Volume 2 No. 3 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences
©2010-11 CIS Journal. All rights reserved.
http://www.cisjournal.org

Trust Based Authorization Framework for Grid Services

Sarbjeet Singh
Computer Science and Engineering, UIET
Panjab University, Chandigarh, India – 160014
sarbjeet@pu.ac.in

ABSTRACT

Grid computing allows sharing of services and resources distributed over geographically dispersed, heterogeneous, autonomous administrative domains. As a domain generally has no idea about the trustworthiness of other domains, it may hesitate in accessing shared services and resources provided by other domains. Accessing resources and services from untrusted domains may pose dangerous consequences to the source domain. Trust is an important parameter in achieving faithful domain to domain interaction. Domains must be able to determine the trustworthiness of each other for the access of a particular service. Domains must also provide trust based access to resources and services that they expose in the environment. This paper describes different facets associated with trust issues among different entities in a grid environment and proposes a trust model to establish and manage trust relationships. The trust model provides support to calculate direct as well as recommended trust. Based on this model, a trust based authorization framework is proposed that can be used to provide trust based access to grid services. The goal of the model is to encourage trust based domain to domain interaction and increase the confidence of domains in accessing shared resources provided by other domains. The framework has been implemented in .NET environment with the support of WSE 3.0 toolkit. The framework has been evaluated by implementing a scenario that involves enforcement of different trust policies. The time taken by the enforcement component to evaluate trust policies has been noted. The results obtained from the implementation imply that the approach is workable and can be used to provide trust based access to grid services.

Keywords: Grid *computing; trust relationships; grid services; trust based authorization framework.*

1. INTRODUCTION AND BACKGROUND

Grid computing systems enable sharing of resources and services distributed over geographically dispersed, heterogeneous, autonomous administrative domains [1], [2]. The context of grid computing introduces challenging trust related issues as both, resource providers and resource consumers can come from mutually distrusted administrative domains and any of them can behave maliciously. The usage of grid system can become severely limited if participant are not given any means to access the trustworthiness of other participants in the environment. To achieve faithful domain to domain interaction and confidence of participants in accessing and providing resources from/to other administrative domains, it is extremely important for domains to address trust issues by introducing a trust model and integrating that model into authorization framework to enable trust based access to resources.

Trust is a complex concept and can be defined in different ways at different levels. It is a subjective and elusive notion. The concept of trust is excessively complex and appears to have many different meanings depending on how it is used [3]. Trust is a multifaceted issue that may be related to other attributes such as reliability, accuracy, honesty, risk, competence, security, belief, perception, utility, benefit, expertise etc [3], [4]. Trust is generally defined as having confidence that a service/resource will behave in an expected manner despite the lack of ability to monitor or control the environment in which service/resource operate [5]. A very good definition of trust defined in [6] as: "Trust is the firm belief in the competence of an entity to act as expected such that this firm belief is not a fixed value associated with the entity but rather it is subject to the entity's behaviour and applies only within a specific context at a given time". This definition captures many important properties of trust as:

(i) Trust is not a fixed value, it is dynamic.
(ii) It is context and time dependent.
(iii) It depends on entity's past and present behaviour.

Another property of trust is that it is not transitive. i.e. if A trusts B and B trusts C then it does not mean that A trusts C. Transitivity may hold if certain conditions are met but in general, trust is not always transitive. Trust can also be affected by those actions that we cannot monitor.

As defined in [6], trust can be classified into two categories: Identity trust and Behavior trust. Identity trust is concerned with verifying the authenticity of an entity to determine its trust level with respect to what the entity is authorized to do and what can be expected from it. Behavior trust, on the other hand, is more real and deals with a wider notion of entity's trustworthiness. A trust model must have the capability to support both, the behavior trust as well as identity trust, depending upon the requirements of the environment in question. The proposed trust model addresses both of these trusts.

The activity of collecting, encoding, analyzing and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding trust relationships is

©2010-11 CIS Journal. All rights reserved.

called Trust Management [3]. Two main approaches, currently available for managing trust are:

(i) Policy based trust management
(ii) Reputation based trust management.

Policy based trust management deals with using policy languages and engines for specifying and reasoning trust. The goal is to determine whether or not an unknown user can be trusted based on a set of credentials and a set of policies. Policy based trust is involved in making access control decisions. It is intended for systems where behavior is guided by complex rules and policies. In reputation based trust, the focus is on trust computation models capable of estimating the degree of trust that can be invested in a certain party based on the history of its past behaviours [3]. It is intended for distributed systems where a system only has a limited view of the information in the whole environment. New trust relationships and trust values are inferred based on the available information. The available information is based on the recommendations and experience of other users.

2. RELATED WORK

Much of the work to enable trust relationships in grid is through the use of X.509 based digital certificates which verify the identity of the requester [3]. Here the trust is assumed to be associated with the identity of the requester but this approach cannot be used to address complex trust related security issues. Another approach to manage trust is by using Trusted Authority (TA) where the basic concept is same as that of a CA (Certificate Authority) but are specifically meant for dealing with trust. But in large scale distributed computing system like grid, its credibility and usage decreases and uncertainty increases as the community of its trustees grows [7].

