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

A synopsis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

by

Madhumita Chatterjee
(Roll No. 04429802)

Under the guidance of
Prof. G. Sivakumar
and
Prof. Bernard Menezes

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY–BOMBAY
2012
**Motivation**

Web based collaborative groups are quite popular as peers with common interests form a network among themselves and automatically tend to create interest groups among each other, called communities. Applications like F/OSS, file-sharing, online gaming, video/audio conferencing, collaborative work-space, virtual meetings, distance learning environments, discussion forums and board rooms are examples of applications that are organized as peer groups. Such communities or groups need a communication model in which in which all the peers must collaborate in order to provide the basic services such as content or messages. The group is governed by a set of rules that describe the conditions required to be part of the group. These initial or minimal set of rules or access policies form the group charter.

Such groups generally have frequently changing application requirements and varied security requirements, and require a secure and reliable group communication system to ensure that they *evolve* and *adapt* according to interactions within the group. Peers with common interests and functionalities can join these groups subject to acceptance and authentication. Depending on their functionality and capability peers could join at different hierarchial levels. While authentication is required to allow only authorized members to join the group, access control is needed for restricting rights at different levels and for different roles within a level. In the dynamic scenario where peers are constantly leaving and joining this becomes a difficult task. In the interests of the group there is a need for peers to collaboratively modify access control policies dynamically. Traditional authentication and access control are effective only in situations where the system knows in advance which users are going to access and what their access rights are. Another major challenge for collaborative P2P systems is the ability to manage risks involved in interacting and collaborating with previously unknown and possibly malicious peers. In such cases it becomes necessary to develop strategies for establishing trust among peers.

An interesting feature of collaborative groups is that they tend to organize themselves into different system models. A system model defines the behavior of peers, different roles, events, and policies for membership control, access control, task assignment, security and others. When providing access control during an application, the security context(e.g., the policies, team members and/or roles) may change. Such changes must also be controlled and fed into access control system. While evolving, peers thus have to decide which policies to chose and which system model to follow.
Consider an example of a developers community like F/OSS for some critical security related open-source software. The communication forum could be either chat or e-mail. New users would be allowed into this forum to put in their suggestions, only if they are introduced by an existing member. Depending on the sensitivity level of the code being developed, there could be hierarchial levels for the members of this forum. New members and existing members could be periodically rated or evaluated by existing members based upon the quality of their contribution. A member receiving a high rating value could be elevated to a higher level in the forum. A peer or a member in any level need not always be evaluating other peers, i.e a member can have different roles in a group. For critical software like security applications, authentication of developing members would be important as well as deciding which member is allowed to play which role and participate in which level.

Depending on the group composition and the current skill set and average trust value of peers, such groups might warrant a change in policy for secure admission control or access control. A group which had initially permitted only three roles might perform better after some time if there was an additional reviewer role or if intermediate code were submitted to a reviewer instead of complete code. Thus peers need to be able to collaboratively frame and revise the policies for group management.

**Problem Statement**

Collaborative groups are heterogeneous, dynamic, decentralized and large-scale, with large number of autonomous entities wishing to access and share resources in a secure and controlled fashion. For a group to survive it must adapt to the environment and to the behavior of the peers. The major challenges for collaborative P2P groups are given below.

1. One main challenge of P2P groups is Admission control [1, 2]. Most prior work on peer group security has focussed on key management and authentication. A group membership control mechanism must guarantee that the group members are approved to join the group by satisfying the admission criteria. Admission criteria are group specific and form an integral part of the group security policy. A prospective member must learn these rules, hence they need to be specified in a readily available document, referred to as the Group Charter. In addition to group charter a well defined procedure to admit a new member to a group is needed. In a static group this can be achieved by a pre-defined access control
list ACL. However in dynamic groups this would not work.

2. In collaborative groups the authorisation policies (i.e., the rules governing who can access which resources) can be extremely complex, and may partially rely on and interact with policies of other users, and they can frequently change. Access control therefore cannot be solely based on identification and authentication of individuals [3, 4]. In the interests of the group and for its survival there is a need for peers to collaboratively modify group policies dynamically and the system should be able to evolve and adapt by switching between different policies. As groups evolve, there is a need for policies also to evolve and the group must adapt to these evolving policies for optimum performance. These changes should be within the frame-work of a well defined group charter.

