TUX-TMS: Thapar University Extensible-Trust Management System

Shashi shashi@thapar.edu
Centre of Excellence in Grid Computing
Computer Science and Engineering Department
Thapar University
Patiala-147004, INDIA

Seema Bawa seema@thapar.edu
Centre of Excellence in Grid Computing
Computer Science and Engineering Department
Thapar University
Patiala-147004, INDIA

Abstract

In a Grid Computing scenario, where the market players are dynamic; traditional assumptions for establishing and evaluating trust, do not hold good anymore. There are two different methods for handling access controls to the resources in grids: first by using policy based approach; where logical rules and verifiable properties are encoded in signed credentials and second by using reputation based approach; where trust values are collected, aggregated and evaluated to disseminate reputation among the market players. There is a need for dynamic and flexible general-purpose trust management system. In this paper TUX-TMS: an extensible reputation based Trust Management System is presented for establishing and evaluating trust in grid systems.

Keywords: Trust, Trust Management System, Grid Computing, Reputation, Feedback

1. INTRODUCTION

Distributed Environments are touching new heights, becoming more useful, popular and more complex with the emergence of service oriented architecture and computing technologies like peer-to-peer, autonomic, pervasive and grid etc [21]. Grid Computing has evolved into a major computing paradigm, having increased focus on secured resource sharing, manageability and high performance. Grids are distributed computing platforms which are heterogeneous and dynamic in nature. The original vision of Grid computing aimed at having a single global infrastructure and providing users with computing power on demand [19]. Efforts to address this issue include providing interoperability among different Grids and Grid middleware [20], and creating trust federations between Grids to grant users in one Grid easy access to another. Grid systems involve the risk of executing transactions without prior experience and knowledge about each other’s reputation. Recognizing the importance of trust in such environments, there is a necessity of designing strategies and mechanisms to establish trust. Reputation systems [1] provide a way for building trust through social control by utilizing community based feedback about past experiences of domain to help in making recommendation and judgment on quality and reliability of the transactions. In this paper, we propose TUX-TMS (Thapar University Extensible- Trust Management System) for establishing trust in Grid Environments. The proposed

Trust system evaluates trustworthiness of the transacting domain on the basis of number of past transactions, feedback ratings and recommendations.

The remaining paper is organized as follows: The taxonomy, parameters and trust metrics are discussed in section 2, 3 and 4, respectively. In section 5, we have discussed our proposed Trust model. Section 6 highlights the implementation strategies adopted to illustrate how our model evolves and manages trust. Results are briefly discussed in Section 7. Related work in covered in Section 8. Section 9 concludes our work.

2. TAXONOMY FOR TUX-TMS

We have considered following taxonomy for our TUX-TMS:

- **Service Requestor**: A service requestor is an entity or domain, which requests some services such as computing, storage etc. from the service providers.
- **Service Provider**: A domain which provides some services to the user or an entity requesting some service.
- **Services**: Services such as computational, data storage, printing, using software licenses or scientific instruments which may be provided to the service requestor.
- **Transaction**: When a service requestor uses the services of the service providers and pays the amount for the services used and submits feedback for the same.
- **Trust Context Factor**: At any time t, the purpose of using services of a service provider by a service requestor can be termed as the trust context factor Ω, which may vary from application purpose. The trust weights may be assigned and evaluated likewise.
- **Service consumed**: A service is said to be consumed when a transaction has completed.
- **Service Initiated**: Services are initiated when jobs are mapped onto resources.

3. TRUST PARAMETERS

We have considered following trust parameters for our TUX-TMS:

3.1.1. Trust

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's behavior and applies only within a specific context at a given time [11].

3.1.2 Reputation

Reputation can be taken as a means of building trust, as one can trust another based on a good reputation. Therefore, reputation can be a measure of trustworthiness, in the sense of reliability. We will allow reputation to be assessed from the recommendation score and the trustworthiness of the domain. The recommendation can be submitted as highly recommended or not recommended on the context of service used.

3.1.3 Trust asymmetry

If an entity X trusts another entity Y, then Y should also trust X is not necessarily, yes. This situation can be termed as trust asymmetry, the solution to which is trust symmetry [12]. Here, the user needs to position itself as the resource provider host(s) to estimate their trusts on the user from user's point of view, i.e. to evaluate the trust reflection.

3.1.4 Aggregation of past behavior

A domain stores the values of past transactions so as to determine and estimate the trustworthiness of the domain for future transaction purposes.

3.1.5 Trust Level

Trust can be categorized into various levels ranging from very low trust level to extremely high trust level. Whereas, trust level *et* i.e extremely high trustworthy is not provided by any existing trust relationship. The trust level of a domain varies from not trustworthy to extremely high trustworthy.

3.1.6 Trust Inheritance

In a dynamic grid environment, market players are allowed to join or leave as per the policies agreed upon. When an entity joins a domain, it inherits the recommendation trust table [9] of the domain. There is a member weight associated with every entity to indicate if the entity is a new or an old member with its domain and it is up to the individual domain to decide what constitutes an entity to fall in one of these member weights.

3.1.7 Identity Trust

Identity trust is concerned with verifying the authenticity of an entity and determining the authorizations that the entity is entitled to access and is based on techniques including encryption, data biding, digital signatures, authentication protocols and access control methods [9].

3.1.8 Behavior Trust

Behavior trust is concerned with monitoring and managing of the entity's behavior which is accumulated and evaluated with a period of time, an entity or domain is involved with transactions.

3.1.9 Evolving trust as a newcomer

As a newcomer, the trust values are empty and have to evolve over a period of time after a number of transactions.

3.1.10 Trust Threshold

It is taken as a minimum value required depending on the sensitivity of the application, service requested or provided to establish trust relationship with any entity.