To address a wider notion of trust, a deeper understanding of interaction among service providers and service requesters is needed which cannot be achieved by identity based trust alone. Trust models such as PolicyMaker [8] and KeyNote [9] are concerned with identity trust. These trust mechanisms do not consider behavior trust which is dynamic and changes with time and thus these approaches have no mechanism to dynamically establish and monitor complex trust relationships. The models that support behavior trust based on experience and reputation are proposed in [7] and [10-12]. These models are really a key inspiration for our work. Compared to these models, the proposed model is less complex. It makes use of simple equations to calculate direct and recommended trust and can be extended to incorporate other attributes like accuracy, honesty, time decay etc. Moreover, the models proposed in [10-12] do not address how to express trust policies and the integration of trust model with the authorization framework.

Other security models for grids like proposed in [13-14] do not deal with trust explicitly and do not provide any trust model to express, validate, update and manage trust relationships. Trust relationships implied from these security models are static and limited. Services like CAS [15] and VOMS [16] address general policy based authorization issues but do not make use of any trust model to express trust policies and to base authorization decisions on it. The specifications like WS-Trust [17] and WS-Federation [18] provide methods for issuing, renewing and validating security tokens and ways to establish access the presence of and broker trust relationships but do not give any comprehensive trust model to determine the trustworthiness of entities and address complex trust related issues. Thus a more comprehensive approach for handling trust, trust relationships and policies is required.

In this paper an attempt has been made to define a general trust model that deals with behavior as well as identity based trust and its integration with the authorization framework. The model provides the mechanisms to express, interpret, evaluate, update and manage trust relationships and policies. In next section, the elements of the trust model will be discussed. Section 4 describes the trust model and algorithms to calculate direct and recommended trust. Implementations details are given in Section 5 with discussion. Finally, Section 6 concludes the paper and also talks about future plans.

3. ELEMENTS OF TRUST MODEL

A Trust Model can be defined as a system that allow service requesters and service providers to assess trustworthiness of each other and state, evaluate and enforce trust requirements and policies among themselves. It is a detailed description of all aspects of a system related to trust. A trust based authorization system grants specific type of access to specific requesters based on their authentication, trust status, what services/resources they are accessing, current state of the system and their conformance to established trust and other security policies. Trust Model should be integrated with the Authorization Framework to provide trust based access to services. This paper presents a Trust Model and its integration with Authorization Framework. In order to understand the architecture well, the entities as defined in [19] are being used:

3.1 Elements of Trust Model

Subject (SU): Subject is an entity that accesses services provided by a service provider in a domain. Subject can be a user, an account, a service or any other entity on its behalf.

Service (SR): Service is a piece of software that provides some functionality and can be accessed by Subjects or other Services. Services are exposed in the environment along with their associated trust requirements and other security policies. Services are provided by different service providers.

©2010-11 CIS Journal. All rights reserved.

Domain (DO): Domain refers to a specific set of Subjects and Services under a unique Domain Policy (DP).
DO = (SU, SR, DP). All entities of a Domain are bound by this Domain Policy. Subjects and Services of a Domain form collaboration. There exist complex trust relationships between Subjects and Services of same/different Domains.

Domain Policy (DP): Domain Policy refers to the set of rules/regulations/requirements of a Domain to which an entity must conform to in order to be in that Domain. Different Domains can have different rules and mechanisms for authentication, authorization and other security requirements.

Service Policy (SrP): Service Policy refers to the set of rules /requirements associated with a particular Service. A Subject must conform to associated Service Policy in order to access that Service.

Service Provider (SP): Service Provider is a physical organization/ institution that expose services/resources in a Domain. Services/Resources in a Domain may come from same or different Service Providers. Thus a Domain is a more dynamic entity than individual physical organization/institution.

Resource (R): Resource is an object that is accessed by Subjects. It can be a CPU, a storage device, software, data, scientific instrument or any other peripheral. Subjects access Resources through Services. In other words, a Resource is a Service. Subjects can access Resources based on their authorization status and their conformance to established security policies.

Trust Policy (TP): Trust Policy refers to the set of trust rules/requirements associated with a Service/Subject. A Subject must conform to Trust Policy associated with the Service in order to access that Service and a Service must also respect the trust requirements of a Subject. Trust Policy (TP) is a subset of overall set of policies (PO).

Access (AC): Access is an operation that a Subject/Service performs on other Service. Access is provided based on conformance to security policies that are associated with that Service.

Policy (PO): Policy is a set of rules/requirements that can be associated with a Subject/Service/Domain. It can be represented as:

PO = (AP, AuP, TP, PP, MP, OP) where
AP is Authentication Policy
AuP is Authorization Policy
TP is Trust Policy
PP is Privacy Policy
MP is Management Policy
OP refers to any Other Policy
Service Policy, Domain Policy and Trust Policy, all are subsets of Policy. i.e. SrP $\subseteq$ PO, DP $\subseteq$ PO and TP $\subseteq$ PO.

