

# Secure Group Communication

by  
**Ajit Burad**

*under the guidance of*  
**Prof. G. Siva Kumar**



Department of Computer Science and Engineering  
Indian Institute of Technology

# Overview

- Introduction
- Secure group communication framework
  - Admission and Access Control
  - Key Management
  - Role(level) updating
- Implementation of Secure MultiChat
  - Functional Components
  - JXTA Messages
- Conclusion & Future Scope

# Introduction

## What is peer to peer ?

A large group of users with common interests communicate with one another and each having equal responsibility.  
As opposed to client-server model

## Why peer to peer ?

- Decentralized architecture
- Flexible and dynamic behavior

**Possible Scenarios.:** Multiplayer gaming, file sharing.

# Requirement for Secure Group Communication

- Authentication, Admission Control
  - Admission process: Collaborative decision by group members.
- Member level Updation
  - Feedback from other members of group.
- Hierarchical access control
- Key Management
  - Ensuring forward and backward secrecy.
  - Generating and distributing group keys.

# Admission Control

Standardizing the admission procedure for a new peer.  
Without secure admission control, secure communication (e.g., key management) is useless.

## How can it be achieved ?

- Admission via ACLs
  - Group members known a priory.
- Admission by Group Authority
  - Group authority must be trusted, need to be online all time.
- Admission by members
  - Voting by existing members.

# Admission Process

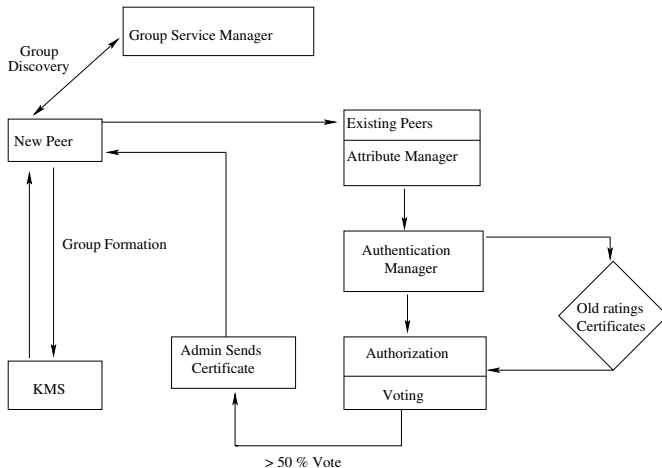


Figure: Admission process

# Admission Process

Various steps involved in admission of new peer:

1. Advertising the created groups
  - Public server making groups info available.
  - Group creator and other members periodically broadcast group advertisement.
2. Request for joining group

$$P_{new} \longrightarrow P_i : \{JoinRequest\}_{PrivateKey}, Cert$$

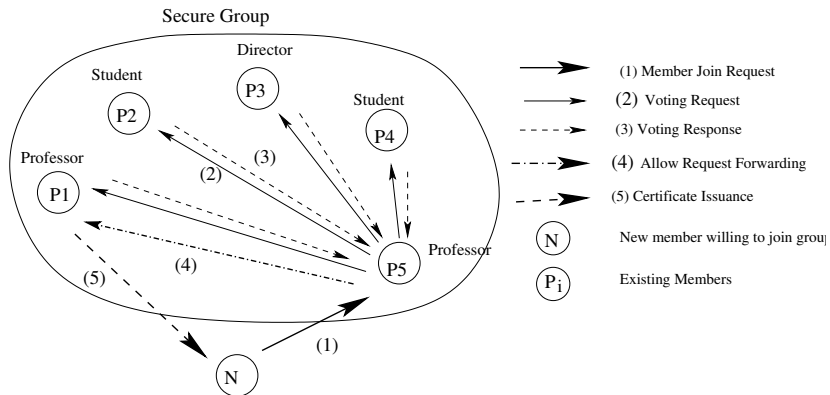
$$Cert = UUID, PK, Rating, \{H[UUID||PK]\}_{PrivateKey}$$

## Admission Process Cont.

3. Authentication of user who wishes to join
4. Collaborative decision by peers
  - Decide based on majority of votes.
  - Look for previous available certificates.

$$P_n \longrightarrow P_i : \{Vote, Level\}_{PrivateKey_n}$$

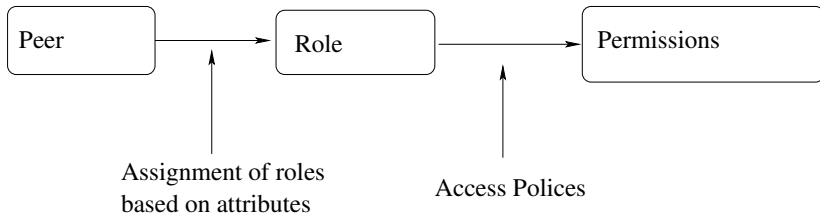
5. Issuance of admission certificates.
6. Group Rekeying