3.1.11 False recommendation

An entity or domain can submit malicious or fraudulent recommendation about an entity after a number of transactions due to business competition, enmity or to degrade the reputation of an entity. Therefore, the recommendation values are aggregated for an entity to ensure reliable and trustworthy services.

3.1.12 Trust and Reputation Decay

The value of trust decays with time as grid environments are dynamic in nature and market players are allowed to volunteer or withdraw as per the policies agreed upon. Therefore, the trust value decays, which enforces the domain to re-establish the value when it participates in the grid.

3.1.13 Feedback

Feedback is a taken as a value p such that $0\leq p\leq 1$, which can be issued by the service requestor about the quality of a service provided by a service provider in a single transaction and vice versa.

3.1.14 Trustworthiness

An entity's trustworthiness is an indicator of the quality of the entity's services over a period of time. It is often used to predict the future behavior of the entity. Intuitively, if an entity is trustworthy, it is likely that the entity will provide good services in future transactions.

3.1.15 Trust relationships

Determining trust relationship is essential while accessing resources/services. We have assumed the three Trust Relationships for our system: Direct Trust, Indirect Trust and Recommended Trust.

3.1.16 Intrusion Detection and Audit Trails

Intrusion Detection is a process of monitoring the events occurring in the system or network and analyzing them for signs of possible violations. Of the security policies agreed upon such that a garbage collector will clear the unused data upon completion of task or job etc. Audit trails are responsible for establishing accountability of users for their actions and provide evidence, if any.

3.1.17 Trust and Reputation Update

After the transaction, intrusion detection and audit trail has taken place the feedback is updated in the databases for trustworthiness and recommendation.

3.1.18 Risk Assessment

The service quality provided over the service quality expected leads to calculation and assessing the risk involved in the transaction**.**

3.1.19 Interoperability

The trust model should be able to interoperate at various levels such as protocol, policy and identity level.

4. TRUST METRICS IN TUX-TMS

A trust metric is a measure of how an entity of a domain is trusted by the other entities. TUX-TMS is a transaction based feedback system, where the feedback is mandatory with each transaction. In TUX-TMS, the trust metrics derived are:

- The trustworthiness is an indicator of the quality of the entity's or domain services. The higher the value, the higher the trustworthiness of the domain.
- The feedback scores that a domain or entity receives from other domains in terms of service provided. This value is average of the feedback score over a number of transactions. As the number of transaction increases, the composite feedback score decreases. We have assumed various trust relationships between domains: indirect, recommendation and direct trust relationship.
- The trust and reputation decays as the time progresses and the trust needs to be established again between the domains after a period of time has lapsed.
- Risk assessment establishes the risk level involved in transacting with a domain.

5. THE PROPOSED TRUST MANAGEMENT SYSTEM

The main focus of this paper is on design and development of TUX-TMS-an extensible Reputation based Trust Management System for establishing trust and assessing trustworthiness of domains in grids environments. Reputation based systems rely on feedback score to evaluate trustworthiness of a domain [15].

5.1 Components of TUX-TMS

Various components have been used in our TUX-TMS.

I. **Identity Management System:** An identity management system maintains repository of the user credentials and interacts with various Trusted Third Parties (TTP) to check for validation of the credentials provided by the user. The Identity Management System incorporates authentication, authorization, confidentiality, log related and trust related functions

II. **Trustworthiness:** The trustworthiness is evaluated and calculated from past transaction scores and the feedback provided for the service provided by both service requestor and service provider.

III. **Reputation Engine:** The reputation engine provides the recommended score of an entity or domain.

IV. **Trust Inheritance**: The entity which is a part of a domain inherits some value Δ€ from the domains depending upon the credibility of the domain. For a malicious domain, the value is very low.

V. **Risk Assessment**: The value depends on the rank of the domain in the DET. Higher the rank, minimal is the risk involved in transacting with the domain.

VI. **Trust and Reputation Decay**: The trust and reputation decays on the basis of the time an entity or domain is not transacting. The trust decay factor() decays and the trust and reputation values are to be regained again by the domain for future use purposes.

VII. **Trust Inference Engine**: A trust inference engine takes the value of Trustworthiness, Reputation, Risk Assessment and trust inheritance for calculating the threshold value to allow a domain to transact.

VIII. **IDS and Audit trail**: If a TA (trusted agreement) is violated, the transaction performed is rejected and the feedback scores are not updated in the databases.

5.2 Architecture of TUX- TMS

The architecture of the TUX-TMS is depicted in Figure1. An end user i.e. service requestor/service provider is requested to login into the TUX-TMS using his credentials. If the user is already registered with the TUX-TMS, the Identity Management System checks the authentication and verifies the information provided such as security certificates or else if a user is new, he is requested to register. First, the Trust Inheritance, Risk Assessment and the values of Trustworthiness (TD) and Recommendation (RD) are checked and then a user is allowed a set M [D, Sp, S] where D stands for Domain, Sp for service Provider and S stands for the service a user is allowed to perform. The user can further calculate and check the values of trust of the Service Provider by first checking the trust values, recommended value and further calculate the risk involved in transacting. If the user is satisfied with the values, he can perform the transaction. After the transaction, a user is requested to fill in the Feedback of the trustworthiness and recommendation based on questionnaire. After a service requestor, the service provider is requested to fill up the feedback form. Thereby, making the transaction complete. An Intrusion Detection System and audit trails checks the information provided to be from the intended players and the trust values and recommendation values are updated in their respective databases. If the values are reported from the malicious origin, the values are discarded and the malicious domains are blacklisted. TUX-TMS DB is responsible for maintaining the database.

6. IMPLEMENTATION STRATEGIES

We have performed initial experiments to evaluate feasibility and benefit of our TUX-TMS.