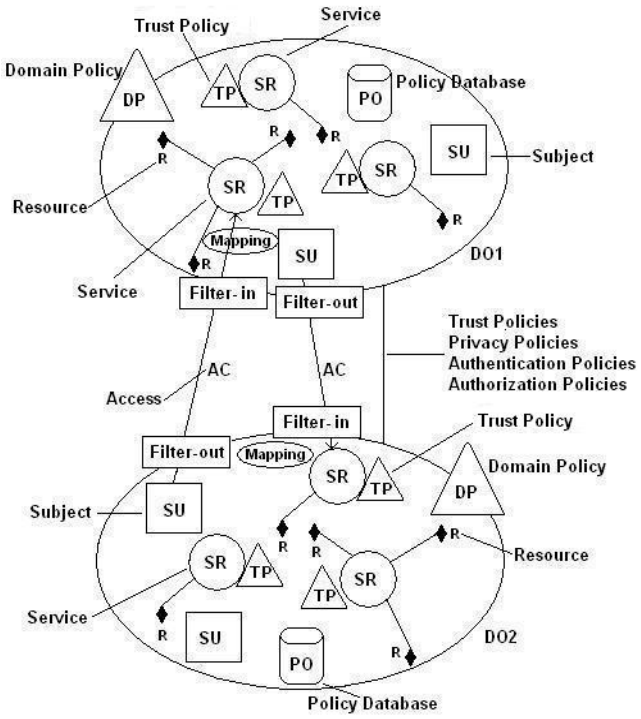

Fig. 1. Grid Environment consisting of two Domains along with other elements of the Model.

Domain Set (DS): Domain Set is simply the set of Domains. i.e. DS = (DO1, DO2, …. , DOn).

Domain Set Policy (DSP): Domain Set Policy refers to the Policy that applies to Domain Set. DSP $\subseteq$ PO.
Filter: The rights/privileges of a Subject are different in different Domains. Filter is a component through which rights/privileges of a Subject are filtered for a particular Domain. There are two types of filters (filter-in and filter-out). These are explained later in this section.

MAP (MAP): is an operation that maps/transforms Subject of one administrative domain to Subject in another administrative domain. E.g. SUi (DOk) → map → SUj (DOl) means that Subject SUi of Domain DOk has been mapped to Subject SUj in Domain DOl.
In a typical Grid Environment, the elements described above interact in a complex manner, which will be discussed in the later sections.

Fig. 1 shows a Grid Environment consisting of two Domains (DO1 and DO2) along with other elements of the framework. In the diagram Squares represent Subjects, Diamonds represent Resources, Triangles represent Policies, Rectangles represent Filters and Ellipses represent Domains. As shown in Fig. 1, Subject's access request for Service SR first passes through Filter-out component at the source Domain and then through Filter-in component at the target Domain. During this passage, Subject's access rights are filtered for the target Domain. With Filter-out component, the Subject leaves the Domain with access rights that his parent Domain grants to him. With Filter-in component, the Subject enters the organization with access rights that the target Domain

©2010-11 CIS Journal. All rights reserved.

grants to the parent Domain. In other words, the Subject gets the intersection of the rights that his parent Domain grants to him and the rights that target Domain grants to parent Domain. This enables the implementation of fine grained access control in the environment. Fig. 1 also shows Policy database to store different types of policies.

At the target domain, the proposed Trust based Authorization Framework checks Subject's conformance to Trust Policy. If Subject conforms to trust and other security policies then mapping operation (MAP) is performed to provide the Subject an identity that is local to that Domain. The mapped identity is then used by the target Domain to provide access to the requested Service/Resource. The mapping operation is optional. If the access of a Service/Resource does not require local identity then it can be omitted. If Subject does not conform to trust and other security policies, the access is denied.

Determining whether a Subject conforms to security policies is a complex task and is performed by the Authorization Framework. Trust Model determines the trustworthiness of the target Domain/Subject/Service and passes the result to Authorization Framework to prepare final result. The proposed Trust Model has been integrated with the Authorization Framework and is called Trust Based Authorization Framework. Next two sections explain the proposed Trust Model and Trust Based Authorization Framework.

4. TRUST MODEL

The trust status of a Service/Resource from a different Domain is hard to determine. Subjects generally have no idea whether the Service is compromised or is malicious [5]. Determining trust relationship is essential not only while accessing Resources/Services but also when enabling delegation [20]. Subjects, Services and Domains can have different trust relationships among each other. E.g. Some Subjects of Domain A may trust Domain B and not Domain C whereas others may trust Domain C and not Domain B, or, Subjects may trust Domain B for Service 1 but not for Service 2, etc. The main problem is how to determine the trustworthiness and establish and manage trust relationships and policies among Subjects and Services of different Domains. Domains need a trust enabled security infrastructure to handle trust related requirements and to provide trust based access to Services/Resources. Establishment of trust may be a one time activity per session or it may be dynamic [14] (requiring the establishment of trust for each request). The trust of an entity with other entity is not a fixed value but can change dynamically depending on the past and present behavior of the entity and context in the environment. Trust should be established from the viewpoint of both the parties (service requesters and Service Providers). Requester's trust with the Service Provider may be different from the Service Provider's trust with the requester. This situation is modeled using two different types of trust: code trust and execution trust, as defined in [5]. Execution trust exist from Subject's side to Service Provider's side that Service Provider will correctly and faithfully allocate resources for the efficient execution of job with respect to established trust and other security policies. Code trust exist from Service Provider's side to Subject's side that Subject will generate a legitimate request consisting of virus free code and will not produce malicious results and does not temper other results/information/code present at Service Provider's end. Except execution and code trust, trust can also be classified into following categories:

Direct Trust: is the trust that a Subject holds on other Service Provider without any intermediate Service Provider or entity.

