Policy Based Framework for Trust Management and Evolution of Peer to Peer Groups.

Madhumita Chatterjee

(04429802)

Advisors:
Prof G. Sivakumar &
Prof Menezes. Bernard
Outlay of Presentation

- Context – Motivation
- Status of Work

- Generic Model
- Trust Model and state model
- Modelling F/OSS for experiments
- Experiments and results
- Thesis TimeLine

3 slides
Context

- Collaborative P2P groups
  - Self organising
    - E.g Multiproject software ecosystem F/OSS

- Security
  - Admission Control
  - Dynamic access Control
  - Adaptive and evolving trust
**Motivation and Context**
- Collaborative peer groups — decentralized, dynamic, self-organized.
- Need for admission control, access control, trust.
- Dynamic Policy based and reputation based access control
- Adaptive and evolving trust for self-organising groups
- Tool to be able to select optimum policy for a specific application

**Work Done**
- Framework and protocols for secure communication in collaborative peer groups
- Dynamic Policy based model for multi-level access control integrated with trust.
- Implemented framework in Peersim and policies in Prolog and interfaced the two with Interprolog. (ACK. dual degree student Ashish Arya)
- Performed several experiments to validate the framework.
- Improved upon system model and proposed generic model with an adaptive trust mechanism capable of handling malicious peers in self-organising groups.

**Publications**
- Integrated Framework for Authentication and Access control in peer to peer groups
  - WISA 2007 (Korea)
  - Status - published
- Dynamic Policy based model for trust based access control in peer to peer applications
  - ICC 2009 (Germany)
  - Status – Published in IEEE
- Dynamic Policy Adaptation for Collaborative Groups.
  - CNSA 2010 (Chennai)
  - Published in Springer (Recent Trends in Network Security)
P2P Groups Integrated Framework

Global meta polices for multilevel access control

Evolution of P2P Groups
- Dynamic Policies
- Adaptive Trust
- Malicious Peers

Peersim Simulator

Trust Engine
- Weighted Context Specific Parameter
- Peer Credibility

Self Trust
Feedback by Direct interaction
Indirect Recommendation
Incentives for rating

Trust Computation

Context based policy
<table>
<thead>
<tr>
<th>E-commerce Domain</th>
<th>Open Source Software Development</th>
<th>Online gaming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy 1</td>
<td>Policy 2</td>
<td>Policy 3</td>
</tr>
</tbody>
</table>

L1
L2
L3

Role 1
Role 2
Role 3

Adaptive Trust
Malicious Peers

E-commerce Domain
Open Source Software Development
Online gaming

Policy 1
Policy 2
Policy 3
Trust Engine

Self Trust

Feedback by Direct interaction

Indirect Recommendation

Incentives for rating

Trust Computation

Weighted Context Specific Parameter

Peer Credibility

P2P Groups Integrated Framework

Authentication  Access Control  Key Mgmt.

Global meta polices for multilevel access control

L1

L2

L3

Role 1

Role 2

Role 3

Role 1

Role 2

Role 3

P2P Groups Integrated Framework

Evolution of P2P Groups

Dynamic Policies

• Adaptive Trust

• Malicious Peers

Peersim Simulator

Context based policy

E-commerce Domain  Open Source Software Development  Online gaming

Policy 1  Policy 2  Policy 3
Generic Model for collaborative groups.

P2P Collaborative Groups

- Set of Peers with common interest
  - Group Charter
    - Group Description
    - Group Activities
    - Meta Rules for membership
    - Group Composition
  - Service Provider
  - Rater

- Set of Activities
  - Group Management Activities
  - Task Oriented Activities
  - Social Tasks
  - Malicious or Destructive Activities

- Set of Events
  - Internal Schedule
    - Periodic
  - External Schedule
    - Event Driven
Group Roles

- **Member peer** A peer in this role is a minimal functionality peer who participates in the normal group activities but does not contribute to the admission of new peers, nor updating of levels of existing peers. Thus this peer is only a service provider.

- **Admission Peers** Those which are allowed to register new group members.

