CHAPTER 6
Experiment Evaluations

6.1 Introduction

This chapter performs a comparative analysis of GridPeer Trust with other reputation models based on the security threats. The chapter also describes the methodology used for conducting the testbed experiments and consecutively presents the experimental results themselves, it also provides a discussion based on an interpretation of the experimental results and concludes over the required criteria by which GridPeer Trust is expected to improve resource management in Global Grids.

6.2 Simulation Setup

This section evaluates GridPeerTrust’s performance and shows its effectiveness under different adversarial strategy. The experiments were carried to achieve its accuracy in terms of trust computation is evaluated in the presence of malicious entities. The experiment shows how quickly it adapted to strategically oscillating behavior. The experiments demonstrated the robustness of GridPeerTrust compared to other existing trust models under different security threats. The simulation environment is developed using GridSim toolkit. The general simulation setup includes the environment setting, entity’s behavioral pattern, transaction setting and Output Measurement Index.

6.2.1 Environment Setting

The simulated environment involved N Grid Entities. N is set to 100 in all the experiments. However, in one experiment, the value of N is varied to show the scalability
of the trust model. It was evident from the experiment that the variation in N did not affect the performance of the trust model and as such N was set to 100 for most of the experiments. The Grid entities are of mainly two types- good and malicious. Good Grid entities always cooperate in providing both good service and honest feedback. In contrast, malicious entities are opportunistic in the sense that they cheat whenever it is advantageous for them. Malicious entities provide both ineffective service and false feedback. The percentage of malicious entities in the environment is denoted by the parameter $m_{\text{ind}}$ which is varied in different experiments.

6.2.2 Entity’s Behavioral Pattern

The behavioral pattern of good Grid entities was quite easy to be simulated as they will always provide good service and honest feedback. However, most challenging is to simulate an entity’s malicious behavior realistically. These malicious entities create security threats to the Grid environment, which can only be managed by accurate management of these threats using an efficient trust and reputation model.

6.2.3 Transaction Setting

Two types of transaction setting are simulated namely, random setting and trust prioritized setting. In the random setting, Grid resource consumers randomly interact with Grid resource providers. In the trust prioritized setting a consumer first initiates a transaction request. Against each request certain percentage of Grid resource providers responds. The response percentage is controlled by $m_{\text{rate}}$ parameter. The initiating consumer then sorts the responders based on their trust value and selects the provider with the highest trust value to perform the desired transaction.

6.2.4 Transaction Consistency Rate

To compare the performance of GridPeerTrust with other existing trust models a parameter named, Transaction Consistency Rate (TCR) is used. TCR is described as the
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Environment Setting</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>Number of entities in Grid</td>
<td>100</td>
</tr>
<tr>
<td>m_ind</td>
<td>% of malicious Grid entities</td>
<td>50%</td>
</tr>
<tr>
<td>f_feedback</td>
<td>% of time malicious Grid entity gives false feedback</td>
<td>100%</td>
</tr>
<tr>
<td>m_group</td>
<td>% of malicious Grid entity forming malicious group</td>
<td>0%</td>
</tr>
<tr>
<td>m_spies</td>
<td>% of Grid entity becoming malicious spies</td>
<td>0%</td>
</tr>
<tr>
<td><strong>Trust Computation Setting</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>α</td>
<td>normalized weight factors for the feedback factor</td>
<td>1</td>
</tr>
<tr>
<td>β</td>
<td>normalized weight factors for the trust context factor</td>
<td>1</td>
</tr>
<tr>
<td>FeedBack Decay</td>
<td>To what extent the feedback given by the recommender about a Grid resource provider is important.</td>
<td>0</td>
</tr>
<tr>
<td>Trust Computation Factor</td>
<td>An important factor which include issues like consistency, type of job, type of service provider etc</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6.1 Simulation Parameter Setting
ratio of the number of successful transactions to the total number of transactions. Since computed trust values may range differently for different trust models, other form of evaluation index such as trust computation error is not suitable for comparison.

It really does not matter what range of trust value is assigned to an entity, what matters is how efficiently the model can filter out malicious entities based on the calculated trust value. In other words, the relative ranking of entities based on their trust values is comparable and that’s why only compute TCR for comparison with other models. TCR against the variation of $m_{ind}$, $m_{rate}$, false_feedback and $m_{group}$ is determined. All experimental results are averaged over 50 runs.