Full Trust: A Subject is said to have full trust on a Service Provider/Domain if it trust on all the services provided by that Service Provider/Domain.

Partial Trust: A Subject is said to have partial trust on a Service Provider/Domain if it trust on some of the services provided by that Service Provider/Domain.
Recommended Trust: is the trust of one entity on second entity that is recommended by other entities.
Authentication Trust: is the trust of an entity on the authenticity of an identity certificate signed by a certificate authority.

Privacy Trust: is the trust of an entity on the privacy features provided by other entity.
Trust Model is proposed to be a system represented as TS = (EN, TR, OP). Where EN refers to different entities in a grid environment among which trust relationships exist. These entities can be Subjects, Services, Service Providers or Domains etc. TR is the set of trust relationships according to the types of trust discussed above. OP is the set of operators used to handle trust relationships.

Trust Relationship are proposed to be represented as: TR = (SU1, DO1, TT, SR1, DO2, c, t, tv) i.e. Subject SU1 of Domain DO1 trust Service SR1 of Domain DO2 with trust value tv with respect to trust type TT at time t for context c. TT represent the type of trust which can be Authentication Trust, Privacy Trust or Authorization Trust. Since in this paper, a trust based authorization framework is being proposed, the type of trust being dealt with is Authorization Trust.

In order to establish and evaluate trust, every Domain needs to implement a Trust Model integrated with the Authorization Framework. The purpose of the Trust Model is to provide trust based access to Services/Resources. The proposed Trust Model is capable of capturing different types of trust and also provides mechanisms for trust evaluation, recommendations and updation of trust. Fig. 2 shows a high level view of the trust model used in the trust based Authorization Framework which has been implemented as a Trust Manager Service.

As shown in Fig. 2, access request coming from Subject, through Authorization Engine (explained in

©2010-11 CIS Journal. All rights reserved.

Section 5), is first intercepted by Trust Evaluator component. Trust Evaluator is responsible for evaluating trust based on established trust relationships and providing trust evaluation result to Authorization Engine. To evaluate trust, Trust Evaluator makes use of Trust Inference Engine. Trust Inference Engine calculates tv value which is passed to trust evaluator to prepare final result. Trust Evaluator is also responsible for updation of trust among entities. Trust is updated with every interaction that takes place between a Subject and a Service. Trust Policy base fetches established trust relationships and trust management rules from policy database and provide these to trust evaluator for evaluating trust. Trust policies have been expressed in XACML [21].

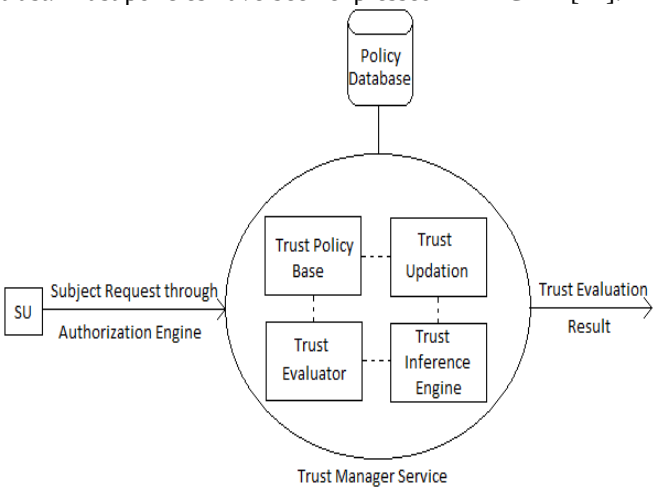

Fig 2. Trust Model Architecture for Trust based Access

Now suppose a Subject from a particular Domain wants to access a Service from another Domain. For this, tv can be calculated as:

$$tv = wdt * dt + wrt * rt \quad (1)$$

where dt is Direct Trust and rt is Recommended Trust. wdt and wrt are the weights assigned to direct and recommended trust and wdt + wrt =1. Let s and f denote the success and failure evidences experienced by the Subject on Service. Then,

$$dt = s / (s + f) \quad (2)$$

rt = Average of
(source's trust on R1 * R1's trust on target,
source's trust on R2 * R2's trust on target,
……..……...
……...……
source's trust on Rn * Rn's trust on target) **(3)**

where R1, R2, …, Rn are recommenders of source Domain. Recommender Domains are trusted by Source Domains to make recommendations about trustworthiness of other Domains. In case of full trust, f can be set to 0 and in case of partial trust, s and f take different values depending on the level of trust. In case of no trust, s can be set to 0. A history of these values is maintained by every Domain. For s and f values, any last n interactions can be considered where n is any number depending on the requirements. A large value of n models accurate behavior and a small value of n models current behavior of one entity with another. For updation of trust, changes are made to s and f. Based on the values of dt and rt, trustworthiness of an entity have been categorized into following levels:

