Integrated Framework for Authentication and Access Control in Peer to Peer Groups

Madhumita Chatterjee (04429802) Advisors: Prof M. Bernard & Prof G. Sivakumar.

1/24/2007
Previous APS

- Issues in Secure Group Communication
- Group Key management Taxonomy
- Comparative Survey
- Need for Admission Control & Authentication
Outline

- Peer-to peer groups
- Admission and Access control
- Related work
- Proposed Integrated Framework
- Protocols for member join and level updation.
- Hybrid access control policy
- Future work
Introduction

- Popularity of Web based collaborative groups
- Peers with common interests create interest groups among each other
- Group is governed by a set of rules e.g. Yahoo Groups, Google Groups
- Applications like --- file-sharing, online gaming, video/audio conferencing, collaborative workspace, virtual meetings etc.
Advantages of peer to peer

- **Scalability:**
- **Reliability:** No single point of failure
- **Self-organization:** Autonomous decisions to adapt to different loads
- **Resource aggregation:** Take advantage of existing resources
Security in collaborative groups

- governed by
  - authentication & membership control
  - access control
  - key management.
Access control

Static access control

- Identity based access control
- Role Based access control
  - Non-authenticated
  - authenticated
- Attribute based access control
Access control Cont...

Dynamic Access Control

- Trust Based access control
  - Reputation Based
    - Ebay online auction
    - Kazaa file sharing
  - Global reputation based
    - Eigen Trust
    - P2PRep – peers polling for reputation
Drawbacks of reputation based model

- Ebay—relies on centralised system to store ratings
- Overall reputation is summation of ratings over 6 months
- Sellers may gain good reputation fast but may default on larger orders
- Kazaa—good behavior is rewarded but bad behavior is not punished.
Eigen Trust

- Local trust is calculated as
- $s_{ij} = \sum tr_{ij} = (sat(i,j) - unsat(i,j))$

- Global trust is
  - local trust value assigned to peer $i$ by other peers, weighted by global reputations of assigning peers.
  - Value is normalised between 0 and 1
Drawbacks of Eigen trust model

- Does not differentiate between:
  - peers with whom peer $i$ did not interact,
  - or peers with whom peer $i$ has poor experience

- Does not take into account:
  - user dynamics
  - user credibility
Related Work in SGC

- Antigone project includes a flexible framework for secure group communication
  - utilizes a centralized member admission approach
  - not designed for peer to peer
- Secure group layer SGL
  - access control mechanism is not dynamic or scalable.
Related work....

• Secure Spread
  ■ distributed group key generation protocol
  ■ does not provide any authentication or group access control mechanisms

• Kim et al proposed an admission control framework for peer to peer groups
  ■ scheme lacks the attributes of peers
  ■ all members have equal access rights,
Scenario

- Massively Multiplayer Online Games (MMOGs)
- Video/audio conferencing
- Collaborative news-groups
- Secure Messaging
Motivation

- Peers in self-organizing groups with multi-levels should have the right to:
  - dynamically change access levels
  - collaboratively modify the access control policies governing them
  - allow incremental building of trust during the communication.

- Should support:
  - Dynamic authentication of participants who were previously part of group.
Motivation…..

- Need for framework which integrates authentication, access control and key management in decentralized groups
Proposed Integrated Architecture

- Every peer $P_i$ has a unique user identity $\text{UUID}_i$
- Peers should be individually capable of performing the tasks of authentication, admission control, access control and key management.
- Each peer can have different functional components
Functional Components of a peer

- Attribute Manager
- Authentication Manager
- Authorization Manager / Policy Manager
- Trust Engine
- Updation Manager
- Key Manager
P2P Collaborative Applications

Policy Manager

Trust Engine
Feedback Engine

Updation Manager

Group Rekeying

Multi Level Access Control

Services

Attribute Manager

Pseudo certificates
Rating certificates
User IDs

Authentication Manager

Encryption, Hash, Signatures

Authorization Manager

Voting, Group Key Agreement

1/24/2007 Integrated Framework 19
Phases in Proposed Model

- Group Initialization
- Admission Request
- Authentication
- Voting and Authorization
- Access control and Trust Building
- Key Management
- Per-Session Authentication
Cases

- **New** peer wishing to join an existing group.
- An **existing** member of the group who wants his access level **updated**.
- A peer who was **previously** part of the group and wishes to rejoin in a new session.
- Dynamic **leaving** of peers from the group.
Notations used