Table 6.1, summarizes the different parameters related to the environmental setting and trust computation. The table also lists the default values of the different parameters used. These default values have been empirically tuned.

6.3 Decay of trust with lapse of time

This experiment emphasizes the importance of attenuation of trust with elapse of time in absence of interaction. Since Grid entities are highly dynamic in nature, it is realistic that trust should decay with time. Assume entity $u$ is malicious and it belongs to a community, Where, the most important transactions happen on Friday before the weekends and comparatively less important transactions occur from Monday to Tuesday. Now, entity $u$ can intentionally act well from Monday to Tuesday and builds its reputation with the long term plan of utilizing its reputation during the transactions on Friday to cheat other entities. With no decay metric present such scenario is possible.

To prevent such a case to occur the decay metric is incorporated in Final Trust equation. It is observable that the attenuation function defined, initially has a higher degradation rate than later one. Realistically it should be the reverse i.e., initially the degradation rate should be smaller and as more and more time elapse without interaction the degradation rate should increase. By including a time decay metric it is established the more recent
the transaction the more reliable it is.

\[ \Delta d_{\text{set\_time}} = \begin{cases} 0 & \text{if last successful transaction is within set time} \\ .1 & \text{if there is no transaction is within set time} \\ .5 & \text{if last unsuccessful transaction is within set time} \end{cases} \]

Where

set\_time can be as per Consumer’s requirement

6.4 Trust Computation Accuracy

To compare the performance of GridPeerTrust with other existing trust models the parameter named, Transaction Consistency Rate (TCR) is used. TCR is described as the ratio of the number of successful transactions to the total number of transactions. Since computed trust values may range differently for different trust models, other form of evaluation index such as trust computation error is not suitable for comparison. It really does not matter what range of trust value is assigned to an entity, what matters is how efficiently the model can filter out malicious entities based on the calculated trust value.

In other words, the relative ranking of entities based on their trust values is comparable and that’s why only compute TCR for comparison with other models. TCR determine the variation of (malicious individuals) \( m_{\text{ind}} \) and Malicious Collectives \( m_{\text{group}} \). All experimental results are averaged over 50 runs.

Test Scenario 1. Malicious Individuals

Description

This is the simplest threat that can be found in a trust and reputation system and every trust and reputation model can deal with this kind of attack. Malicious Grid entities
always provide bad services when selected as service providers (Fig. 6.2). The way of preventing such misbehavior is by decreasing the level of trust or reputation of those participants who always provide bad services, categorizing them, therefore, as malicious Grid entity.

**Comparison**

Through Fig 6.3, it is shown that GridPeerTrust shows superiority over the remaining trust models as the amount of malicious entities in the network increase beyond 50%. In this model as a list of type service provider is maintained and a Grid resource provider giving TCR less than 50% is considered a bad service provider so its easy to discard malicious entities even their percentage increases.

Due to the ease of accessibility, networks today are home to a significantly large number of malicious entities, in other words, threats and risks are implicitly increasing as Grid expands. So, in such environment GridPeerTrust would be the best option.
Test Scenario 2. Malicious Collectives

Description

As shown in Fig. 6.4, Malicious Grid entities always provide bad services when selected as service providers. Malicious Grid entities form a malicious collective by assigning the maximum trust value to other malicious Grid entities in the network. Not many trust and reputation models treat the problem arisen from the constitution of collusion among
malicious Grid entities, having thus an important security deficiency. The first thing needed to be able to overcome this threat is to somehow manage, not only the goodness of every user when supplying services, but also their reliability when giving recommendations about other Grid entities. Thus, a user who provides unfair ratings should be also discarded as a service provider.

Fig. 6.4 Malicious Collectives

**Comparison**

In Fig 6.5, it is shown that in GridEigen Trust, forming a malicious collective does not increase the global trust values of malicious entities enough in order for them to have impact on the Grid due to the presence of pre-trusted entities. A user will always have the opportunity to perform a transaction with one of those pre-trusted entities and if an interaction is performed with a malicious entities (which occurs again around 10% of the times), it will be identified as malicious by the whole system where as in GridPeer Trust,
the accurate management of the credibility of an entity as a recommender, as well as the context factor allows this model to effectively overcome this threat.