3. Collaborative groups tend to organize themselves into different system models. A system model defines the behavior of peers, different roles, events, and policies for membership control, access control, task assignment, security and others. While evolving, peers have to decide which policies to chose and which system model to follow. For a collaborative group like Wikipedia for example, the group could start with an initial system model defining three roles say readers, editors and administrators. As the group evolves the admin peers may feel the need to introduce roles like proof readers or translators, which means that the group needs to change its system model. The system model or process workflow of the group sometimes needs to undergo a change for optimum performance. Thus the group should be able to re-organize its structure dynamically by adding or deleting roles, and/or changing the process workflow.

4. Another major challenge for collaborative P2P systems is the ability to manage risks involved in interacting and collaborating with previously unknown and possibly malicious peers. Some peers might be buggy and cannot provide services as they advertise. Some might be malicious by providing bad services to get more benefit. Since there are no centralized nodes to serve as an authority to supervise peer behavior and punish peers that behave badly, malicious peers can get away with their bad behaviors. How to distinguish potential benevolent peers from potential buggy or malicious peers is a current area of research. The way to resolve this problem is by building trust and reputation [7, 8]. However the trust context could vary from transaction to transaction and from communities to communities. It is important to build a reputation based system that is able to adapt to
different configurations and different situations. In order to enable practical information sharing in such decentralized and dynamic systems, a viable trust model needs to be incorporated that will allow peers to have varying amounts of dynamically changeable trust amongst each other. The main challenges that need to be addressed are: how to describe if a peer is trustworthy, what low-cost verification algorithm can be executed by a peer to determine the trust value of some other peer, how are trust values about peers exchanged within the system, how can dishonest peers be punished.

5. Another major problem is resource management and scheduling problem.

- How to efficiently motivate peers to provide satisfactory services to users
- How to guide users to select reliable resources in the presence of malicious peers.

To address above mentioned challenges of collaborative peer groups, our goal was to design an integrated system with a dynamic policy based access control model based on an adaptive trust mechanism.

Solution Approach

Traditionally, policy-based management models have considered that roles, their membership and the policies assigned to them can be determined statically in advance by the administrator. However such models are not suitable for distributed and collaborative peer groups as they require anticipation of potential behaviors, events and their likely outcomes.

In this thesis we have designed and implemented an integrated framework (see figure 1) for collaborative groups for authentication, admission control and access control. The framework has a novel method for policy based management of groups for membership control and access control where it is possible to deploy access control policies flexibly and dynamically. The initial group policies are defined in a Group Charter, normally by the creator of the group. Based on the behavior and trust level of peers in the group and the current group composition, it is possible for peers to collaboratively modify policies governing their level.

One possible approach to handle the dynamic and unreliable behavior of peers is to provide an environment for different functional roles which are self-organizing. We define two types of roles viz. Group Roles and Domain Specific Roles which differ in the context of a peer group. Peers can be allowed to dynamically make decisions based on specific conditions and assume
additional functionality to ensure that the group reaches a certain satisfaction level. Peers can dynamically switch between roles. Peers can collaboratively modify policies based on current group composition and trust level of existing peers. New polices can be added dynamically by peers within the framework of the original group charter. The group policy can also dynamically prioritize requests for join. Low level join requests can be postponed in a group which has a certain threshold of peers already at the same level, by changing the join policy dynamically. Peers in one level are also allowed to be dynamically updated to a higher level. Update policy also depends on current group composition and behavior. A peer could also be collaboratively ejected from a higher level to a lower level or even out of the group if his behavior in the group degrades. It is also possible for peers in the highest level to introduce a new level in the group if the situation so demands. Multiple policies can apply to a group, and the system can switch between policies over time, influenced by the system’s state or environment. Some policies can have higher priorities than others. Thus the peers are able to self regulate and self organize, based on trust and group composition.

The diversity of emerging applications with differing security requirements has led to the development of policy languages supporting a wide variety of policy-related constructs, e.g. role hierarchies, delegation, appointment, or separation of duties. For the peers in a group to be able to dynamically modify the admission and access control policies a declarative language offers more advantages. Our policy language is thus a declarative language based on Prolog, where we can model dynamic policies depending on changing trust level using assert and retract clauses. Some work has been done to address the need for dynamic policy adaptation such as [9, 10]. Previous research work has focussed on changing group policies dynamically within the same system model. We feel this is a limitation as the group should be able to reorganize into a structure based on a different system model if need arises. This is one main focus of our work, where groups can dynamically switch between different system models to optimize performance. We have been able to achieve this using statecharts.