Table 1 Levels of Trustworthiness

Level	dt	rt	tv (with wdt=wrt=0.5)	Meaning
L1	>0.5	>0.5	>0.5 and <=1	Trust (White Zone)
L2	>=0.5	<=0.5	>=0.5 and <=0.75	Can be trusted with some risk (Gray Zone)
L3	<=0.5	>=0.5	>=0.5 and <=0.75	Can be trusted but greater risk (Gray Zone)
L4	<0.5	<0.5	<0.5	Don't Trust (Black Zone)

The above table shows four levels of trust ranging from satisfactory trust (tv > 0.5 and <=1, White Zone) to no trust (tv < 0.5, Black Zone). dt and rt can take a maximum value of 1. With wdt=wrt=0.5, tv can take a maximum value of 1. The more the value of tv is, the higher is the trust. A dt and rt value greater than 0.5 means that the Subject and recommender both have more success evidences than failure. This represents a White Zone and target can be trusted with a satisfactory level of trust. If a Subject experiences more success evidences and recommender experiences failure evidences, or vice versa, then the trustworthiness of target is doubtful. This represents Gray Zone. At this point, a decision should be made whether source Subject wants to give more weightage to recommender trust (rt) or his direct trust (dt) on target. Generally, dt value is given more weightage over rt value. So if a Subject experiences success evidences and recommender experiences failure evidences, then the trustworthiness of target is less doubtful compared to Subject being experiencing failure evidences and recommender experiencing success evidences. If the source Subject and recommender, both have more failure evidences than success evidences, then the target is definitely not to be trusted. This represents a Black Zone. Two important data structures for the calculation of trust value (tv) are:

(i) Trust Table
(ii) Recommender Table

Each Domain maintains a copy of these tables. The format of these tables is:

(i) Trust (sourceDomain, sourceServiceProvider, sourceSubject, targetDomain, targetServiceProvider, targetSubject, success, failure, threshold, context, time)

©2010-11 CIS Journal. All rights reserved.

(ii) Recommender (recommenderDomain, recommenderServiceProvider, recommenderSubject, context, time)

Using above equations and data structures, trust of a Subject on Subject/Service of another Domain can be calculated. Fig. 3 and Fig. 4 describe the algorithms being used to calculate direct and recommended trust. These algorithms make use of the equations and data structures described above.

CalculateTrust (S-DO, S-SP, S-S, T-DO, T-SP, T-S, c)
S-DO: Source Domain
S-SP: Source Service Provider
S-S: Source Subject
T-DO: Target Domain
T-SP: Target Service Provider
T-S: Target Subject
c: context
tv: trust value
dt: direct trust
rt: recommended trust
w_{dt}: direct trust weightage
w_{rt}: recommended trust weightage

i) Set tv = 0, dt = 0, rt = 0, w_{dt} = 0.5, w_{rt} = 0.5.
ii) Select success, failure from Trust table where sourceDomain=S-DO and sourceServiceProvider=S-SP, sourceSubject=S-S and targetDomain=T-DO and targetServiceProvider=T-SP and targetSubject=T-S and context=c
iii) Set dt=success / (success + failure)
iv) Set rt=calRecTrust(S-DO, S-SP, S-S, T-DO, T-SP, T-S, c)
v) Set tv= w_{dt}*dt+w_{rt}*rt
vi) Return tv

Fig 3. Algorithm to calculate trust value

calRecTrust(S-DO, S-SP, S-S, T-DO, T-SP, T-S, c)
S-DO: Source Domain
S-SP: Source Service Provider
S-S: Source Subject
T-DO: Target Domain
T-SP: Target Service Provider
T-S: Target Subject
c: context
nr: number of recommendres
i: loop variable
str: source's trust on recommender
rtt: recommender's trust on target
CR-D: current recommender Domain
CR-SP: current recommender Service Provider
CR-S: current recommender Subject
rt: recommended trust

i) Set rt=0
ii) Set nr = Select count (*) from Recommender Table
iii) for (i =1; i<nr; i++)
a) Set CR-D= Select recommenderDomain[i] from Recommender Table
b) Set CR-SP= Select recommenderServiceProvider[i] from Recommender Table
c) Set CR-S= Select recommenderSubject[i] from Recommender Table
d) Select success, failure from Trust table where sourceDomain=S-DO and sourceServiceProvider=S-SP and sourceSubject=S-S and targetDomain=CR-D and targetServiceProvider=CR-SP and targetSubject=CR-S and context=c
e) Set str = success / (success + failure)
f) Set rtt = getDirTrust(CR-D,CR-SP,CR-S,T-DO,T-SP,T-S)
g) rt = rt + str*rtt
iv) Set rt = rt/nr

Fig 4. Algorithm to calculate recommended trust

Fig. 3 describes the algorithm to calculate trust value. As shown, the direct trust, recommender trust and trust value are initially set to 0. The direct and recommended trust weightage values wdt and wrt are set to 0.5 which means that both direct and recommended trusts are given equal weightage. Then the success and failure evidences experienced by the source Subject with target Domain are fetched from the Trust table and dt is calculated using Eq. (2). To calculate recommended trust, calRecTrust() algorithm is used which is shown in Fig. 4. After determining dt and rt values, trust value tv is calculated using Eq. (1) This is shown in step v of the algorithm. To calculate recommended trust, first, the trust of the source Subject on recommender is calculated and then the recommender's trust on target is called from the recommender. This is shown in steps iii-e and iii-f of calRecTrust() algorithm. The process is repeated for all the recommenders in the Recommender table. After getting the recommendation from all the recommenders, average value of rt is calculated using Eq. (3). This is shown in steps iii-g and iv of calRecTrust() algorithm.

