protocol [15], because they are all base on the Burmester-Desmedt protocol and employ the signature schemes. In the broadcast version, the Katz-Yung protocol requires signing the messages of Round 1 and Round 2, respectively. Therefore, each member in the Katz-Yung protocol demands double signature operations compared with our improvement on the broadcast version. To the best of our knowledge, no other signature-based improvement on the cyclic version is proposed. However, under resource-constrained environments, our signature-based improvement is still not practical in view of the implementation costs.

Author Contributions: Conceptualization, D.-Z.S.; methodology, D.-Z.S.; validation, D.-Z.S. and Y.T.; formal analysis, D.-Z.S.; investigation, D.-Z.S.; writing—original draft preparation, D.-Z.S. and Y.T.; writing—review and editing, D.-Z.S. and Y.T.; supervision, D.-Z.S.; funding acquisition, D.-Z.S. All authors have read and agreed to the published version of the manuscript.

Funding: The work of Da-Zhi Sun was supported in part by the National Natural Science Foundation of China under Grant No. 61872264. The APC was funded by the National Natural Science Foundation of China under Grant No. 61872264.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The authors would like to thank the editor and the reviewers for their valuable suggestions and comments.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Boyd, C.; Mathuria, A.; Stebila, D. *Protocols for Authentication and Key Establishment*, 2nd ed.; Springer: Berlin, Germany, 2020; pp. 389–440.
2. Ramakrishna, O.; Churi, P. Systematic Survey on Cryptographic Methods Used for Key Management in Cloud Computing. In Proceedings of the International Conference on Innovative Computing and Communications (ICICC 2021), Delhi, India, 20–21 February 2021; Khanna, A., Gupta, D., Bhattacharyya, S., Hassanien, A.E., Anand, S., Jaiswal, A., Eds.; Advances in Intelligent Systems and Computing (AISC); Springer: Singapore, 2022; Volume 1388, pp. 445–460.
3. Juric, R.; Lyth, A.; Larson, D. Group Key Management in Wireless Sensor Networks: Introducing Context for Managing the Re-Keying Process. In Proceedings of the 55th Hawaii International Conference on System Sciences (HICSS–55), Virtual Event, 15 June 2022; pp. 7434–7443. Available online: https://scholarspace.manoa.hawaii.edu/items/3f78810f-0fd2-41b7-bbe8-c2bacd9420de (accessed on 29 August 2022).
4. Diffie, W.; Hellman, M.E. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *IT-22*, 644–654. [CrossRef]
5. Burmester, M.; Desmedt, Y. A secure and scalable group key exchange system. *Inf. Process. Lett.* **2005**, *94*, 137–143. [CrossRef]
6. Ingemarsson, I.; Tang, D.T.; Wong, C.K. A conference key distribution system. *IEEE Trans. Info. Theory* **1982**, *IT-28*, 714–720. [CrossRef]
7. Steer, D.G.; Strawczynski, L.; Diffie, W.; Wiener, M. A Secure Audio Teleconference System. In Proceedings of the Conference on the Theory and Application of Cryptography: Advances in Cryptology (CRYPTO'88), Santa Barbara, CA, USA, 21–25 August 1988; Goldwasser, S., Ed.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1990; Volume 403, pp. 520–528.
8. Steiner, M.; Tsudik, G.; Waidner, M. Diffie-Hellman Key Distribution Extended to Group Communication. In Proceedings of the 3rd ACM Conference on Computer and Communications Security (CCS'96), New Delhi, India, 14–15 March 1996; Association for Computing Machinery: New York, NY, USA, 1996; pp. 31–37.
9. Kim, Y.; Perrig, A.; Tsudik, G. Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups. In Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS 2000), Athens, Greece, 1–4 November 2000; Samarati, P., Ed.; Association for Computing Machinery: New York, NY, USA, 2000; pp. 235–244.
10. Bresson, E.; Manulis, M. Securing Group Key Exchange against Strong Corruptions. In Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security (ASIACCS 2008), Tokyo, Japan, 18–20 March 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 249–260.
11. Harn, L.; Lin, C.L. Efficient group Diffie–Hellman key agreement protocols. *Comput. Electr. Eng.* **2014**, *40*, 1972–1980. [CrossRef]
12. Harn, L.; Hsu, C.F. A practical hybrid group key establishment for secure group communications. *Comput. J.* **2017**, *60*, 1582–1589. [CrossRef]
13. Moriya, T.; Takashima, K.; Takagi, T. Group Key Exchange from CSIDH and Its Application to Trusted Setup in Supersingular Isogeny Cryptosystems. In Proceedings of the International Conference on Information Security and Cryptology (Inscrypt 2019), Nanjing, China, 6–8 December 2019; Liu, Z., Yung, M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; Volume 12020, pp. 86–98.
14. Fan, X.J.; Xu, X.; Li, B. Group Key Exchange Protocols from Supersingular Isogenies. In Proceedings of the International Conference on Information Security and Cryptology (Inscrypt 2020), Guangzhou, China, 11–14 December 2020; Wu, Y.D., Yung, M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12612, pp. 157–173.
15. Katz, J.; Yung, M. Scalable protocols for authenticated group key exchange. *J. Cryptol.* **2007**, *20*, 85–113. [CrossRef]
16. Bresson, E.; Chevassut, O.; Pointcheval, D. Provably secure authenticated group Diffie-Hellman key exchange. *ACM Trans. Inf. Syst. Secur.* **2007**, *10*, 10. [CrossRef]
17. Bresson, E.; Manulis, M.; Schwenk, J. On Security Models and Compilers for Group Key Exchange Protocols. In Proceedings of the Advances in Information and Computer Security, Second International Workshop on Security (IWSEC 2007), Nara, Japan, 29–31 October 2007; Miyaji, A., Kikuchi, H., Rannenberg, K., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4752, pp. 292–307.
18. Gorantla, M.C.; Boyd, C.; Nieto, J.M.G. Modeling Key Compromise Impersonation Attacks on Group Key Exchange Protocols. In Proceedings of the International Workshop on Public Key Cryptography (PKC 2009), Irvine, CA, USA, 18–20 March 2009; Jarecki, S., Tsudik, G., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5443, pp. 105–123.
19. Baouch, M.; López-Ramos, J.A.; Schnyder, R.; Torrecillas, B. An Active Attack on a Distributed Group Key Exchange System. *arXiv* **2016**, arXiv:1603.09090. Available online: https://arxiv.org/abs/1603.09090 (accessed on 29 August 2022). [CrossRef]
20. Yang, Z.; Khan, M.; Liu, W.P.; He, J. On Security Analysis of Generic Dynamic Authenticated Group Key Exchange. In Proceedings of the Secure IT Systems–23rd Nordic Conference (NordSec 2018), Oslo, Norway, 28–30 November 2018; Gruschka, N., Ed.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 11252, pp. 121–137.
21. Cohn-Gordon, K.; Cremers, C.; Garratt, L.; Millican, J.; Milner, K. On Ends-to-Ends Encryption: Asynchronous Group Messaging with Strong Security Guarantees. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS 2018), Toronto, ON, Canada, 15–19 October 2018; Backes, M., Wang, X.F., Eds.; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1802–1819.