StateCharts [12] have been proposed as high end specification techniques to represent complex systems for discrete event modeling. They were originally designed to represent and simulate real time systems [11]. They have interesting features to describe complex systems, such as notions of hierarchy and parallelism. We were able to adapt statecharts to express the dynamic behavior of collaborative groups. We were able to integrate the dynamism of Prolog with the adaptive behavior of collaborative groups using statecharts. Each composite
state in the state chart depicts the group in a different system model. Within a composite state governed by a system model, the peers optimise group performance by dynamically switching between different policies and thus different sub-states. The group evolves by transiting from one composite state having a system model $SM_1$ to another composite state with system model $SM_2$ which could have a different set of policies, roles, levels and so on.

We define an abstract function Group Health which is domain specific and depends on the achievement of Group goals. Peers monitor the group health by evaluating different parameters like average trust value of peers, rate and efficiency of task completion, rate of join, rate of leave, rate of task allocation, group composition and so on. These parameters cannot be known apriori and have to be periodically measured by peers as the group evolves. When the performance of the group falls beneath a certain threshold, the framework allows the group to dynamically change its entire behavior and adapt to a different system model, which may have different number of roles, levels and different policies, or even a different trust model. Thus
the framework is adaptive and supports re-organization of the system model itself. In this new
model again the peers can self regulate their behavior based on an adaptive trust mechanism and
the system can dynamically switch between different policies. The reorganization depends on
the qualities of peers joining the group and their behavior within the group. This is controlled
to some extent by policies for admission and access control. Thus we are able to achieve both
dynamic policy adaptation as well as system model adaptation, enabling the group to evolve
and adapt to evolving policies. We show that this dynamic adaptation and switching between
different system models and different policies helps to optimize group performance.

We propose an adaptive multi-dimensional trust model with a hierarchy of privileges
which builds upon the Peertrust [6] model. Our trust metric combines context specific attributes,
self-proclaimed ratings and recommendations received from other peers. Parameters of the trust
metric can be dynamically adjusted. We incorporate various attributes for feedback to add mul-
tiple dimensions for peer selection, into the basic trust metric with a weight function. We assign
weights with different attributes depending on the type of application. We include a decay
factor as trustworthiness is not always the same and may change with time. We incorporate
different methods to calculate the credibility of the peer who is giving feedback, based on the
situation. We propose a dynamic trust policy which dynamically assigns weights to different
trust attributes. Our trust model along with our dynamic trust policy helps us to slowly remove
malicious peers whose behavior is dependent upon a specific pattern. We compare the results
of our model with the P2P Trust model [5] which calculates trustworthiness of a peer using di-
rect reputation, and indirect recommendation. We show that our model handles malicious peers
more efficiently and their effect is mitigated and they are gradually ejected from the group.

Simulations and Results

To model our P2P collaborative group we chose the self organizing group F/OSS which rep-
resents a group of like minded participants to develop software systems and related artifacts
intended to be shared freely. We performed different simulations to prove the validity and ca-
pability of our framework.

The simulator starts with some initial peers in the group, and then events such as join,
leave, update are triggered. The events have been triggered in the event mode of peersim.
Another event which is triggered is the entry of different job modules into the group. A module
could have different difficulty levels associated with it such as easy, moderate or complex. A module has parameters like Starting time, Current time, Expected Time of completion, Bugs, and Difficulty Level. Peers are assigned modules based on the current active job allocation policy. A module can have more than one peer working on it at a time. As soon as a module is finished, those peers will be allocated some other modules to work on. A priority queue is maintained based on the current time of start of a module so as to maintain the concurrency between the modules.

Peers join and leave the group dynamically. In the initial set of experiments we tested the behavior of peers with varying join policies and varying job allocation policies. Various runs show that the group performance varies with changing group composition and evolving trust of peers. Thus a policy which may give good results initially may not be viable after the group size increases or decreases drastically. We then show that when peers have the flexibility to change the policies dynamically depending on various group parameters, the group gives optimum performance.

In the next set of experiments we gradually inject malicious peers into the network. We conducted the same set of runs with varying policies in presence of 10 percent and 30 percent malicious peers. It was observed that the dynamic policy gave better results than the static in the presence of 10 percent malicious peers. However one surprising observation was that when the malicious peer percentage was increased to 30 percent the group took more time to converge in the presence of the dynamic policy except when all easy jobs were allowed to enter the system. This allowed us to conclude that the dynamic behavior has to change depending on the type of peers and their trust value acquired while performing in the group as well as the type of jobs entering the system. Thus only dynamically switching between different policies is not effective for optimizing group performance. The process workflow or the system model of the group needs to be changed at times to achieve the desired optimum results.