We have taken 8 departments, 7 Schools and five hostels for experimental purpose which has approx. 100 systems each and provides services such as computation, printing and data storage. The database details have been stored in TUX-TMS DB. The tables ETT (Entity Evaluation table) has been designed using following attributes: Service Requestor (SR), Service Provider (SP), Number of transactions (N_T), Feedback Score (F_s), Recommendation Score (R_c), Trust Relationship (TR), Total Entity Value (TEL), Trust Context Factor (Ω). There are other repositories Domain Evaluation Table (Domain Evaluation Table), Service Provider Data (SPD) and Temporary Storage Table (TST).

In TUX-TMS, a domain's trustworthiness is defined as the degree of trust other domains can have on one domain to initiate communication or participate in any kind of transaction. The trustworthiness can be computed using trust decay factor T*decay* with the feedback score.

$$Tdecay = \sum Tw / \sum Nt \qquad (1)$$

$$Rdecay = \sum Rc / \sum Dn \qquad (2)$$

$$Tfin(D) = \alpha * Tr + \beta * Rc + \gamma \qquad (3)$$

$$Ra = Rank / Tdn \qquad (4)$$

$$TI = Tfin - Ra * 100 \qquad (5)$$

Here, Tw: Trustworthiness
T_{decay}: Trust Decay Factor
N_T: Number of transactions

D: Domain
R_{decay}=Reputation Decay
Dn: Total number of domains with whom transaction has taken place
T_{fin}=Total trust value
T_{dn}: Total number of domains in the system
Ra=Risk Assessment
R_{ank}=Rank of a domain
TI=Trust Inheritance
α, β, γ=Constants used for belief, disbelief and plausibility

Algorithm 1. *TUX_TMS(ETT, DET, TST,SR_ID, SPD, domain)*

```
Input: ETT, DET, TST,SR_ID, domain    Output: ETT, DET, TST
If authenticated then
   If authorized then
       Resource_request
      TW <= Trust_Inference_Engine(DET, domain, r, sr, rp, R, P)
       Resource_requested <= select k from TST_ID
       If trust_requirement of ID=Resource_requested from SPD<TW  then
          Notify_ID about request
          If  ID approves then
              T_id <= Transaction ID
              Transaction occurs
             Transaction ends
         End if
       End if
    If enter_feedback then
       Valid_tid <= Verify_Transaction(t_id)
       If Valid_tid = true then
           Valid_user <=Perform_IDS( SR_ID)
           If Valid_user=true then
              Add_feedback(TEL_sr, TEL_sp, R, P)
          End if
      End if
    End if
  End if
 Else register
End if
```

Here, the Feedback Score can be calculated by giving a value p, such that 0≤ p≤1 score may be provided for both the Service Requestor and the Service Provider. The service is rated on the basis of trust and reliability only.

Considering the malicious intent of some entities who would try to increase the rating of an entity by giving more score for the services provided, the overall score will reduce as the number of transactions will increase, by calculating the score on the basis of equation 3 for the domains transacting.

$$Fs(D) = \sum Fs / \sum Nt \qquad (6)$$

The rating R of a Domain D can be calculated as

$$Tw = Fs(D) + TR + \Omega \qquad (7)$$

Here,
Fs=Feedback Score
Nt=Total number of transaction
TR=Trust Relationship
Ω=Trust Context Factor

Algorithm 2. *Trust_Inference_Engine(DET, domain, r, sr, rp, R, P)*

```
Input:  DET, domain, r, sr, rp, R, P                                  Output: DET, TST
call Decay_Trust(DET)
call Risk_Assesment( DET, domain )
call Trust_Inheritance( DET, domain )
threshold_SR <= w1 * Tdecay of DET(r) + w2 * Rdecay of DET(r) + w3 * Trust Inheritance of
DET(r) - w4 * Risk_Assesment of DET(r)
for i=1 to length(DET) do
     if   threshold of SPD(i) < threshold_SR then
          TST_ID   <= i
          TST_SR<= threshold_SR
     end if
end for
```

The trustworthiness of the domain with high feedback scores increases. The threshold value is used to determine a demarcation between domains with higher reputation. Algorithm 2 is of the trust inference engine which takes the value of risk assessment, trust inheritance and trust and reputation decay values before determine the threshold. The threshold is further required for providing services to the service requestors as per their values.

***Algorithm 3**. Decay_ Trust(DET)*

```
Input: DET                                                            Output: DET
for i=1 to length(DET) do
     ∑Tu of DET(i) <= ∑Tu of DET(i) – Tdecay of DET(i)
     ∑Rc of DET(i) <= ∑Rc of DET(i) – Rdecay of DET(i)
     Tdecay of DET(i) <=∑Fs of DET(i) / ∑Nt of DET(i)*100
     Rdecay of DET(i)<= ∑Rc of DET(i) / ∑Dn of DET(i)*100
End for
```

Here, Decay_Trust( ) decays the values of trustworthiness by total trustworthiness value by total number of transactions incurred and reputation values by number of domains transacted with. These decayed values are added to estimate rank of a domain in Domain Evaluation Chart (DEC).

Algorithm 4. *Risk_Assessment (DET, domain)*

```
Input: DET                                                            Output: DET
for i=1 to length(DET) do
     if DET[i].Domains= domain then
          DET[i].risk <= rank/max_rank * 100
     end if
end for
```

Risk_Assessment () calculates the risk involved in interacting with one particular domain and is calculated dividing the by rank of a domain in The risk is assessed by percentage of rank of a domain by total number of domains in DEC (Domain Evaluation Chart).