Fig. 6.5 Comparing GridPeerTrust with other models in terms of TCR against Malicious Collectives (m_group)

In Path Trust, these attacks are actually not handled but avoided by collecting fees for every transaction that are supposed to capture the additional profit gained by the fake transactions, but the more vulnerable a reputation system is to this attack, the higher the fees have to be. A built-in resistance to this attack allows the fees to be lower covering the costs of the transaction rather than being used as a deterrence to create fake
transactions. Due to the experimental randomness, the gradient of the curves may vary from experiment to experiment.

In Fig. 6.5, GridEigenTrust and Path Trust have negative gradient so in their case TCR is actually decreasing as collusive group size is increasing where as in case of GridPeer Trust, the feedback credibility measure filters out false feedbacks. Here false high ratings come from agents with low feedback credibility as a result they have no impact on TCR. In order to attain a high credibility malicious entity would have to provide honest feedback which goes against their true nature.

**Test Scenario 3. Malicious Collectives with camouflage**

**Description**

As shown in Fig. 6.6, malicious Grid entities provide bad services in p% of all cases when selected as service providers. Malicious Grid entities form a malicious collective by assigning the maximum trust value to other malicious peers in the network. In many cases, this threat is not always easy to tackle, since its resilience will mostly depend on the behavioral pattern followed by malicious entities.

Furthermore, the variable behavior is not even considered as a threat in many models in the sense that they do not punish that kind of behavior, but they just try to adjust the trust and reputation given to an entity to its real and current goodness. There needs to be process to somehow distinguish the confidence deposited in an entity as a recommender and the trust deposited in the same entity as a service provider. This mechanism is quite helpful when trying to avoid unfair ratings from malicious entities. Additionally, the variable behavior of an entity, when detected, could be punished and avoided.

**Comparison**

Malicious Grid entities provide bad services in p% of all cases when selected as service
providers. Malicious entities form a malicious collective by assigning the maximum trust value to other malicious entities in the network.

Fig 6.6 Malicious Collectives with Camouflage

Both GridEigenTrust and GridPeerTrust are able to handle this security threat due to the decay function that discounts the old feedback of entities thereby obtaining good outcomes where as in path trust as there is no decay function so the result is generally increasing and decreasing due to the fact that these attacks are actually not handled but avoided by collecting fees for every transaction that are supposed to capture the additional profit gained by the fake transactions, but the more vulnerable a reputation system is to this attack, the higher the fees have to be.

The TCR increases when these malicious entities behave in good manner and slowly decreases when entities start misbehaving. In this experiment it’s analyzed that the impact of malicious response on TCR. As seen in Fig. 6.7 that the malicious entities tend to fool other entities by oscillating between good and malicious nature. In this experiment
two scenarios were tested with malicious per set to 40% and 60% respectively while collusion is set to 0% in both the cases.

Fig. 6.7 represents the computed TCR against malicious collective camouflage. From the figures it’s analyzed that GridPeerTrust out performs all other trust models significantly. This is because our model keeps track of sudden rise and fall of trust by agents and penalizes any entities showing frequent trust fluctuations. While other models fail to identify the strategic alternations made by malicious entities, GridPeer Trust model quickly distinguishes such alternations through our deviation reliability metric.

![Graph comparing GridPeerTrust with other models](image)

**Fig. 6.7** Comparing GridPeerTrust with other models in terms of TCR against Malicious Collectives (m_group) with Camouflage

**Test Scenario 4: Malicious Spies**

**Description**

In fig 6.8, it is shown that some malicious Grid entities always provide bad services when
selected as service providers. Those malicious Grid entities form a malicious collective by assigning the maximum trust value to other malicious Grid entities in the network. Other distinct malicious Grid entities, known as malicious spies, always provide good services when selected as service providers, but they also give the maximum rating values to those malicious Grid entities that always provide bad services. In this threat, the malicious spies may gain a high level of trust and reputation, since they always provide good services, being able then to easily subvert the trust and reputation mechanism applied in the system.

An accurate management of the reliability of the Grid entities, not only as service providers, but also as recommendation providers effectively help to prevent this kind of abuse, although it will probably take longer (more effort and more resources needed, therefore) in order to be able to identify both the malicious Grid entities and the malicious spies.

![Diagram](image)

Fig. 6.8 Malicious spies

**Comparison**

Some malicious entities always provide bad services when selected as service providers.
Those malicious entities form a malicious collective by assigning the maximum trust value to other malicious entities in the network. Other distinct malicious entities, known as malicious spies, always provide good services when selected as service providers, but they also give the maximum rating values to those malicious peers who always provide bad services.