- **Control peer** The control peers are the super nodes of the framework. Every group must have at least one control peer. These peers are responsible for broadcasting essential messages like
  - 1. Joining of a new peer
  - 2. Policy change (if it happens)
  - 3. Updated ratings of peers.
Domain Specific Role

- These are system roles at hierarchical levels, depending on the application domain, access rights/capability and trust value.
- We define Users/Peers(Pi), Roles(Ri), Permissions assigned(PA), Permissions(P).
- \[ R = DR U GR \] where \( DR = \) domain role and \( GR = \) group role and
- \( PA \in P \times R \)
Change of Roles

- Model focuses on the self-organizing and collaborative nature of peers
- any peer which is part of the group may apply for a role change or level change.
- if a CP decides to change its role to MP for some reason, such a peer would lose group membership unless it can transfer control
- If a peer is a single CP then it cannot be allowed to change its role
Group Charter

- defines global policies needed to frame the group
- describes the roles permitted in the group and the rules associated with each role.
- policies for new peer join, peer update, minimum trust/reputation levels, discarding malicious peers etc
- Rules for creating destroying a group
- Signed Hash value of policies stored
Set of Activities

Group Management Activities
- Admission Control
- Access Control
- Publishing of Group Existence/Charter
- Creation of Groups
- Removal of Groups
- Resource Management
- Scheduling Tasks

Task Oriented Activities
- Procedural (function/task of Groups i.e Domain Specific)
- Information/Resource request
- Opinion/Reputation request (Trust)
- Opinion/Reputation response
- Record Maintenance (of Group activities)
- Maintenance of certificates, keys, ratings.
- Voting
- Evaluator/critic (measures group activities against some Group Performance Index GPI)

Social Activities
- Reward actions
- Punish actions
- Observer actions i.e maintaining records of those group activities that help in providing feedback.

Malicious Activities
- Aggressor behavior i.e peers who lower the rating of other peers.
- Recognition seeker i.e colluding peers who call attention to self by providing very good behavior for some time and increasing each others ratings.
Set of Events

- External
  - Join
  - Leave

- Internal
  - Update
  - Task Allocation and Scheduling
  - Policy change
Assumptions.

- The system is **asynchronous** i.e there is no global clock.
- Domain specific rules and policies reside in the **local memory** of a peer.
- Communication primitives used are unicast, multicast and broadcast.
- Reliable message delivery is assumed.
- **No peer symmetry**, i.e peers do not have equivalent functionalities.
Assumptions cont...

- Peers can be queried about their group specific activities/capabilities and other information.
- No distinction is made between node failure, link failure and unannounced exit of a peer.
- Every group must have a well defined Group Charter at the time of creation, which is updated periodically.
Layered and context sensitive reputation model

- **Self Trust**
- **Feedback by Direct interaction**
- **Indirect Recommendation**
- **Incentives for rating**

Weighted Context Specific Parameter ➔ Peer Credibility ➔ Trust Computation
Trust Model

- A peer’s trustworthiness value depends on good behavior and enhances slowly, but drops rapidly in case he exhibits bad behavior.
- If the trust value of a peer falls below a certain threshold value then he is either removed from the group or his level is demoted.
- Calculation of trust value includes a peer’s direct experiences as well as recommendations from other peers and no of feedbacks provided.
Trust Metrics

- Feedback Rating
  - Let $a_i = a_1, a_2, a_3, \ldots, a_n$ be the set of attributes.
  - Let $f_E(x, y)$ denote an evaluation given by peer $y$ for peer $x$ at a transaction.

$$f_E(x, y) = (f_{a_1}(x, y), f_{a_2}(x, y), f_{a_3}(x, y), \ldots, f_n(x, y))$$

where $f_{a_i}(x, y) \in [0, 1]$ is the feedback score given by peer $y$ about peer $x$ for attribute $a_i$.

Relative importance assigned to each attribute can be modelled as weight $w_{a_i}$

$$\sum w_{a_i} = 1.$$
Reputation

- Assume that peer y stores up to n feedback ratings of previous transactions with peer x.

- Then reputation of peer x is calculated by peer y as

\[ R(x, y) = \frac{\sum_{i=1}^{n} \alpha^i \cdot f_{Ei}(x, y)}{\sum_{i=1}^{n} \alpha^i} \]

- where \( f_{Ei}(x, y) \) denotes the \( i \)th feedback given by peer y to x
- \( \alpha \in [0, 1] \) is a decay factor which indicates how important the most recent interaction is to the user.
Direct and Indirect Trust

\[ T(D) = \frac{\sum I(x, y).R(x, y).T(y)}{\sum I(x, y).T(y)} \]

Assume that each user \( x \) receives job ratings for peer \( y \) from \( n \) references \( k = (1, 2, ... n) \) and all nodes use the same decay factor \( \alpha \).