- $P_{\text{new}}$: New Peer
- $P_i$: Existing Peer
- $SK_i$: Private key of $i$
- $PK_i$: Public key of $I$
- $\text{UUID}_i$: Unique user ID of $i$
- $\text{Cert}_i$: Certificate of $i$ which contains the ID, Public key, Level/rating, Validity period.
Protocol for New Member Join

Diagram:

- New Peer
  - Group Discovery
  - Group Service Manager
  - Existing peer
    - Attribute Manager
    - Authentication Manager
      - Old rating cert exists?
        - Yes
          - B
        - No
          - Old rating cert exists?
            - Yes
              - B
            - No
              - Voting
  - KMS

Condition: > 50% votes
Steps in Protocol

1. Group Initialization and Advertisement
   - Periodic broadcast by existing peers
   - Details of group included in group charter

   • New peer wishing to join searches for relevant group
2. Admission Request

- \( P_{\text{new}} \rightarrow P_i : \{ \text{JoinREQ} \}_{SK_{\text{new}}} , \text{Cert}_{\text{new}} \)

- \( \text{Cert}_{\text{new}} = \text{UUID}_{\text{new}}, \text{PK}_{\text{new}}, \text{Rating}_{\text{new}}, \{ H[\text{UUID}_{\text{new}} || \text{PK}_{\text{new}}] \}_{SK_{\text{new}}} \)

- Digital signature serves as credential for joining peer
3. Authentication

- Receiving peer computes hash and verifies possession of secret key
- Broadcasts voting request
- **Co-signs** certificate of requesting peer
- \[ P_i \rightarrow \text{All}(P_n) : \{\text{JoinREQ}\}_{SK_{\text{new}}} , \{\text{Cert}_{\text{new}}\}_{SK_i} , \text{VoteReq} \]
4. Voting and Authorization

- peers all verify the authenticity of the new peer
- return the results of voting to Access Policy Manager of the peer who had initiated it.

- $P_n \rightarrow P_i : \{\text{Vote, Level}\}_{SK_n}$.
- Policy manager issues GC signed with $G_{\text{key}}$ of appropriate level
- $P_i \rightarrow P_{\text{new}} : \{GC_{\text{new}}\}_{G_{\text{key}}}$
Format of $GC_{\text{new}}$

- The certificate would contain
  - $\text{UUID}_i$ and $\text{PK}_i$
  - Access Rights– Level, rights, allow, deny
  - UUID of issuer
  - Signature of issuer
  - Expiry Date and Time
5. Key Management.

- Peer receives membership certificate encrypted with group key of proper level
- Submits this to key management component to invoke group rekeying
Protocol for Existing Member Level Updation

- Periodic request for rating.
- Appropriate access privileges granted based on a scoring system.
- A rating certificate used as a means of recommendation.
- Local Trust calculator available with each peer to compute the rating of every other peer.
Rating Certificate

- Recommending peer’s identity
- Recommended peer’s identity
- Original trust value
- Contribution score
- Issuing date and time
- Expiry date and time
- Signature of recommending peer
Member Level Updation

Diagram:

- Peer_j
- Trust Engine
- Reputation R_I_j
- RI_j
- Updation Manager
- Entry into new level
- Key Mgmt System
- Policy Manager (is value > threshold?)
Steps

1. Request for Updation
   - $P_i \rightarrow P_j : \{\text{UpdateREQ}\}_{SK_i}, \text{Cert}_i$
   - Cert$_i$ includes rating certificates from other peers.
   - Updation manager calls trust engine and calculates value. Checks for revoked certificates.
   - Gives input to policy manager
2. Access Control

- Policy Manager takes inputs from Trust engine and feedback engine
- Peer with overall trust value > threshold is granted GC of higher level
- Policy itself is flexible so that it can be modified after a certain level
Rating Algorithm

• Factors on which reputation depends are
  ■ Peer Feedback
  ■ Weighted Cost of transaction
  ■ Credibility of peer who is giving the rating.
Trust calculation

- **Direct trust:** $\sum T_{ij}$
  
  peer $i$’s belief on peer $j$ based on direct interaction
  
  $k$

- **Indirect trust:** $T_{ij} = \sum (T_{ik} \cdot T_{kj})/k$
  
  Peer $i$’s belief on peer $j$ based on recommendations from other peer’s ($k$)
  
  $K$ is a number fixed by $i$. 