In this threat, the malicious spies may gain a high level of trust and reputation, since they always provide good services, being able then to easily subvert the trust and reputation mechanism applied in the system.

Most of the times, this kind of attack have not a trivial or easy way of being effectively tackled. GridPeerTrust has the ability to deal with this threat due to, the definition of
credibility in terms of the similarity between two entities, which allows the model to accurately detect and identify in the malicious service providers as well as malicious recommenders where as there is no check for the credibility of recommender in GridEigenTrust and Path Trust.

**Test Scenario 5. Sybil Attack**

**Description**

An adversary initiates a disproportionate number of malicious entities in the network. Each time one of the entities is selected as a service provider, it provides a bad service, after which it is disconnected and replaced with a new entity identity. This kind of attack might prove quite problematic since it could prevent good entities from being able to gain a good reputation, since they might not be selected most of the times. Again, not many trust and reputation models deal with such an important and potentially dangerous threat like the Sybil attack leading thus to an underestimated but great risk.

One of the most common solutions proposed in the literature for this kind of threat consists of associating a cost to the generation of new identities in the community. This cost is not necessarily economic, but it can also be a cost in terms of time or resources, for instance. Another suggested way of dealing with this problem makes use of a central entity managing (virtual) identities in the system, or even a set of identity providers ensuring that every participant in the community has a unique and immutable identity.

**Comparison**

This type of threat is quite difficult to be handled so it is avoided by all three (GridPeerTrust, GridEigenTrust, PathTrust) by imposing a cost to the generation of new identities like Member Registration Fees. Every time a malicious entity enters a Grid to become a Grid resource provider certain membership cost is imposed on it and it will help to reduce the threat to certain extent.
Test Scenario 6: Man in the middle attack

Description

As shown in Fig 6.10 a malicious entity can intercept the messages from a benevolent service provider entity to the requestor and rewrite them with bad services, making therefore the reputation of the benevolent entity to decrease. That participant could even maliciously modify the recommendations given by an honest entity, in order to benefit his/her own interests. One more time, this is a threat which has not been associated with trust and reputation systems traditionally.

Assume the authenticity of the entity providing either a service or a recommendation. This attack can cause a great damage and effect in the system if its application is possible. A simple way of avoiding this risk could be by the use of cryptography schemes in order to authenticate each user in the system (maybe with a digital signature or any similar mechanism).

![Fig. 6.10 Man in Middle Attack](image)

Comparison

However, and unfortunately, it is not always feasible to apply a solution, above all in highly distributed environments like Grid environment. This type of threat is tackled by both GridPeerTrust and GridEigenTrust by the use of cryptography schemes in order to
authenticate each user in the system where as Path Trust lack this scheme. In GridPeer Trust model this scenario is handled using Secured GridPeerTrust Algorithm.

**Test Scenario 7: Driving down the reputation of a reliable entity**

**Description**

Malicious entities always provide bad services when selected as service providers. Malicious entities form a malicious collective by assigning the maximum trust value to other malicious entities in the network. Additionally, they give the worst rating to those benevolent entities, which indeed provide good services. This kind of attack can be even worse than the ones named malicious collectives and malicious spies, since in this case benevolent entities also receive unfair critics from malicious entities.

![Fig. 6.11 Driving Down the reputation](image)

In such a situation if an interaction with a malicious entity is carried out it can be identified as malicious, but if an interaction has never been performed with an entity which is actually benevolent but whose reputation has been driven down by malicious participants, then that entity will not probably be chosen as the entity to have an
interaction with. The differentiated management of the trust given to a participant when supplying services and the reliability of his/her recommendations can be very useful in this scenario as well.

However, there are some trust and reputation models where this distinction is not explicitly done but, due to their dependency on the topology of the network, are able to find the most trustworthy path leading to the most reputable entity offering a certain service.

**Comparison**

GridPeerTrust has the ability to deal with this threat due to its capability to rate the credibility of recommenders and Grid resource providers and this threat has less impact on GridEigenTrust due to the way the trust is calculated as institution is trust is taken into account so this threat has meager effect on the trust value of a reliable entity where as this threat has a high impact on Path Trust.