# Access control

Rules to decide upon what resources are accessible to a group member. Two categories:

- Role Based Access Control (RBAC)



- Attribute based access control (ABAC)

## RBAC vs ABAC

**RBAC** : Permissions are associated with roles not with individual peer.

**ABAC** : Permission rules are functions of User's identity and authenticated attributes.

- For hierarchy based access control group members need to be categorized to different Roles.
- Fine grained access control involves multiple attributes, as this number increases number of roles and permission sets increase exponentially.
- RBAC might need roles to be stored on some centralized server.

# Hierarchical access control

- Peers organized in hierarchy, (similar to assigning roles)
- Different keys for different hierarchical level.
- All the members from level  $H_i$  communicate among themselves by encrypting messages with  $K_i$ .
- When N joins group rekeying is done for N's hierarchical level ( $H_i$ ).

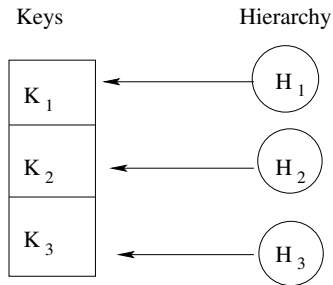


Figure: Hierarchical access control

# Member level updation

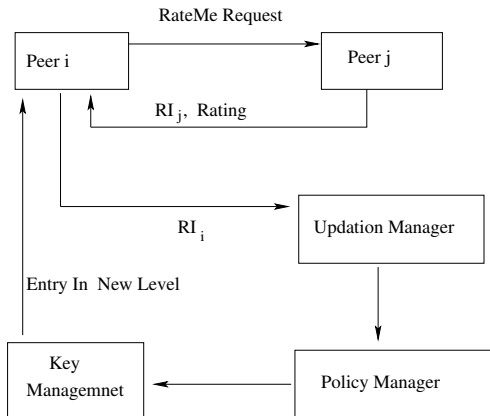


Figure: Level updating procedure

# Member level updation

Reputation index of peer is calculated as :

$$R_i = (R_j * T_{ij}) / \sum T_{ij} \quad (1)$$

which depends upon

- Recommending peer's identity
- Recommended peer's identity
- Recommending peer's own rating ( $R_j$ )
- Rating given by recommending peer ( $T_{ij}$ )
- Signature of recommending peer.

# Key Management

Key shared by group members.

- Message encryption
- Rekeying, Security requirements :
  - *Forward Secrecy* : Allows new member to decrypt future message but not previous messages.
  - *Backward Secrecy* : Peer should not be able to access messages prior to his joining.
- Scalable : Large group sizes.
- Dynamic : Multiple join and leave.

# Key Management

## Classification:

- Centralized vs Distributed Key Management
  - Central party to be Trusted third party or Group authority
  - Group key generated by contribution from all group members.
- Public-key based vs. secret-key based

## Various key management protocols:

- Ring Based Approach
- Hierarchy based cooperation
  - Skinny Tree (STR) protocol)
- Broadcast based approach

# STR Protocol

Each member  $P_i$  comes up with random number  $r_i$  and calculates  $br_i$  as  $br_i = g^{r_i} \text{ mod } p$

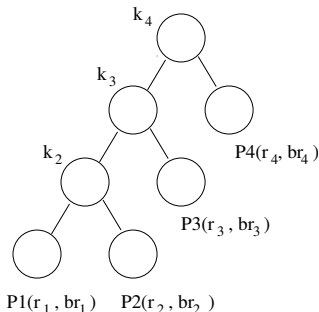
Using Diffie Hellman

$$k_2 = br_2^{k_1} \text{ mod } p = g^{r_0 r_1} \text{ mod } p$$

$$k_3 = br_3^{k_2} \text{ mod } p$$

$$k_n = br_n^{k_{n-1}} \text{ mod } p$$

Tree is built recursively from bottom to top.



Final group key computed is:

$$g^{k_n * g^{k_{n-1}} \dots g^{k_2 * g^{k_1}}}$$

## Example Scenario

Consider a multi-chat application in IIT based on Secure group communication.

- Multiple session exist, each implemented as group.
- Various user join session in different roles e.g Professor, student, TAs etc.
- There exist multiple rules governing session.
  - Professor communication should be limited to them only.
  - A peer with higher hierarchy might be directly allowed to join without voting.
- Members capable of updating there level through feedback mechanism.

## Peer to Peer architecture : JXTA

**JXTA** : An open, network computing platform designed for peer-to-peer communication.

JXTA protocols :

- Finding peer on network.
- Peer status information.
- Creating, joining, leaving of peer groups.
- Routing messages for peers.