Then user \( x \) can compute the reputation \( f_E(ki, y) \) of each indirect peer \( ki \) to resource \( y \).

\[ T_{ID}(x, y) = \frac{\sum_{i=1}^{n} I(x, y).R(k, y).T(k)}{\sum_{i=1}^{n} I(x, y).T(k)} \]
Global Trust value

- \( T \) is an aggregation of the direct experiences of every peer about peer \( x \) as well as the recommendations received about peer \( x \), and the context factor.

\[
\frac{F_x}{I_{x,y}}
\]

\[
T = \alpha \ast T_{D}(x,y) + \beta \ast T_{ID}(x,y) + \gamma \ast Context\ factor
\]

- \( \alpha \) is the weight associated with direct experience,
- \( \beta \) with indirect reputation
- \( \gamma \) is a fine tuning constant to control the amount of reputation gained by rating others.
Group Charter

- Group Charter defines global policies needed to frame the group, describes the roles permitted in the group and the rules associated with each role.
- Policies for new peer join, peer update, minimum trust/reputation levels etc
- Signed Hash value of policies stored
Task Allocation

- Group could have different tasks at various difficulty levels
- Tasks could be sequential or concurrent with or without interdependency.
- Scheduling of tasks is done to peers based on availability and peer capability
Group behavior measure

- Measure of group behavior could be in terms of:
  - average reputation of group
  - task completion of group with respect to time
  - current group composition i.e. no of expert peers in highest layer of group.
DPDTBAC model

- Global Policies
- Domain Specific Policies
System model as state behavior
Modelling using Prolog

- Use of logic programming language prolog
- Rule is an expression of the form
- \( R_0(\text{uo}) : - R_1(u_1), \ldots , R_n(u_n) \)
  - If Prolog knows that body follows from the information in the knowledge base, then Prolog can infer head.
  - The Prolog inference engine provides a mechanism to derive consistent access control decisions at runtime.
  - New facts from independent policy sources can be added to the policy base before decisions are made, ensuring dynamic decisions at runtime.
Global policy for join

- join(join).
- update(update).
- member(member).
- admission(admission_peer).
- maximal(maximal_peer).

- verify(Request,RI,Level,Rate,SPR,Expert,Tot) :-
  join(Request), admission(Rl), Level =:= 1, SPR > 40.

- verify(Request,RI,Level,Rate,SPR,Expert,Tot) :-
  update(Request), maximal(Rl), Level =:= 2, Rate >= 6, Expert < 20.
Dynamic rules

- Assert and retract clauses of Prolog can be used to express dynamism

update_engine(Levels, MPs, APs, CPs) :- CPs > 20, retract((verify(Npeer, Request, RI, Level, Rate, Vote) :- update(Request), maximal(RI), Level =:= 3, Rate >= 7)), assert((verify(Npeer, Request, RI, Level, Rate, Vote) :- update(Request), maximal(RI), Level =:= 3, Rate >= 9)).
F/OSS Domain

- Represents a group of like minded participants to develop software systems and related artifacts intended to be shared freely
- F/OSS systems, hyperlinked artifacts and tools and project web sites serve as venues for
  - socializing, building relationships and trust, sharing and learning with others.
- Software evolution in a multi-project F/OSS ecosystem is a process of co-evolution of
  - interrelated and inter-dependent projects, people, artifacts, tools, code and project specific processes.
- Tasks in the group
  - development of software modules of different difficulty levels
  - content distribution
  - resource sharing
  - publish/subscribe
  - postings news-group.
Modeling F/OSS

- **Role Set**
- **Group Roles** = \{ MP, AP, CP\}
- **Domain/System Roles** = \{Developer, Reviewer, Moderator\}

- \( R_i = \text{GR U SR} \)
- \( P = \{\text{set of permissions}\} \)
- \( PA \in R_i \times P \)
The attribute set for trust in this domain would consist of parameters like...