![Comparison](image)

Fig. 6.12 Comparing GridPeerTrust with other models in terms of TCR against Driving Down the Reputation
Test Scenario 8. Partially Malicious Collectives

Description

Malicious entities always provide bad services when selected as certain service providers. However, they always provide good services when selected as other different service providers. That is, for certain services they behave properly, while for other specific services, they act maliciously. Malicious entities form a malicious collective by assigning the maximum trust value to other malicious peers in the network. There are some trust and reputation models which are not resilient to this kind of attack since they just perform a global computation of the trust and/or reputation of an entity, regardless the service they are providing.

![Fig 6.13 Partially Malicious collectives](image)

In such a situation some distortion can emerge, considering an entity as fully or quite benevolent (malicious) although it can also provide some fraudulent (good) services. By
just considering a different score for every service offered by an entity, this threat is mitigated most of the times.

However, it is not always possible to make this distinction since in some environments (for instance, those with a great amount of services offered) it could lead to some scalability problems.

**Comparison**

GridPeerTrust can also overcome the threats of partially mischievous collectives since it introduces a trust context factor to measure the importance of each transaction where as GridEigenTrust and Path Trust handle this threat to some extent by considering a different score for every service offered by an entity.

![Fig. 6.14 Comparing GridPeerTrust with other models in terms of TCR against Partially Malicious Collective](image)

In GridPeerTrust, following are some of the issues included in Trust context Factor:

a) Consistency = No of Successful Transaction / Total Number of Transactions
b) Type of job = Computing/Database/Printing/Misc

c) Non-repudiation = No of Incomplete Transaction / Total Number of Transactions

d) Size of job = Large/Medium/Small

e) Type of Service Provider = Excellent/Good/Average/Poor

**Test Scenario 9. Malicious Pre-trusted Entity**

**Description**

Some or all the pre-trusted benevolent entities become malicious ones, may be by always providing bad services when selected as service providers or by rating with maximum trust value other malicious entities who always provide bad services when selected as service providers. It is worth mentioning that it is not always feasible to find a set of peers that can be trusted before any transaction is carried out in the system.

![Fig. 6.15 Malicious Pre-trusted Entities](image-url)
Comparison

This type of threat is an algorithmic specific as only applicable on GridEigenTrust as the base of their strategy on this kind of participants. However, and maybe in a paranoic way of thinking, every user in a virtual community can behave inappropriately at some point. If such a thing occurred with a pre-trusted entity, GridEigenTrust model is in a risk.

6.5 Comparative Analysis

1. Design Analysis

In GridEigen Trust and Path Trust, when a new entity joins the Grid as resource provider, its assigns a fixed low trust value, whereas in GridPeer Trust, the initial trust is evaluated based on the satisfaction criteria. Path Trust does not accounts global trust whereas GridEigen Trust and GridPeer Trust both take care about global trust. The control of Grid entities in both GridEigen Trust and Path Trust is centralized whereas in GridPeer Trust it’s decentralized.

<table>
<thead>
<tr>
<th>System</th>
<th>GridEigen Trust</th>
<th>Path Trust</th>
<th>GridPeer Trust</th>
<th>NICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized Data</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Trust Metric</td>
<td>[0,1]</td>
<td>[0,1]</td>
<td>[0,1]</td>
<td>[0,1]</td>
</tr>
<tr>
<td>Trust Aggregation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Type of Feedback</td>
<td>Continuous</td>
<td>Negative &amp; Positive</td>
<td>Continuous</td>
<td>Continuous</td>
</tr>
<tr>
<td>Storage Cost</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Scalability</td>
<td>Not Applicable</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

Table 6.2 Design Analysis
2. Security Analysis

The Table 6.3 summarizes the security analysis w.r.t the security threats.

<table>
<thead>
<tr>
<th>Security Threats</th>
<th>GRID PEER TRUST</th>
<th>GRID EIGEN TRUST</th>
<th>PATH TRUST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malicious individual Grid entity</td>
<td>Handled</td>
<td>Handled</td>
<td>Handled</td>
</tr>
<tr>
<td>Malicious collectives</td>
<td>Handled</td>
<td>Handled</td>
<td>Not Handled</td>
</tr>
<tr>
<td>Malicious collectives with camouflage</td>
<td>Handled</td>
<td>Handled</td>
<td>Not Handled</td>
</tr>
<tr>
<td>Driving down the reputation of a reliable entity</td>
<td>Handled</td>
<td>Handled</td>
<td>Handled</td>
</tr>
<tr>
<td>Malicious spies</td>
<td>Handled</td>
<td>Not Handled</td>
<td>Not Handled</td>
</tr>
<tr>
<td>Sybil Attack</td>
<td>Handled</td>
<td>Handled</td>
<td>Handled</td>
</tr>
<tr>
<td>Man in the middle attack</td>
<td>Handled</td>
<td>Handled</td>
<td>Handled</td>
</tr>
<tr>
<td>Partially malicious collectives</td>
<td>Handled</td>
<td>Handled</td>
<td>Not Handled</td>
</tr>
<tr>
<td>Malicious pre-trusted entity</td>
<td>Not Applicable</td>
<td>Not Handled</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>

