Distributed Collaborative Key Agreement Protocol for Dynamic Peer Groups

Vibha
Indira Gandhi Institute of Technology
GGSIPU, Kashmere Gate
Delhi

Neha Bhatnagar
Indira Gandhi Institute of Technology
GGSIPU, Kashmere Gate
Delhi

Sakshi Gupta
Indira Gandhi Institute of Technology
GGSIPU, Kashmere Gate
Delhi

ABSTRACT

In recent years, group-oriented applications and collaborative protocols are gaining popularity. This calls for group key management which forms the basic building block in achieving secure group communication. The points that differentiate it from the traditional communication are that they do not have a centralized server but a distributed system, group key being contributory and dynamic nature wherein members can join and leave the group when required. The group key management requires for function such as join, leave along with the concept of rekeying. The group key is recomputed every time a member adds or leaves the group. The work in this paper provides a fundamental understanding about establishing a group key via a distributed and collaborative approach for a dynamic peer group.[1].

Keywords
Distributed, Collaborative Systems, Tree Based Diffie Hellman Protocol (TGDH), Rekeying

1. INTRODUCTION

As a result of the increased popularity of group-oriented applications and protocols, group communication occurs in many different settings: from video-conferencing to transferring important documents in a group[2]. This calls for group management in a secure manner. Centralized key management methods (key distribution) are appropriate for two-party (e.g., client-server or peer-to-peer) communication as well as for large multicast groups. However, many collaborative group settings (e.g., conferencing, white-boards, shared instruments, and command-and-control systems) require distributed key management techniques.[3]

The group is continuously with the members joining and leaving the group.

To illustrate the importance of distributive and collaborative system, consider a group of people in a peer-to-peer network in a closed and confidential meeting. The communication between them is susceptible to eavesdropping since they lack a common previously agreed upon communication key. To solve the problem, we need a secure distributed group key agreement and authentication protocol so that people can establish and authenticate a common group key for secure and private communication.

Note that this type of key agreement protocols is both distributed and contributory in nature: each member of the group contributes its part to the overall group key.

Another motivation for such kind of systems is that it provides stronger security. Additionally, it adapts to the heterogeneous environments and also provides for efficiency in communication and computation.

It is noted that the type of distributed group key agreement is very different from the traditional centralized protocols. The traditional protocols rely on a centralized key server which efficiently distributes the group key. This requires for group members to be arranged in a logical hierarchy called as key tree. Using a tree topology helps in distributing the keys very efficiently whenever there is a change in the group membership i.e., a new member join or leaving the group. In this system we are supposing that there is no centralized key server since in many situations a central system might not be available, for example adhoc systems. The advantage of a distributed system over centralized is that there is no central point failure and the key is contributory in nature.

This paper basically concentrates on the implementation of the distributive and collaborative network in a dynamic peer group. This explores the algorithm that is used and the rekeying concept used to achieve the same.

2. RELATED WORK

We studied a plethora of papers related to group communication and the various protocols that have been used to implement secure group communication. This was basically done to understand the pros and cons of the various methods used in achieving group communication in dynamic peer groups.

A. Overview of the existing approaches

There have been numerous approaches that have been described by researchers in this field. These basically differ in the protocol that has been used in achieving the group...
communication. The initial approach that dates back ten years down the line is focused on a centralized approach[8][9]. In [10], in order to achieve packet delivery from one or more authorized senders to a very large number of authorized receivers, a novel solution was suggested by using the concept of key graphs [5] to specify secure groups. But this concept had the disadvantage that it had centralization. In [14], Iolus, a framework for scalable secure multicasting was introduced but there was high latency since it involved decrypting and re-encrypting each packet from subgroup to subgroup, all the way to the destination members. Thus the centralized method suffers from the fact that the servers need great computation power, large communication bandwidth, and considerable storage space. These servers tend to be the performance bottleneck of the entire system. The distributed schemes are often based on the two party Diffie-Hellman key exchange (2DH) or its extensions[4]. In [13], the concept of key graphs was introduced. This involved generation of a one way function tree which made the group communication dynamic, large, any to any, was based on the concept of key distribution and was centralized.