22. Fischlin, M.; Günther, F. Multi-Stage Key Exchange and the Case of Google's QUIC Protocol. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS 2014), Scottsdale, AZ, USA, 3–7 November 2014; Yung, M., Li, N.H., Eds.; Association for Computing Machinery: New York, NY, USA, 2014; pp. 1193–1204.
23. Alwen, J.; Coretti, S.; Dodis, Y.; Tselekounis, Y. Security Analysis and Improvements for the IETF MLS Standard for Group Messaging. In Proceedings of the 40th Annual International Cryptology Conference: Advances in Cryptology (CRYPTO 2020), Part I, Santa Barbara, CA, USA, 17–21 August 2020; Micciancio, D., Ristenpart, T., Eds.; Lecture Notes in Computer Science. Springer: Cham, Switzerland, 2020; Volume 12170, pp. 248–277.
24. Barnes, R.; Beurdouche, B.; Raphael, R.; Millican, J.; Omara, E.; Cohn-Gordon, K. The Messaging Layer Security (MLS) Protocol. Technical Report. 2020. Available online: https://datatracker.ietf.org/doc/draft-ietf-mls-protocol/ (accessed on 29 August 2022).
25. Alwen, J.; Coretti, S.; Jost, D.; Mularczyk, M. Continuous Group Key Agreement with Active Security. In Proceedings of the 18th Theory of Cryptography Conference (TCC 2020), Part II, Durham, NC, USA, 16–19 November 2020; Pass, R., Pietrzak, K., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12551, pp. 261–290.
26. Hougaard, H.B.; Miyaji, A. Group Key Exchange Compilers from Generic Key Exchanges. In Proceedings of the 15th International Conference on Network and System Security (NSS 2021), Tianjin, China, 23 October 2021; Yang, M., Chen, C., Liu, Y., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 13041, pp. 162–184.
27. Poettering, B.; Rösler, P.; Schwenk, J.; Stebila, D. SoK: Game-Based Security Models for Group Key Exchange. In Proceedings of the Cryptographers' Track at the RSA Conference 2021: Topics in Cryptology (CT–RSA 2021), Virtual Event, 17–20 May 2021; Paterson, K.G., Ed.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2021; Volume 12704, pp. 148–176.
28. Katz, J.; Lindell, Y. *Introduction to Modern Cryptography: Principles and Protocols*, 3rd ed.; Chapman & Hall/CRC: Boca Raton, FL, USA, 2020; pp. 463–495.
29. Sun, D.Z.; Sun, L. On secure simple pairing in Bluetooth standard v5.0-part i: Authenticated link key security and its home automation and entertainment applications. *Sensors* **2019**, *19*, 1158. [CrossRef] [PubMed]
30. Sun, D.Z.; Sun, L.; Yang, Y. On secure simple pairing in Bluetooth standard v5.0-part ii: Privacy analysis and enhancement for low energy. *Sensors* **2019**, *19*, 3259. [CrossRef]