A threshold value of tv can be set depending on the level of trust required. After authenticating Subject, tv value can be calculated and checked to see whether tv is greater than or less that threshold value. If tv>= threshold value then access to Resource/Service is provided otherwise access is denied. After each access of Service/Resource, following updations are made to s and f values:

s = s + 1, if source is satisfied by the Service/Resource access provided by the target, otherwise no change

f = f + 1, if source is not satisfied by the Service/Resource access provided by the target, otherwise no change

Trust Model and its integration with Authorization Framework is explained in the next section.

©2010-11 CIS Journal. All rights reserved.

5. IMPLEMENTATION DETAILS

Implementation has been done in .NET environment with the help of WSE 3.0 toolkit which provides support for several web services security specifications like WS-Security, WS-Trust, WS-SecureConversation etc. Today the world is witnessing the convergence of grid and web services. The security requirements of grid services overlap deeply with the security requirements of web services as grid services are stateful web services. The two (grid and web) started far apart in specifications, technology and applications but now they are converging into a common set of standards and specifications. So the proposed Trust Model has been implemented in terms of web services. Web services security specifications [22] consisting of WS-Security [23], WS-Trust [17], WS-SecureConversation [24] etc. have been submitted to OASIS by IBM, Microsoft and other leading organizations. These specifications addresses security requirements like how to associate security tokens with messages (WS-Security), how to express constraints and requirements of a web service (WS-Policy), how to request and issue security tokens to establish trust (WS-Trust and WS-Federation), how to establish and share security contexts (WS-SecureConversation) etc. These specifications, supported by WSE 3.0, are being used to implement Trust based Authorization Framework.

In order to provide trust based access control, the XACML based Authorization Framework as shown in Fig. 5 has been used. The XACML based access control framework has also been used in GT4 (Globus® Toolkit 4) [25], which is the most commonly and widely used toolkit to construct grid services and applications. In XACML, policy is constructed as a set of rules against the target defined as a triod (Subject, Resource, Action). Fig. 5 also shows other components of the model and depicts how the proposed trust model fits into the authorization framework.

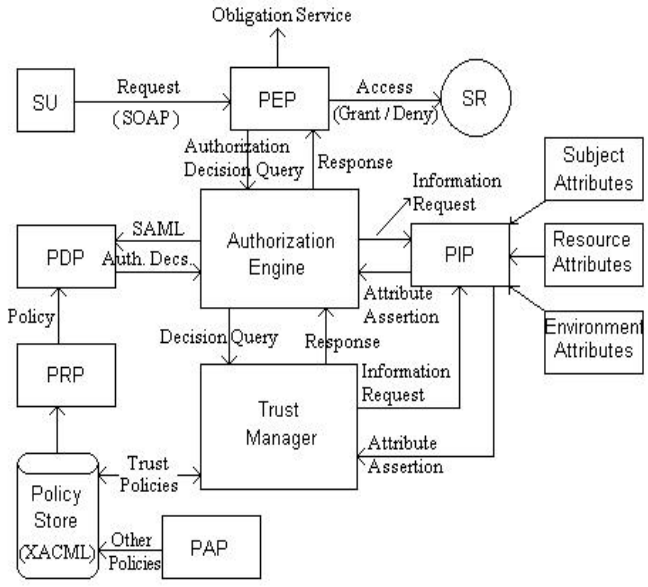

Fig 5. Trust based Authorization Framework

As shown in Fig. 5, authorization request from Subject SU is first intercepted by PEP (Policy Enforcement Point). PEP constructs an authorization decision query and passes it to authorization handler. The result of this query determines if the request is to be granted/deny access to requested Service/Resource. The authorization decision query has details about the identity of the Subject, the Service requested, the purpose for which Service is requested and other details. Authorization Engine passes this information to PDP (Policy Decision Point). The Policy is retrieved by PDP from PRP (Policy Retrieval Point). If the policy information is not available at PRP, it may be retrieved from Policy Store. The Policy Store contains different types of access control policies, expressed in XACML. Trust policies are also stored in Policy Store. The Policies are written by administrator using PAP (Policy Administration Point). PIP (Policy Information Point) is used by Authorization Engine to retrieve attributes of Resources, Subjects and Environment. Trust Manager provides trust based access control information to Authorization Engine. The architecture and working of Trust Manager Service has been explained in Section 4. After receiving this information, Authorization Engine prepares a final result and passes it to PEP. Depending upon the result, PEP either grants the access of requested Service/Resource to the Subject or denies it. After access, the obligation Service, if any, is also executed by PEP.