Algorithm 5. *Trust_Inheritance (DET, domain)*

```
Input: DET                                                            Output: DET
for i=1 to length(DET) do
     if DET[i].Domains= domain then
          DET[i].safety_factor <= 100-DET[i].risk
          DET[i].Trust Inheritance <= DET[i].TEL * DET[i].safety_factor
     end if
end for
```

The trust inherited is the percentage of risk assessed subtracted by 100 and is stored for the domains which are interacting for the first time.

A malicious user may try to increase values of untrustworthy domain by giving good feedback scores. The rating R is calculated by reducing the score as the number of transaction increases. A window *W (N, T)* is applied which calculates N number of transactions in a time interval T and further decreases the value of the domains by using equation 4.

The trust value is decayed using equation 2 after a fixed interval of time. As the value becomes 0, the domain is updated as non functional in Trust Finder and a log file is maintained at a remote server which can be invoked as per need basis. The domain whose trust score becomes 0 is enforced to establish trust again by re-establishing trust relationships and transacting with other domains.

Algorithm 6. *Add_Feedback(ETT)*

```
Input: ETT                                              Output: DET, ETT
for i=1 to length(DET) do
        receiver <= SR of DET ( i )
        if receiver = R then
            TEL_SR<= SR of TEL of ETT(i)
        end if
        provider<= SP of DET( i )
        if provider = P then
            TEL_SP <= SP of TEL of ETT(i)
        end if
end for
ETT_SR <= TEL_SR + sr
ETT_SP <= TEL_SP + sp
for j=1 to length(DET) do
        if R = Domain Name of Domain Evaluation Table then
            ∑Fs of Domain Evaluation Table <= ∑Fs of Domain Evaluation Table + sr
            ∑Rc of Domain Evaluation Table <= ∑Rc of Domain Evaluation Table + sr
            ∑Nt of Domain Evaluation Table <= ∑Nt of Domain Evaluation Table + 1
        end if
        if P = Domain Name of Domain Evaluation Table then
            ∑Fs of Domain Evaluation Table <= ∑Fs of Domain Evaluation Table + sp
            ∑Rc of Domain Evaluation Table <= ∑Rc of Domain Evaluation Table + sp
            ∑Nt of Domain Evaluation Table <= ∑Nt of Domain Evaluation Table + 1
        end if
end for
```

Add_Feedback() aggregates the feedback for service requestor as well as provider for trustworthiness and recommendation.

In case the domain is not able transact or report status due to connection failure or hardware failure. The Trust Finder database is updated and domain is communicated to report status. The domain is erased form the Trust Finder database after waiting for a reply for a time interval t.

7. RESULTS AND DISCUSSIONS

In addition to developing a theoretical model for TUX, we also conduct a comprehensive performance analysis using various trust metrics as discussed in Section 4, The figures shown are self explanatory however a brief analysis is given here:

Firstly, to evaluate and establish trustworthiness and reputation of a domain in a dynamic environment as Grid, we have taken feedback of the players i.e. service requestor and service provider's into account. On the basis of which, we have further calculated their ranks based on their trustworthiness and recommendation score.

In Figure 2, as the rank of the domain decreases, the values of trustworthiness and reputation as well decrease. Due to dynamic nature and uncertainty in grid environment, the values of trustworthiness and reputation are used only after decaying them with the T_{decay} and R_{decay} functions as shown in Figure 3.

In Figure 4, it is observed that he risk involved in transacting with a domain increases with the decreases in rank of a particular domain. Figure 5, 6 and 7 shows that the behavior of α, β and γ when we observe the service as printing, computational and data storage varies.

In Figure 8, the trust inherited is decreased with the increase in the rank of the domains.

8. RELATED WORK

In grid environments, there are two different types of trust management systems: Reputation based and policy based [17]. A number of trust models have been proposed by different researchers for evaluation of trust in a grid.

Li Xiong and Ling Liu have proposed a PeerTrust[2] model. PeerTrust-is a reputation based trust supporting framework, which includes a coherent adaptive trust model for quantifying and comparing trustworthiness of peers based on a transaction-feedback system and a decentralized implementation of such model over a structured P2P overlay network.

B. Dragovic and E. Kotsovinos's XenoTrust [3] is built on the XenoServer Open Platform [4]. Unlike simple peer-to-peer recommendation services, XenoTrust is concerned with pseudonymous users, associated with real-world identities, running real tasks on real servers for real money within a global-scale federated system whose constituent parts may have different notions of "correct" behavior. NICE trust management system [5] was developed at the University of Maryland. The NICE framework, is a platform for implementing cooperative applications over the Internet, which can be defined as a set of applications that allocate a subset of resources, typically processing, bandwidth, and storage, for use by other nodes or peers in the application. Therefore, grid computing is naturally an application for the NICE trust management framework.

Secure Grid Outsourcing (SeGO) system [6], [7] was developed at the University of Southern California. SeGO is developed for secure scheduling of a large number of autonomous and indivisible jobs to grid. A unique feature of the work is that the authors use a fuzzy inference approach to binding security in trusted grid computing environment. Abdul Rahman and Hailes proposed a Trust–Reputation Model [8] based on prior experiences based on trust characteristics from social sciences. F. Azzedin and Maheswaran proposed a Trust Model for Grid Computing Systems [9] which is extension of [8] and [10]. They have insisted on that a direct trust value weighs more than a recommender value. The model lets a newcomer to build trust from scratch by enforcing enhanced security. Here, trust is dynamic, context specific, based on past experiences and spans over a set of values ranging from very trustworthy to very untrustworthy. Farag Azzedin and Muthucumaru proposed a Trust Model [11] for peer to peer computing systems also. In addition to previous model [11], an accuracy measure is associated with each recommendation. Chin Li, V Varadharajan, Yan Wang and V. Pruthi proposed a Trust Management Architecture [12] for enhancing grid security that explores the three dimensional view of trust which includes belief, disbelief and uncertainty. This subjective logic based trust evaluation is based on Dempster-Shafer theory [13]. Z. Liang and W. Shi proposed a PErsonalized Trust Model (PET) [14] for peer to peer resource sharing. PET has accommodated risk assessment which is done to perceive the suddenly spoiling peer. Eigentrust[18] computes a global trust value for a peer by calculating the left principal eigenvector of a matrix of normalized local trust values, thus taking into consideration the entire system's history with each single peer.