- $P \rightarrow$ Coding cost (no of peers)
- $T \rightarrow$ Completion time
- $L \rightarrow$ Lines of Code
- $Q \rightarrow$ Quality
- Thus \{a1,a2,a3 ....\} = \{ P,T,L,Q\}
Modeling rating for role updation

- Cumulative contributions of active developers can be defined in terms of the non-linear equation

\[ C(t+1) = C(t) + \alpha \times P \times R(t) - \beta \times C(t) \]

- Where \( C(t) \) = contribution \( \beta \) at time \( t \)
- \( P \) = no. of generated proposals/submitted code
- \( R(t) \) = current reputation level of developer
- \( \alpha \) = acceptance rate by reviewer/moderator
- \( \beta \) = overriding of prior contributions by new developments
Mapping of Roles and Levels
Task Allocation

- Tasks enter the system randomly or following a Poisson Distribution
- Tasks can be concurrent or sequential
- Task consists of different modules of different difficulty levels
- Job-diff =\{easy, moderate, difficult\}
Modeling task skill-set

- We specify a project over a space of binary strings of length n, where n is the size of the largest set of possible features for the application of interest.
- E.g. the project specified by the binary string.
- \(<0,1,0,0,1,0,1,\ldots,1,0,1>\) depicts those projects that combine features \(f_2, f_5, f_7, f_{n-2}, f_n\).
Overall Simulation View
A project module consists of the following parameters:

- Starting time
- Current time (clock of every module is different and works concurrently)
- Time of completion
- Expected time of completion
- Status (Finished, started, working)
- Bugs (number of bugs)
- No of bugs is a Uniform Distribution
- Module arrival is as per Poisson distribution.
- Average completion time of a module depends on difficulty level.
- Attributes considered while rating the performance of a peer developing some code is
  - No of bugs, Time Taken, Lines of Code.
- Module completion is also assumed to be a Poisson distribution
- Each module has an expected time of completion
### Probability Dist of No of bugs

<table>
<thead>
<tr>
<th>No. of bugs/Type of peer</th>
<th>Expert Peer</th>
<th>Average Peer</th>
<th>OK Peer</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>0.33</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.33</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.33</td>
<td>0.25</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>0.25</td>
</tr>
</tbody>
</table>
Domain Specific Policies

- `update_level(CodesSubmitted, PercentAccept, Level, Role):-member(Role), CodesSubmitted > 6, PercentAccept > 50, Level =:= 1.`

- `update_level(CodesSubmitted, PercentAccept, Level, Role):-admission(Role), CodesSubmitted > 20, PercentAccept > 50, Level =:= 1.`

- `introduce_level(Role, Level, CPs, NPeers, High_Level, Vote):-maximal(Role), CPs > 40, NPeers > 40, Level =:= High_Level, Vote =:= 100.`
Average Rating for oversmart vs honest peers
Job Success with varying Job Compositions
Dynamic vs static job allocation policies
Dynamic vs static job allocation policies
Varying job allocation policies for difficult jobs
Job Success with varying Job Compositions

![Graph showing success rates for different job compositions]

<table>
<thead>
<tr>
<th></th>
<th>Easy</th>
<th>Medium</th>
<th>Difficult</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compos1</td>
<td>20</td>
<td>20</td>
<td>60</td>
</tr>
<tr>
<td>Compos2</td>
<td>40</td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>Compos3</td>
<td>60</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>
Comparison of 10 percent vs 30 percent malicious peers
Static vs dynamic policies in presence of malicious peers
Future Plan

- Testing the flexible policy framework for evolution of P2P groups in a different application context.
- Improving the model for malicious peer behavior
CONTRIBUTIONS

- Survey on Group Communication and Key Management
- Proposed Integrated framework for authentication, access control and key management in dynamic peer groups
- Proposed Dynamic Policy model using Prolog and integrated with the framework.
- Simulated the framework using Peersim and integrated it with Prolog. Performed preliminary experiments for application F/OSS
- Proposed Generic Model for framework, and adaptive trust model. Performed several experiments to show that group efficiency increases with our dynamic policies.
- Modelling malicious peers and testing. Simulating another application.
Publications....


“Dynamic Policy Based Model for Trust Based Access Control in P2P Applications”

“Dynamic Policy Adaptation for Collaborative Groups”
CNSA 2010, Chennai, India, Published in Springer.
Acknowledgements

- Prof G. Sivakumar
- Prof M. Bernard
References