This had the disadvantage of single point of failure and large bandwidth for distribution of keys. It proved to be more expensive for joining of new members than leaving of members. Thus there was a shift from centralized to distributed group key management solution. In [10], Steer et al. proposed a key agreement protocol STR for secure audio conferencing. The solution is not suitable for large groups because its computation and message size grow linearly with the group size. Cliques [12] group key also forms the primitives for group key management. But they have not been useful in large scalable groups. Therefore there is a need for an efficient protocol and therefore many tree based protocols have come up.

B. Motivation for TGDH

Our main focus is dynamic peer groups, which are characterized by relatively small in size (<100 of members), there is no hierarchy, frequent membership changes and any member can become a sender or a receiver. For such groups, the communication can be made using TGDH efficiently. Tree-based Group Diffie-Hellman (TGDH) contributory key agreement protocol is robust and efficient in the sense that it can deal with network partition and that the number of rounds for rekeying is limited by log N where N is the number of members currently in the group. [7]

Current tree-based group Diffie-Hellman (TGDH) schemes provide key independence within each group. TGDH protocols have achieved performance levels of O(log n) exponentiations and O(1) messages sent for entry into and exit from dynamic groups containing n members[11]. It provides key independence and perfect forward secrecy; it was also proven secure with respect to passive outside (eavesdropping) adversaries[6].

3. DIMENSIONS OF KEY AGREEMENT

The algorithm that is used is based on the contributory key arrangement. This means that a group key K is generated with a randomly taken input chosen by the constituent members. execution from the beginning. First the comparison between the traditional key distribution and the key agreement

Table 1: Comparison Of Key Distribution and Key Agreement.

A. Initial Key Agreement

Initial Key Agreement takes place at the time of group genesis. Naturally, it requires contacting every prospective group member. Contributory key agreement also calls for a key share to be obtained from each member. This continues until the mobile agent returns to its home machine after completing execution on the last machine in its itinerary [2].

B. Group Key Agreement

A comprehensive group key agreement solution must handle adjustments to group secrets subsequent to all membership change operations in the underlying group communication system.

The following membership changes are considered: We distinguish among single and multiple member operations. We also distinguish between additive and subtractive member operations. Single member changes include member join or leave, and multiple member changes include group merge and group partition.

- **Join** occurs when a prospective member wants to join a group
- **Leave** occurs when a member wants to leave (or is forced to leave) a group. There might be different reasons for member deletion such as voluntary leave, involuntary disconnect or forced expulsion. We believe that group key agreement must only provide the tools to adjust the group secrets and leave the rest up to the local security policy. [3]

4. TREE-BASED GROUP DIFFIE–HELLMAN PROTOCOL

To obtain the implementation of the dynamic peer group it is required that we have an algorithm which allows for mutations. The tree-based group Diffie Hellman algorithm is an extension of the basic Diffie Hellman algorithm wherein a key tree is constructed.

![Graph showing the arrangement of members in group according to TGDH](image)
A key tree is formed. Each node \( v \) represents a secret (private) key \( K_v \) and a blinded (public) key \( BK_v \).

\[
BK_v = \alpha^{K_v} \mod p, \quad \text{where } \alpha \text{ and } p \text{ are public parameters.}
\]

Every member holds the secret keys along the key path, and all the blinded keys in the key tree.

\( K_0 \) is the group key.\[1\]

\[
\begin{align*}
K_v &= (BK_{2v+1})^{K_{2v+2}} \mod p = (\alpha^{K_{2v+1}})^{K_{2v+2}} \mod p \\
K_v &= (BK_{2v+2})^{K_{2v+1}} \mod p = (\alpha^{K_{2v+2}})^{K_{2v+1}} \mod p \\
K_v &= \alpha^{K_{2v+1}K_{2v+2}} \mod p
\end{align*}
\]

The above mentioned key tree and computation help in handling the membership events of join and leave.

Rekeying (renewing the keys of the nodes) is performed for every single join/leave event to ensure backward and forward confidentiality.\[1\]

A special member called sponsor is elected to be responsible for broadcasting updated blinded keys.