9. CONCLUSION

We have presented TUX-TMS: an extensible Reputation based Trust Management system which computes the trustworthiness and reputation of the interacting domains in a grid scenario. The grid domains are dynamic and heterogeneous in nature. The Identity Management System incorporates authentication, authorization, confidentiality, log related and trust related functions. The trust decay factor ensures that the database is maintained up to date by updating when the values are used. TUX-TMS is a secure and reliable system. In addition to developing a theoretical model for TUX, we also conduct a comprehensive performance analysis. Our evaluation results show that both reputation (long-term behavior assessment) and risk (short-term behavior assessment) are important in designing a TUX-Trust Management Model. The results also show that the TUX model is flexible enough for identity and behavior trust by incorporating Audit trail and analysis for identity management. The characteristics of Trust Management Model such as scalability i.e Message, storage and computational overhead, security i.e Fabrication, Masquerading, Collusion, Sybil Attack and reliability have been considered while designing the model and are to be tested. The proposed model is interoperable as we have used web services in developing.

REFERENCES:

[1] P. Resnick, R. Zeckhauser, K. Kuwabara, E Friedman,,"Reputation systems", Communications of the ACM, 43(12): 45-48, December 2000.

[2] Li Xiong, Ling Liu, "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities", IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 7: 843-857, 2004.

[3] B., Dragovic, E. Kotsovinos, "XenoTrust: Event-based distributed trust management", Second International Workshop on Trust and Privacy in Digital Business, Prague (Czech Republic), 2003.

[4] B. Dragovic, S. Hand, T. Harris, E. Kotsovinos, "Managing trust and reputation in the XenoServer Open Platform", Proceedings of the 1st International Conference on Trust Management, Crete Greece, 2003.

[5] S. Lee, R. Sherwood, B. Bhattacharjee, "Cooperative peer groups in NICE", Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 50, Issue 4 : 523 – 544, March 2006.

[6] S. Song, K. Hwang, M. Macwan, "Fuzzy Trust Integration for Security Enforcement in Grid Computing", NPC 2004, LNCS 3222: 9-21, 2004.

[7] S. Song, K. Hwang., Y.K Kwok, "Trusted Grid Computing with Security Binding and Trust Integration", Journal of Grid Computing, vol. 3, no. 1: 24-34, 2005.

[8] A Abdul Rahman and S. Hailes, "Supporting trust in virtual communities", Proc. of Hawaii Intl Conference on System Sciences, pp:6007, 2000.

[9] Farag Azzedin and Muthucumaru Maheswaran, "Evolving and Managing Trust in Grid Computing Systems", Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering, 2002.

[10] N. Damianou, N. Dulay, E. Lupu and M. Sloman, "The Ponder Policy Specification Language", POLICY 2001, LNCS 1995:18-38, 2001.

[11] Farag Azzedin and Muthucumaru Maheswaran), "Trust Modeling for Peer-to- Peer based Computing Systems", Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'03), pp:99a., 2003.

[12] Ching Lin, Vijay Varadharajan and Yan Wang and Vineet Pruthi "Enhancing Grid Security with Trust Management", Proceedings of the 2004 IEEE International Conference on Services Computing (SCC'04), pp: 303-310, 2004).

[13] Glenn Shafer, "Perspectives on the theory and practice of belief functions", International Journal of Approximate Reasoning 6(3): 445-480, 1992.

[14] Zhengqiang Liang and Wesiong Shi, "PET: A Personalised Trust Model with Reputation and Risk Evaluation for P2P Resource sharing", Proceedings of the 38th Hawaii International Conference on System Sciences, pp.201 .2, 2005.

[15] Shashi, Seema Bawa, "Securing a Grid", International Conference CGCS-2008 (Cluster and Grid Computing Systems), Proceedings of World Academy of Science, Engineering And Technology, Volume 32 ISSN: 2070-3740, pp 7-12, 2008.

[16] Shashi, Seema Bawa, "Evaluating Trust in a Grid Environment", Student Research Symposium, HiPc 2008 (International Conference on High Performance Computing) Bangalore, India, 2008.

[17] Anirban Chakrabarti Grid Computing Security, Springer-Verlag Berlin Heidelberg, 2007.

[18] Sepandar D Kamvar, Mario T Schlosser, Hector Garcia Molina, “The EigenTrust Algorithm for Reputation Management in P2P Networks”, Proceedings of the 12th International conference on World Wide Web, Budapest, Hungary, pp: 640 – 651, 2003.

[19] I. Foster, and C. Kesselman, “The Grid: Blueprint for a New Computing Infrastructure”, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc., 1999.

[20] Grid Interoperability Now Community Group (GIN-CG)(2006). http://forge.ogf.org/sf/projects/gin.

[21] Sarbjeet Singh ans Seema Bawa), “A Privacy, Trust and Policy based Authorization Framework for Services in Distributed Environments”, International Journal of Computer Science, vol. 2 no. 2: 85-92, 2007.

Vitae:

Shashi, is a Research and teaching assistant in the Computer Science and Engineering Department at Thapar University, India. She received her MCA in 2005 from Punjabi university, India, Her research interests include distributed systems, cluster/network/Grid computing. Currently, she is pursuing her Ph.D. with a focus on Trust management and Interoperability in Grids. She is a member of IEEE, ACM and Anita Borg Institute for Women and Technology.