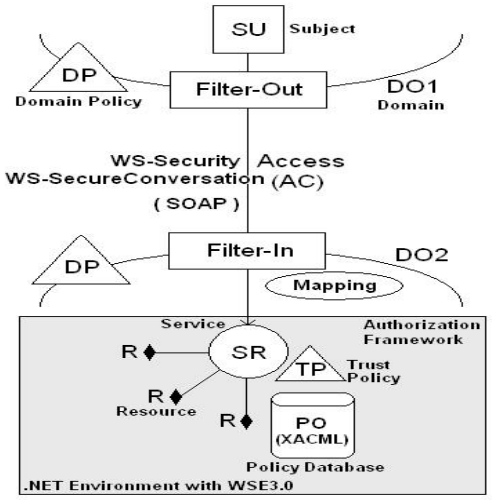

Fig 6. Schematic representing high level view of the implementation. Also showing WS-* and other specifications used

Fig. 6 shows a high level view of the implementation. All the information among communicating parties is exchanged as SOAP messages. WSE 3.0 has been used to construct SOAP messages. WS-Security information is embedded in SOAP message to handle encryption and signature requirements. SOAP messages make use of WS-Security and related specifications for security token exchange and to address security requirements like confidentiality and integrity. The diagram also shows Policy database (PO) to store

©2010-11 CIS Journal. All rights reserved.

trust and other security policies in the XACML format. At the target Domain the SOAP message is processed, credentials are verified and Trust based Authorization Framework (as described in Section 5) is invoked to obtain the trustworthiness of Subject. Subject's conformance to established trust and other security policies is also checked and based on the final result, access to Service/Resource is either granted or denied. During all these steps the relevant information is stored in log tables also to address auditing and accounting requirements.

In the prototype implementation, 10 Domains have been created with users ranging from 10 to 20 in each Domain. All the Domains have more than 5 Service Providers that provide Services/Resources to other Domains. In the database the success and failure incidents, trust value and trust type of a Subject with Subjects/Services of other Domains etc. have been stored. Trust relationship (TR) among different Domains is also stored in the database. Database also contains trust policies established among Subjects and Services of same/different Domains. The XACML skeleton used for the specification of trust policies is shown in Fig. 7. The trust policies are exposed by Subjects/Services in the environment. Other access rules and management policies are also stored in the database in XACML format. The relevant information is fetched and used by Authorization Engine (through Trust Manager Service) to prepare final authorization decision.

```
⋮
<Policy ... PolicyId = "" ... RuleCombiningAlgoId= " " ...   >
      ⋮
      <Target>
      ⋮
      </Target>
      ⋮
      <Rule ... RuleId = "" ... Effect = "" ... >
            ⋮
            <Target>
                  ⋮
                  <Subject>
                  ⋮
                  </Subject>
                  ⋮
                  <Resource>
                  ⋮
                  </Resource>
                  ⋮
                  <Action>
                  ⋮
                  </Action>
                  ⋮
                  <Environment>
                  ⋮
                  </Environment>
                  ⋮
            </Target>
            ⋮
            <Condition>
                  ⋮
            </Condition>
            ⋮
      </Rule>
      ⋮
      <Obligation>
            ⋮
      </Obligation>
      ⋮
</Policy>
⋮
```

Fig 7. XACML Skeleton used for the specifications of trust policies.

For the performance analysis of trust policies, 50 trust related policies have been created. These policies include different aspects related to trust and involve determining trustworthiness of domain, service provider or service. These policies have been attached to the service and the time taken by the PEP component for each trust policy is noted. The average time taken by PEP component in this case comes out to be 207.956 ms. Fig. 8 shows the details.

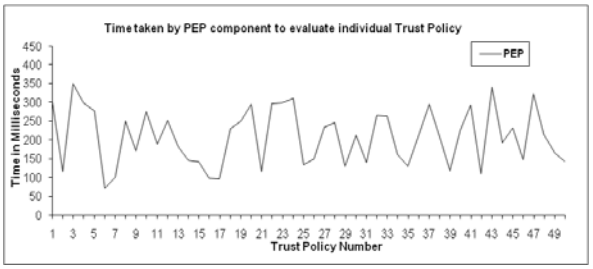

Fig. 8. PEP time to evaluate different trust policies.

The time taken to evaluate different number of trust policies is also noted. For this the number of trust policies attached with the service has been varied from 1 to 50 and the time taken by the PEP component has been noted. Fig. 9 shows the details of results obtained.

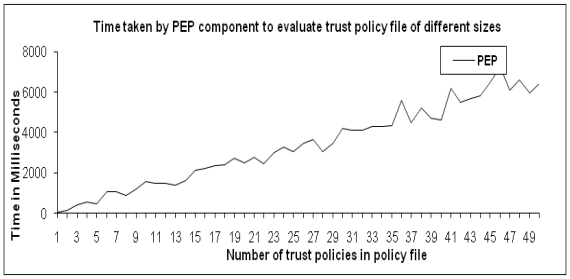

Fig 9. PEP time to evaluate trust policy file of different sizes.