**A. Single Member Leave Case**

\( M_6 \) becomes the sponsor. It rekeys the secret keys \( K_5 \) and \( K_0 \) and broadcasts the blinded key \( BK_2 \).

\( M_1, M_2, \) and \( M_3 \) compute \( K_0 \) given \( BK_2 \).

\( M_6 \) and \( M_7 \) compute \( K_2 \) and then \( K_0 \) given \( BK_{12} \).\[1\]

**B. Single Member Join Case**

\( M_8 \) becomes the sponsor again. It rekeys \( K_5, K_2 \) and \( K_0 \) given \( BK_{12} \) and broadcasts the blinded key \( BK_5 \) and \( BK_2 \).

Now everyone can compute the new group key. \[1\]

## 5. FUNCTIONING OF MEMBER LEAVE AND JOIN

Each group member contributes an equal share to the group key; this share is kept secret by each group member.\[3\]

- The group key is computed as a function of all current group members’ shares.\[3\]
- As the group shrinks, departing members’ shares are removed from the new group key and at least one remaining member changes its share.\[3\]
- In a join or a merge or a leave, since the recomputation takes place therefore the forward and backward secrecy is achieved.

## 6. DISCUSSION

### A. Computation Complexity

<table>
<thead>
<tr>
<th>The group key updating operation</th>
<th>GDH.2</th>
<th>TGDH</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>protocol</strong></td>
<td>( O(N^2) )</td>
<td>( O(\log N) )</td>
</tr>
<tr>
<td><strong>computation</strong></td>
<td>( N - 1 )</td>
<td>1</td>
</tr>
<tr>
<td><strong>unicasts</strong></td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>broadcasts</strong></td>
<td>1</td>
<td>( O(N) )</td>
</tr>
<tr>
<td><strong>total bandwidth</strong></td>
<td>( O(N^2) )</td>
<td>( O(N) )</td>
</tr>
</tbody>
</table>

Table 1: Comparison Of Complexity in Group Key Updating Operation\[15\]
The member joining operation

<table>
<thead>
<tr>
<th>protocol</th>
<th>GDH.2</th>
<th>TGDH</th>
</tr>
</thead>
<tbody>
<tr>
<td>computation</td>
<td>$O(N)$</td>
<td>$O(\log N)$</td>
</tr>
<tr>
<td>unicasts</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>broadcasts</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>total bandwidth</td>
<td>$O(N)$</td>
<td>$O(N)$</td>
</tr>
</tbody>
</table>

Table 2: Comparison Of Complexity in Member Joining Operation[15]

The member leaving operation

<table>
<thead>
<tr>
<th>protocol</th>
<th>GDH.2</th>
<th>TGDH</th>
</tr>
</thead>
<tbody>
<tr>
<td>computation</td>
<td>$O(N)$</td>
<td>$O(\log N)$</td>
</tr>
<tr>
<td>unicasts</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>broadcasts</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>total bandwidth</td>
<td>$O(N)$</td>
<td>$O(N)$</td>
</tr>
</tbody>
</table>

Table 3: Comparison Of Complexity in Member Leaving Operation[15]

GDH- Group Diffie Helman [2]

7. CONCLUSION

The distributed collaborative key arrangement for the dynamic peer group provides for forward and backward confidentiality and secrecy. We show that one can use the Tree Based Diffie Hellman protocol to achieve such distributive and collaborative key agreement. Each member has a group key associated to it and whenever a member adds or leaves a group, rekeying takes place. A group key allows for maintaining dynamic peer groups with secrecy and dynamicity.

8. FUTURE WORK

We have tried to achieve a method of secure group communication. We have designed our system which is based on the key agreement concept and the key is developed in a collaborative manner by the share of the keys from each of the members in a group. We have implemented secure group communication wherein members can join and leave. It uses the rekeying approach for joining or leaving events to maintain backward and forward confidentiality. In future, we aim to achieve the implementation of the distributive collaborative key agreement protocol in real life dynamic peer group wherein we have various groups that can join and leave. This would form the real implementation where factors such as network partitioning, etc also play a vital role.

9. REFERENCES


[5] Yongdae Kim Adrian Perrig, Gene Tsudik ,”Tree-Based Group Key Agreement”.