Table 6.3 Security Analysis

1. PathTrust: - It is one of the first attempts to apply reputation methods to Grids by approaching VO management phases. They approached only partner selection and did not tackled organizational aspects. Their model still lacks of dynamics, as the feedback is collected only at the dissolution of the VO. But, the advance in the field is given by the fact that ideas from previous research were successfully transferred in the area of VOs and Grids. When a set of individual malicious entities are present in the system, those entities receive positive rating only from other malicious entity, since they are the only ones who value the supply of malicious services. And even that only occasionally, since malicious entities have to meet each other through an interaction. Because of their low ratings, malicious entities are rarely chosen as service providers.
Forming a malicious collection and malicious collection with camouflage increases the positive rating of malicious entities enough in order for them to have impact on the network due. Finally, to deal with the problem of Sybil attack some kind of cost to the generation of new identities can be imposed.

2. Grid Eigen Trust: - When a set of individual malicious entities is present in the system, those entities receive high local trust values only from other malicious entity, since they are the only ones who value the supply of malicious services. And even that only occasionally, since malicious entities have to meet each other through an interaction. Because of their low trust values, malicious entities are rarely chosen as service providers.

Forming a malicious collective does not increase the global trust values of malicious entities enough in order for them to have impact on the network due to the presence of pre-trusted entity. A user will always have the opportunity to perform a transaction with one of those pre-trusted entities and if an interaction is performed with a malicious entity, it will be identified as malicious by the whole system. By imposing some kind of cost to the generation of new identities, but they also show the vulnerability of their model against malicious spies, since their opinions and recommendations will be taken into account (even when rating malicious entity) due to their proper behavior when supplying services.

3. Grid Peer Trust: - The accurate management of the credibility of an entity as a recommender, as well as the context factor allows Grid Peer Trust model to effectively overcome many of the security threats. Malicious individual entities, malicious collectives, malicious collectives with camouflage and Driving down the reputations of a reliable entity are some of the threats that are solved by Grid Peer Trust. This ability to deal with those threats is due to, among other factors, the definition of credibility in terms of the similarity between two entities, which allows the model to accurately detect and identify in the malicious service providers as well as malicious recommenders. Additionally it stimulates the community to supply recommendations by building incentives or rewards to those entities that provide feedbacks to others. Finally, Grid Peer
Trust can also overcome the threats of partially malicious collectives (since it introduces a context factor to measure the importance of each transaction) and the man in the middle attack. The latter is tackled making use of cryptographic mechanisms. Specifically, the feedback can be provided in the encrypted form. By imposing some kind of cost to the generation of new identities to become part of any VO, Sybil attacks are avoided.

3. Models fulfillment of pre-standardization recommendations

Table 6.4 summarizes comparison based on the fulfillment of the pre-standardization recommendations by some trust and/or reputation models in distributed systems.

<table>
<thead>
<tr>
<th>Factors</th>
<th>GridPeer Trust</th>
<th>GridEigen Trust</th>
<th>Path Trust</th>
<th>NICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing Behavior of Grid Entity</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Trust Decay Function</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trust Metrics</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Recommendation Credibility</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trust Negotiation</td>
<td>✔</td>
<td>✗</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Awards &amp; Penalty &amp; Pre-trusted Grid Entities</td>
<td>✔</td>
<td>✔</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Trust Context Factor</td>
<td>✗</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.4 Summarization of recommendations w.r.t various models
6.6 Conclusion

This chapter described a comparative analysis among GridPeerTrust and other reputation models (GridEigenTrust, PathTrust) based on the security threats and consequently assess their performance through a series of testbed experiments based on security threats. These activities served the purpose of revealing the circumstances by which GridPeerTrust outperformed other models and consequently constituted as an empirical proof for justifying the reputation-based trust model.