The results show that the prototype implementation is able to meet identified security requirements and trust policies. It is clear from the results that PEP time increases more assertively with the increase in number of trust policies. This is due to the fact that determination of trust value of a target involves calculations and access to history of past interactions taken place between source and target, which is time consuming. Therefore, performance can be a concern if the number of trust policies attached with a service/resource is more. Overall, the results obtained for this implementation suggests that the approach is workable and the proposed framework can be used to provide trust based access to grid services.

6. SUMMARY, CONCLUSION AND FUTURE PLANS

The paper presents a trust model for grid services and describes its implementation with the authorization framework to enable trust based access control. The prototype implementation with a sample scenario has shown that the framework is able to meet the identified trust requirements and trust issues and thus the approach is workable. Currently we have prototype implementation. In future we are planning to use this model in real environment. The process of defining a privacy model and

©2010-11 CIS Journal. All rights reserved.

integrating it with trust based authorization framework is also in the pipeline.

REFERENCES

[1] Foster, C. Kesselman and S. Tuecke, "The Anatomy of Grid: Enabling Scalable Virtual Organizations", *International Journal of Supercomputer Applications*, 2001.

[2] Foster, C. Kesselman, J.M. Nick and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", Open Grid Service Infrastructure WG, Global Grid Forum, 2002.

[3] D. Olmedilla, O. F. Rana, B. Matthews and W. Nejdl, "Security and Trust Issues in Semantic Grids", http://drops.dagstuhl.de/opus/volltexte/2006/408

[4] C. Lin, V. Varadharajan, Y. Wang and V. Pruthi, "Enhancing Grid Security with Trust Management", *Proceedings of IEEE International Conference on Service Computing*, 2004.

[5] M. Humphrey, M. R. Thompson and K. R. Jackson, "Security for Grids", Invited Paper, *Proceedings of IEEE*, 93(3):644-652, March 2005.

[6] F. Azzedin and M. Maheswaran, "Evolving and Managing Trust in Grid Computing Systems", *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering*, 2002.

[7] A. Abdul-Rahman and S. Hailes, "A Distributed Trust Model", 1997, http://citeseer.ist.psu.edu/347518.html

[8] M. Blaze, J. Feigenbaum and J. Lacy, "Decentralized Trust Management",1996, http://citeseer.ist.psu.edu/blaze96decentralized.html

[9] M. Blaze, "Using the KeyNote Trust management System", 1999, http://www.crypto.com/trustmgt/kn.html

[10] A. Abdul-Rahman and S. Hailes, "Supporting Trust in Virtual Communities", 2000, http://citeseer.ist.psu.edu/235466.html

[11] F. Azzedin, M. Maheswaran and A. Mitra, "Trust Brokering and Its Use for Resource Matchmaking in Public-Resource Grids", *Journal of Grid Computing* (2006) 4: 247-263.

[12] F. Azzedin and M. Maheswaran, "Integrating Trust into Grid Resource Management Systems", *Proceedings of International Conference on Parallel Processing (ICPP)*, 2002.

[13] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman and S. Tuecke, "Security for Grid Services", *Proceedings of 12th International Symposium on High Performance Distributed Computing*, 2003.

[14] N. Nagaratnam, P. Janson, I. Foster, et.al., "Security Architecture for Open Grid Services", GGF OGSA Security Workgroup, 2003.

[15] L. Pearlman, V. Welch, I. Foster, C. Kesselman and S. Tuecke, "A Community Authorization Service for Group Collaboration", *Proceedings of IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, 2002.

[16] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell' Agnello, A. Frohner, A. Gianoli, K. Lorentey and F. Spataro, "VOMS, an Authorization Syatem for Virtual Organizations", grid-auth.infn.it/docs/VOMS-Santiago.pdf

[17] S. Anderson, J. Bohren et al., "Web Services Trust Language(WS-Trust)" available at specs.xmlsoap.org/ws/2005/02/trust/WS-Trust.pdf

[18] S. Bajaj, G. Della-Libera et al., "Web Services Federation Language (WS-Federation)", 2003, http://specs.xmlsoap.org/ws/2003/07/secext/WS-Federation.pdf

[19] S. Singh and S. Bawa, "A Privacy, Trust and Policy based Authorization Framework for Services in Distributed Environments", *International Journal of Computer Science*, Vol. 2, No. 1, 2007, pp. 85-92.

[20] M. R. Thompson, D. Olson, R. Cowles, S. Mullen and M. Helm, "CA-based Trust Model for Grid Authentication and Identity Delegation", Grid Certificate Policy WG, http://www.gridcp.es.net/Documents/GGF6/TrustModel-final.pdf, 2002.

[21] XACML Version 2.0, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

[22] "Security in a Web Services World: A Proposed Architecture and Roadmap", Joint Security whitepaper from IBM Corporation and Microsoft Corporation 2002.

[23] B. Atkinson, G. Della-Libera et al., "Web Services Security (WS-Security)" available at www.verisign.com/wss/wss.pdf

[24] S. Anderson, J. Bohren et al., "Web Services Secure Conversation Language(WS-SecureConversation)" available at specs.xmlsoap.org/ws/2005/02/sc/WS-SecureConversation.pdf

[25] B. Lang, I. Foster, F. Siebenlist, R. Ananthakrishnan and T. Freeman, "A Multipolicy Authorization Framework for Grid Security" at www.globus.org