Independent of transport protocols. It can be implemented on top of TCP, IP, HTTP.

## Functional components

- **Attribute manager** : Stores the attributes and PKs of group members
- **Authentication manager** : Verifies identity and invokes voting
- **Access policy manager**
- **Key management system** : Handles rekeying
- **Feedback manger**: Calculating rating (reputation is function of feedback from other peers in form of rating which is based on his performance in the group.
- **Updation manger** : Responsible for level updation.

# JXTA messages

- Broadcast messages
  - Not reliable
  - sent to everybody in group
  - ex: join, leave notification.
- Unicast messages
  - Pipe to pipe message exchange
  - ex : Voting request, feedback

## Event handlers

- One responsible for handling broadcast messages, analyzing, sending response accordingly.
- Other responsible for listening to unicasts messages.

# Messages involved in Multichat client

Messages uniquely identified by description field.

## Unicast Messages

Request Join Group, Request Voting, Request Join Group Granted  
Forward, RateMe, UserPublicKey, OldKeys

## Broadcast Messages

Chat messages, Join and leave notification, Postleave

## Periodically sent messages

- Peers broadcasting their identities
- Publishing their group advertisement

# Policy Language

Standardizing rules governing admission and access control.

- Policy : The combination of rules and services where rules define the criteria for resource access and usage.
- Policy Control : The application of rules to determine whether or not access to a particular resource should be granted.

## XML: Policy language specification

- Extensible, customizable, searchable.
- Specification language for describing peers, peer groups and services.
- JXTA group advertisements and other messages are in XML format.

# Policy language

```
<?xml version="1.0" encoding="UTF-8"?>
  <rules>
    <request type="join">
      <min parameter="votecount" type="percentage">50</min>
    </request>
    <request type="update">
      <newlevel id="1">
        <min parameter="rating" type="number">6</min>
      </newlevel>
      <newlevel id="2">
        <min parameter="rating" type="number">8</min>
      </newlevel>
    </request>
  </rules>
```

Figure: Sample policy file

# Framework

Graphical Interface

Dynamic Access control based on  
Hierarchical arrangement of peers

Policy Language framework

JXTA Services and Core

## Conclusion and future scope

- We implemented secure multichat client in completely distributed manner which authenticates independent of central server by using public-private key pair at small overhead of space.
- Absence of central server also reduces number of exchanges between peers for admission but trust factor comes into play..
- Notion of subgroups for different hierarchies is achieved by maintaining separate keys at each level.
- Trust establishment using feedback mechanism.

### Future scope

- Implementing multiple join-leave.
- Making policy language dynamic and open to wide variety of rules.

## References

-  Yu Zhang, Xianxian Li, J. Huai, Y. Liu : *Access control in Peer to Peer collaborative Systems*, ICDCSW'05, IEEE-2005
-  Y. Challal, H. Seba : *Group key management protocols: A novel taxonomy* IJIT 2005
-  J. F. da Silva, L. P. Gaspar, M. P. Barcellos and A. Destsch : *Policy-based access control in peer to peer Grid systems* Grid Computing Workshop 2005 , IEEE-2005
-  Yongdae Kim, D. Mazzocchi, G. Tsudik: *Admission control in peer groups* IEEE, Network Computing and Applications, p.131, April 16-18, 2003
-  Y. Sun, K. J. Ray Liu: *Scalable hierarchical access control in secure group communications* INFOCOM 2004. IEEE, March-2004

Thanks !!