Weighted Cost

- $\theta$ is an upper bound on transactions
- $\forall$ transactions whose cost or weight is $< \theta$
  - reputation is calculated normally
- For every transaction with weight $> \theta$
  - the reputation value is multiplied by a constant which is a multiple of this threshold value.
Reputation index

- Recommendations from different peers are weighted differently depending on trust level of recommending peers.

\[ RI_i = RI_j \times \sum T_{ij} \times \text{Trans-cost} \]

(for transactions whose cost or weight is > \( \theta \))

(\( \theta \) is upper bound on transactions)
Summary....

- Updation of access level possible within a group
- Voting need not always be invoked
- Opinions of peers are weighted by credibility of rating peer
- Opinions of peers are weighted by cost of transaction
- Evaluation could be restricted to $\theta_T$ trusted responses
3. Key management

- TGDH protocol may be used
- Rekeying at level \( i+1 \) would invoke rekeying at level \( i \) also
Per-Session Authentication of Previous Member

- Existing member has copy of signed recommendations from other peers while leaving the group.
- This could serve as an authentication token
- Policy Manger checks validity period and current member list plus rating in certificate.
Dynamic Leaving of Members

- Peers may send a query message to find out when a particular peer last communicated.
- Periodic re-keying could also be performed.
- Member could periodically assert its presence in a group.
Authentication overhead

- Could be reduced by restricting no of responses required to calculate trust
- Signed hash of rating could be added instead of signing each individually
- Information need not be stored uniformly at all peers....maybe only neighboring peers
Future scope

- Refinement of proposal and implementation of test bed to measure:
  - Latency of join and leaves
  - Computational overhead of storage at peers
  - Functionality of individual peers in a group
  - Effect of authentication and access control on re-keying
  - Effect of dynamic revocation of access rights.
Future scope....

- Composition of groups
  - Groups may be formed with peers having different functional roles/rank.
  - Peer with maximal role may not be highest level peer
  - Quantum of functionality to be assigned to a peer?
  - Effect on the working of the group?
  - How could this be achieved in a distributed manner?
Future scope.....

- Addition of levels in self-organising groups and modification of policies at higher levels
- Dynamic leaving of peers
- Measuring inter-group trust metrics
- Behaviour of system with malicious peers
  - Handling of dishonest feedbacks
Implementation of Secure Multi Chat

- Initial test bed
- Centralized framework
- Voting, security and rekeying added
- Minimum level access policy implemented
JXTA

- set of open protocols that allow any connected device on the network.
- standardize the manner in which peers:
  - Discover each other
  - Self-organize into peer groups
  - Advertise and discover network services
  - Communicate with each other
  - Monitor each other
JXTA framework...
Sample policy in XML

```xml
<rules>
  <rule>
    <requestname>join</requestname><permission>grant</permission>
    <condition>
      <parametername>Rating</parametername><minbound>1</minbound>
      <parametername>vote</parametername><minbound>40</minbound>
    </condition>
  </rule>
  <rule>
    <requestname>updation</requestname><permission>grant</permission>
    <condition>
      <parametername>performance</parametername><minbound>cpi</minbound>
      <parametername>vote</parametername><minbound>60</minbound>
    </condition>
  </rule>
</rules>
```
Policy......

```xml
- <rule>
  <requestname=""read"">permission = "grant"</requestname>
</rule>
- <rule>
  <requestname=""write"">permission = "grant"</requestname>
  <condition>
    <parametername=""rating"">min = "2"</parametername>
  </condition>
</rule>
- <rule>
  <requestname=""modify"">permission = "deny"</requestname>
  <condition>
  ..........</condition>
</rule>
</rules>
```
Joined As: faculty (Professor)
Group: None
Available Peers

RSA Keys
Show
Leave Group

Chat

Found Group!

Group Name: hello
Group Description: testing
Got Advertisement from: Admin

Apply
Cancel

Status: Looking for a group to join...
Joined As: faculty (Professor)
Group: hello
Available Peers (2)
faculty (Professor)
Admin (Admin) (Student)

RSA Keys
Membership Certificate
Group Key

Status: Successfully joined hello ...

KRSIT
1/24/2007
Bibliography


Acknowledgements

- Prof G. Sivakumar
- Prof M. Bernard