In the next set of experiments we test the behavior of groups in presence of different system models. The initial group charter specifies some process workflow which we define as a system model and peers behave according to that model and change the policies dynamically as in the first set of experiments. We go on to show that under specific conditions just the change of a policy does not lead to optimum performance and it is desirable that the group changes its process workflow and adapts to a different system model. Our tool permits the peers to collaboratively decide and switch between system models when required. We discuss some sample runs for
two system models. System model $SM_1$ has three domain roles viz. developer, reviewer and moderator whereas in system model $SM_2$ we define an additional domain role of an intermediary reviewer. The type of jobs entering the system for the run was 10 percent easy jobs, 80 percent moderate jobs and 10 percent difficult jobs. Some sample graphs are given below.

Graph in figure 2 is a comparative plot of the system behavior with system models $SM_1$ and $SM_2$ in presence of moderate jobs. The thin lines are a plot of time taken versus percentage completion of jobs and the boxes indicate the cost associated with percentage completion of jobs. It is observed that for 100 jobs with job composition 10-80-10, the time taken to complete jobs in single review model is more as compared to double review model. On the other hand the cost associated with double level of reviewing is more as compared to that with single review level.

![Figure 2: Cost metric vs time metric](image)

![Figure 3: Dynamic switching between system models along with cost metric](image)

The graph in figure 3 shows the behavior of the system with dynamic switching between
the models. The group starts in system model $SM_1$. After about 40% job completion, since the group performance index (GPI) is less than the required threshold the group switches to another model $SM_2$ resulting in more optimum performance, and then again switches back to model $SM_1$ after around 80% job completion to balance the cost. The cost associated with two levels of reviewing is much more as indicated hence the group tries to maintain a balance between time taken and optimum cost.

We performed similar experiments on varying group sizes for varying job compositions and various group management policies.

To bring out the effectiveness of our trust model we performed a comparative analysis with the TM [5] model on a network of 100 nodes with 800 transactions in each simulation. The parameters used for comparison were

1. Average trust value of good peers
2. Average trust value of malicious peers
3. Transaction success ratio in presence of varying percentage of malicious peers
4. No of times the malicious peers are selected for transacting with increasing percentage of malicious peers.

A sample graph in figure 4 shows that in our model the average Trust value of malicious peers has a gradual degrading curve whereas in case of TM model the average trust value of malicious peers is more or less constant and does not change much with increase in % of malicious peers.

![Figure 4: Aggregate trust values of peers with 10 percent malicious peers](image)
Other graphs show that in our model the successful transaction ratio does not degrade with increase in malicious peers while in the TM model it degrades considerably.

We also show that in our model as the percentage of malicious peers increase, the service selected of malicious peers decreases whereas in case of the TM model with increase in percent of malicious peers service selected of malicious peers also increases.

**Contributions of the Thesis**

We have designed an integrated framework for collaborative groups which has a novel policy based method for membership control, access control and trust management. The framework is flexible and peers are able to self regulate their behavior in presence of dynamic policies. It supports dynamic adaptation of policies and dynamic re-organization of group structure by switching between system models. Main features of our framework are:

- A dynamic policy driven trust based access control model. Dynamic policies are written using a declarative language (Prolog), so that they can be easily modified without changing the code. We have also designed a GUI based converter for converting text based input to Prolog format to assist the user for writing the policies.

- Ability to represent varying system models for collaborative groups. This has been achieved using statecharts.

- Framework supports dynamic adaptation of policies and re-organization of group structure by switching between system models. We have designed a tool capable of achieving this dynamic policy adaptation and optimization by dynamically changing the group system model itself. Our tool allows us to test the evolution of peer groups based on different policies. We give a statechart model to capture the dynamic adaptation of varying system models and policies.

- Our experiments show that dynamic policies based on the adaptive trust and changing group composition lead to better group efficiency as compared to static access control policies. Further the dynamic adaptation of system models helps to optimize group performance. Our framework enables users to evaluate the effect of different dynamic policies on groups and adapt to the right set of policies for evolution of the group. It was observed that even if the group begins with a system model and a set of policies that does
not give optimum performance, it ultimately adapts to the system model and policy set achieving optimum performance.

- We give an adaptive multi-dimensional trust model for P2P applications. Experiments show that the model is able to detect malicious peers and mitigate their effect on group performance. Our trust policy is able to adaptively select different trust models depending on group health.

PUBLICATIONS


Bibliography