Seema Bawa is a Professor in Computer Science and Engineering Department at Thapar University, India. She holds M.Tech (Computer Science) degree from IIT Kharagpur and Ph.D. from Thapar University (TU), Patiala. Her areas of interests include Parallel and Distributed Computing, Grid Computing, VLSI Testing and Network Management. Prof. Bawa is member of IEEE, ACM, Computer Society of India, and VLSI Society of India.

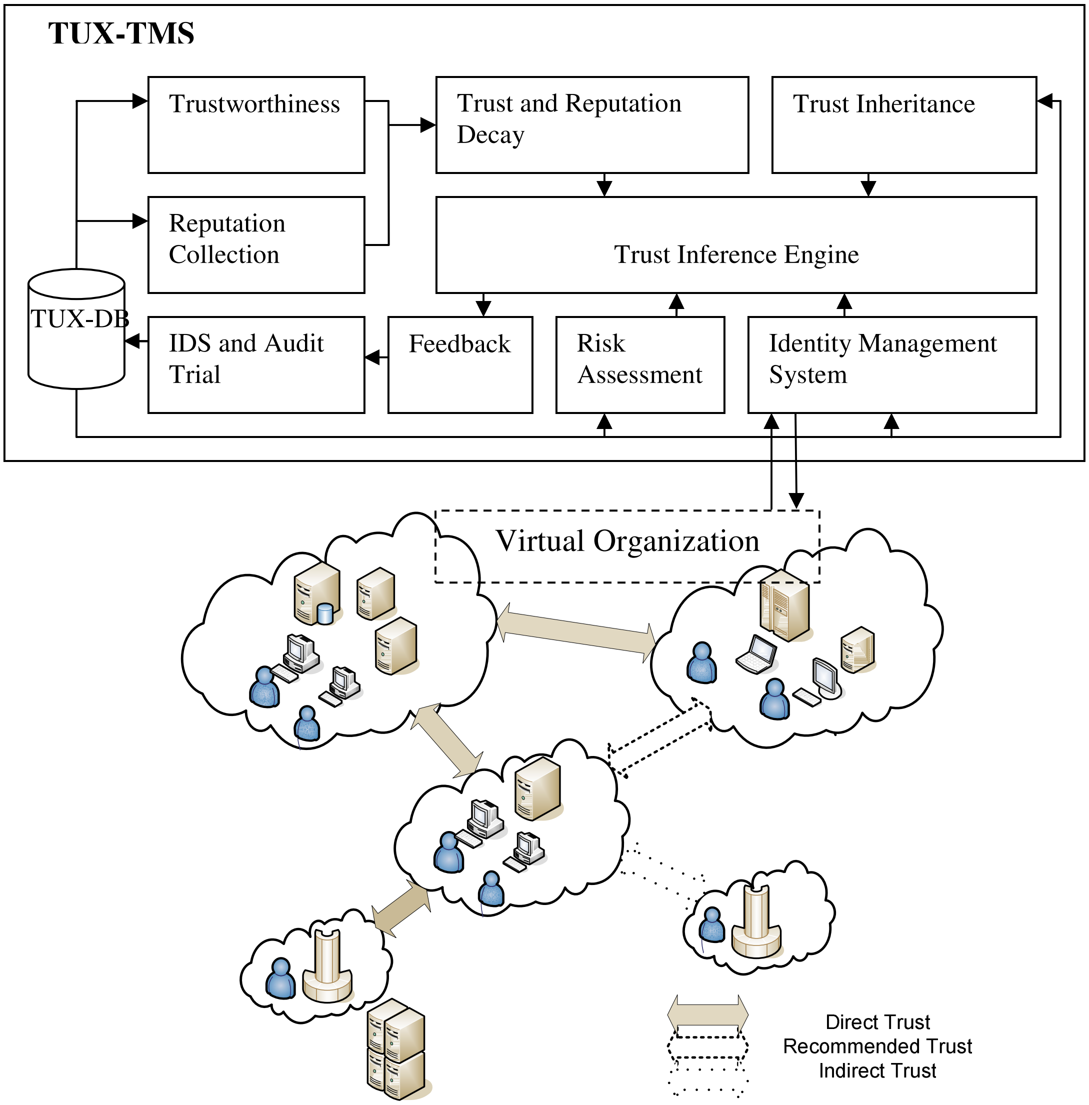

FIGURE1. TUX-TMS Architecture

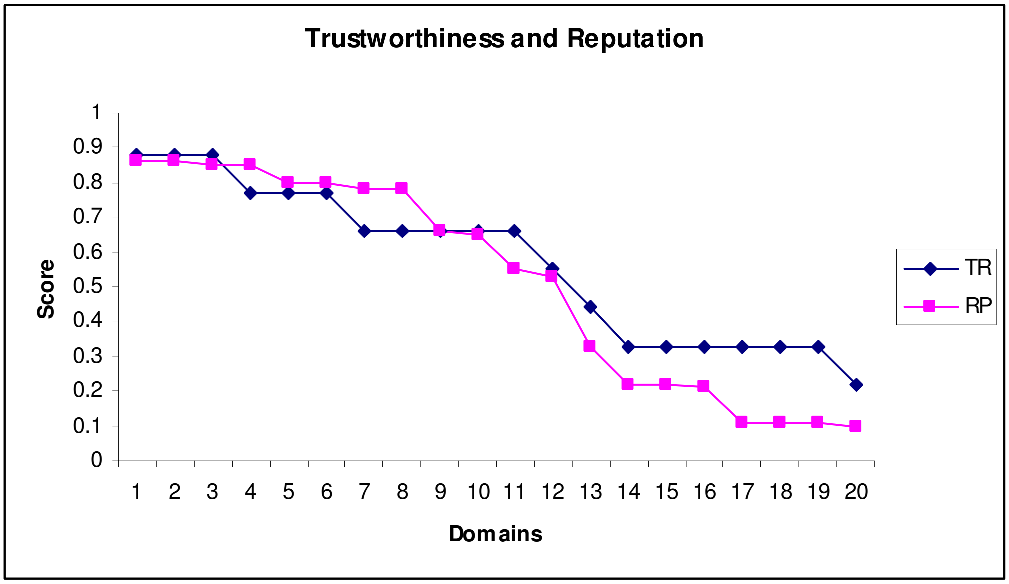

FIGURE2: Trustworthiness and Reputation values without Decay

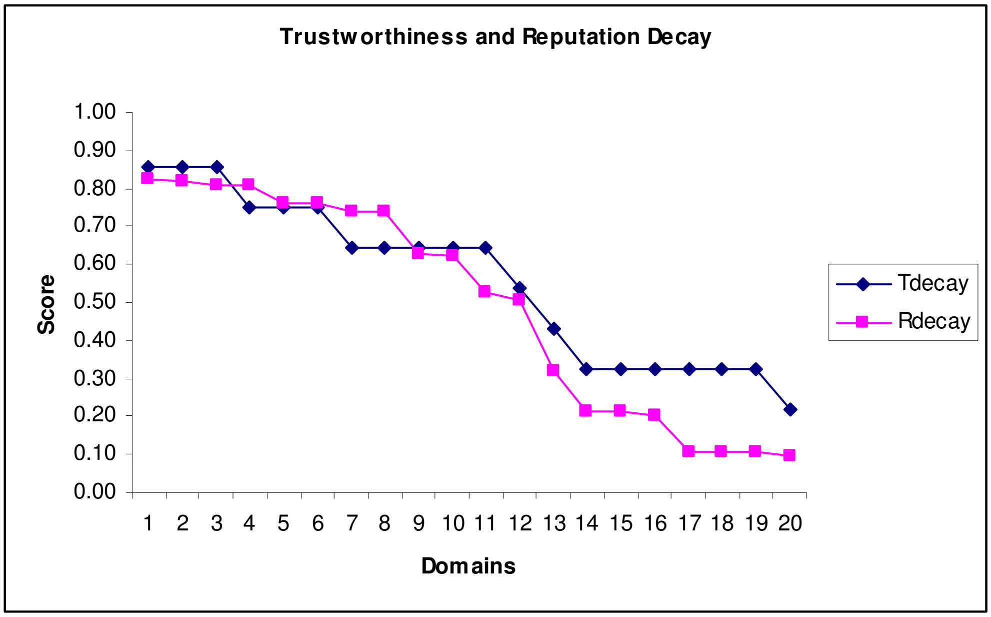

FIGURE 3: Trustworthiness and Reputation values with Decay

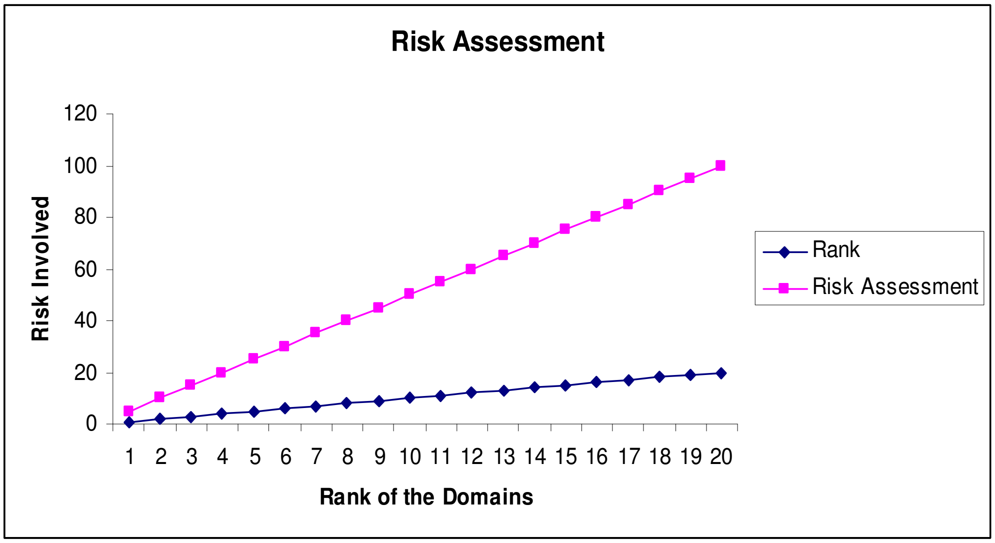