**Questions ?**

# Screenshots

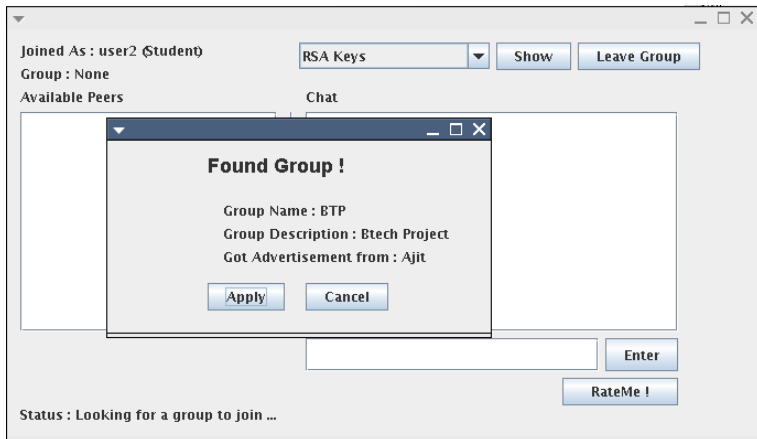


Figure: Found a group advertisement

# Screenshots

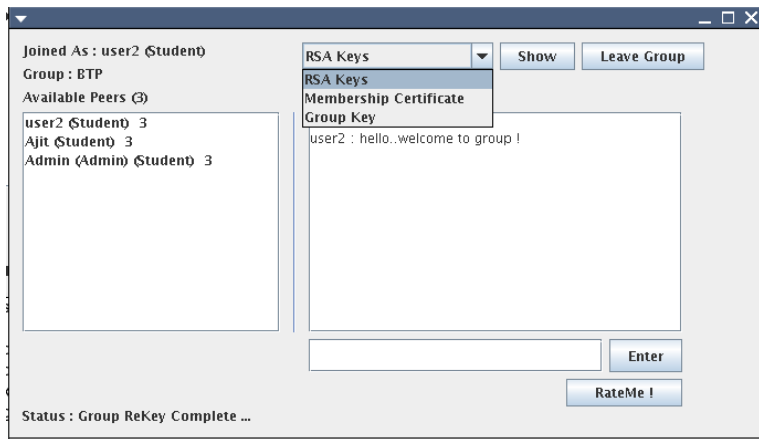


Figure: Chat client

# Screenshots

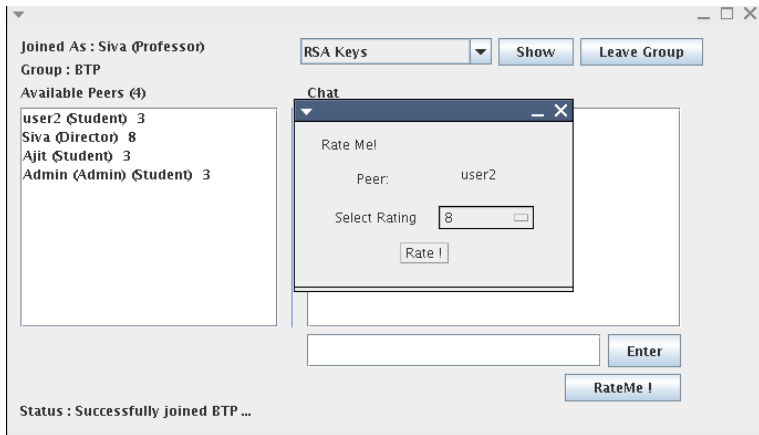


Figure: Rating request

# Screenshots

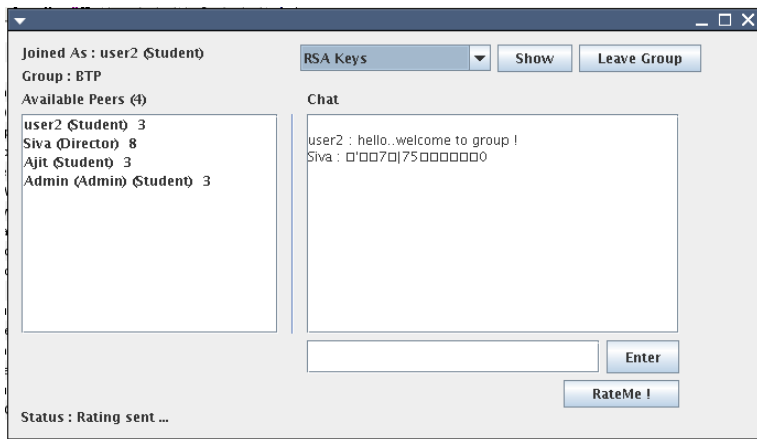


Figure: Rating Updated

# Screenshots

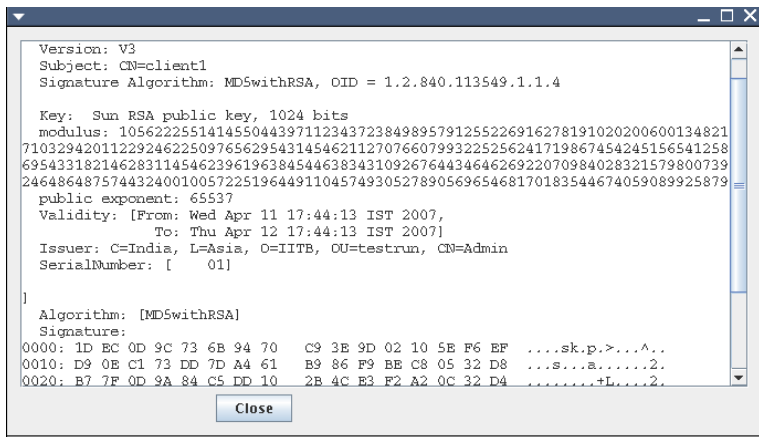


Figure: Certificate