FIGURE 4. Risk Assessment

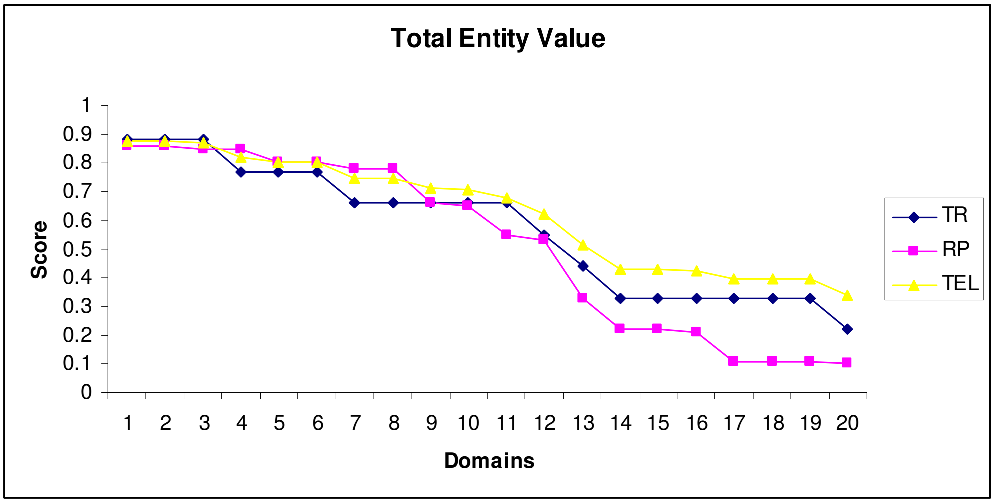

FIGURE 5. Total Entity Value when α=0.5, β=0.3, γ=0.2

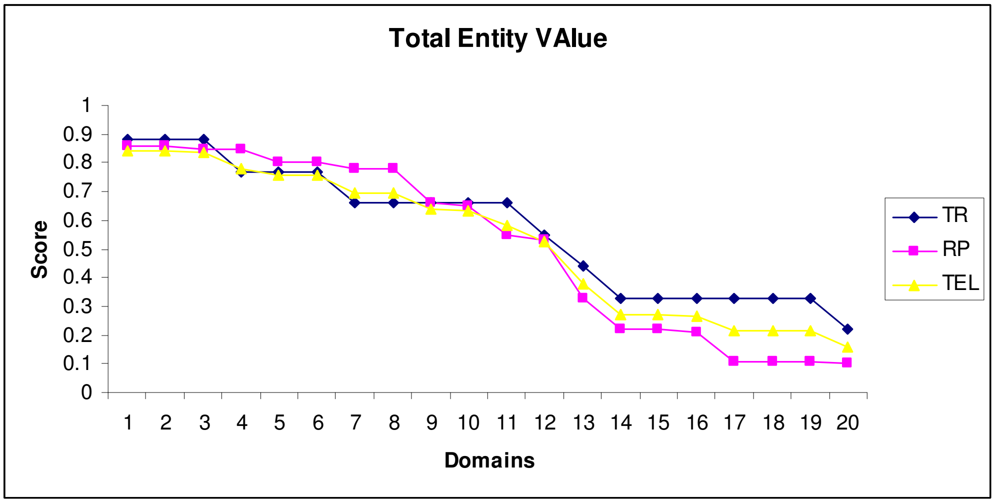

FIGURE 6. Total Entity Value when α=0.7, β=0.3, γ=0

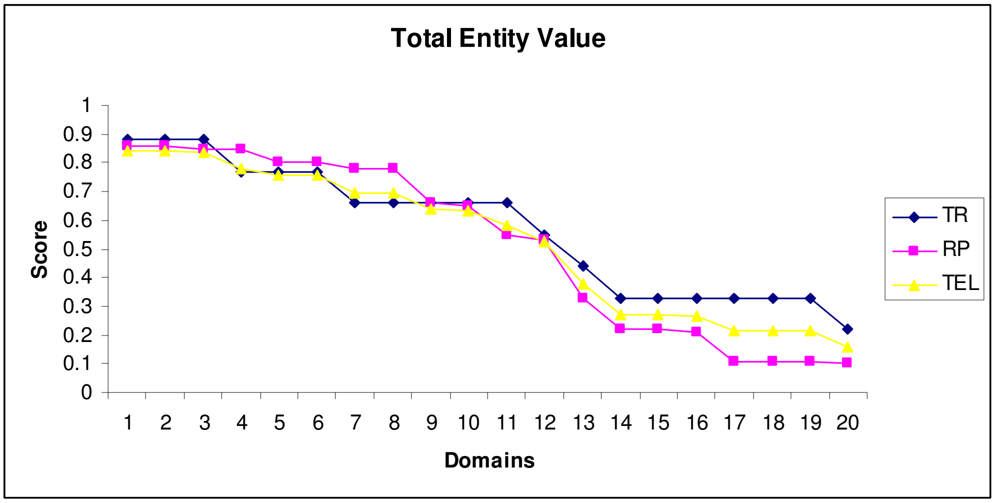

FIGURE 7. Total Entity Value when α=0.5, β=0.5, γ=0

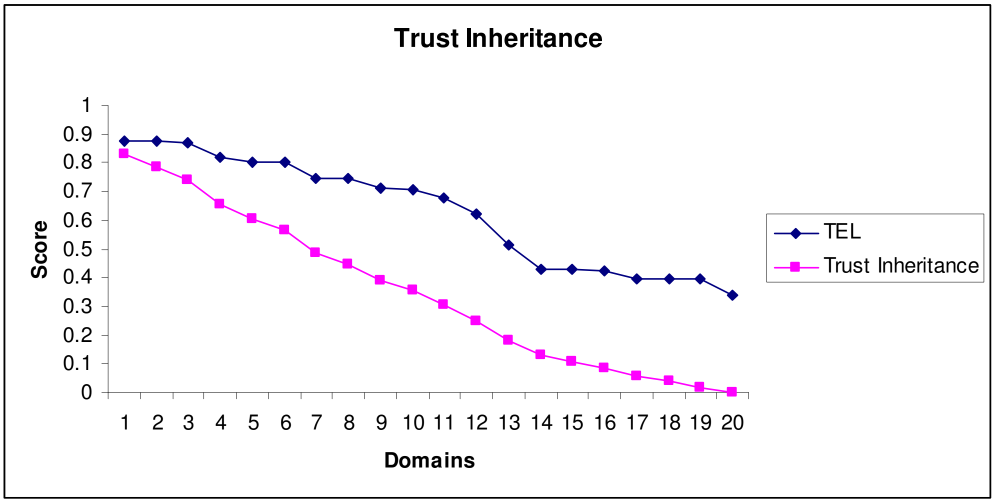

FIGURE 8. Trust Inheritance