Trust Model to Enhance Security and Interoperability of Cloud Environment*

Wenjuan Li[1,2] and Lingdi Ping[1]

[1] College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang 310058, China
[2] Hangzhou Normal University, Hangzhou, Zhejiang 310012, China
liellie@163.com, Ldping@cs.zju.edu.cn

Abstract. Trust is one of the most important means to improve security and enable interoperability of current heterogeneous independent cloud platforms. This paper first analyzed several trust models used in large and distributed environment and then introduced a novel cloud trust model to solve security issues in cross-clouds environment in which cloud customer can choose different providers' services and resources in heterogeneous domains can cooperate. The model is domain-based. It divides one cloud provider's resource nodes into the same domain and sets trust agent. It distinguishes two different roles cloud customer and cloud server and designs different strategies for them. In our model, trust recommendation is treated as one type of cloud services just like computation or storage. The model achieves both identity authentication and behavior authentication. The results of emulation experiments show that the proposed model can efficiently and safely construct trust relationship in cross-clouds environment.

Keywords: cloud computing, trust model, heterogeneous domain, role, trust recommendation.

1 Introduction

Cloud computing based on many other existing technologies is a new method for sharing infrastructure which provides customers with extremely strong computation capability and huge memory space while with low cost. But now cloud computing is faced with many problems to be resolved especially security. Till now most IT enterprises' cloud platforms are heterogeneous, independent and not interoperable. For the benefit of human society and the development of cloud computing, one uniform and interoperable cross-clouds platform will surely be born in the near future. And in cross-clouds environment, security is the most important issue. Compared to traditional technologies, cloud has many specific features, such as it is ultra-large-scale and resources belong to each cloud providers are completely distributed, heterogeneous and totally virtualized. Traditional security mechanisms such as identity validation,

* This project is supported by Chinese National Advanced Science and Technology 863(2008BA21B03 and 2008AA01A323).

M.G. Jaatun, G. Zhao, and C. Rong (Eds.): CloudCom 2009, LNCS 5931, pp. 69–79, 2009.

authentication and authorization were no longer suitable for cloud. Trust which is originally society notion in constructing human beings' relationship is now an essential substitute for former security mechanism in distributed environments. Some experts said the biggest issue of cloud computing 2009 is trust [6].

While in fact trust is the most complex relationship between entities because it is extremely abstract, unstable and difficult to be measured and managed. Today there is no special trust model for cloud computing environment. But as we know cloud has inextricably linked to distributed systems, so we try to establish our cloud trust model on the basis of in-deep research of previous studies.

This paper proposed a novel trust model which ensured the security of cloud entities both customers and providers in cross-clouds applications. It divided cloud nodes into two categories: customers and servers and designed different trust strategies for them. Trust domains were established based on independent single-cloud platform. Trust choice and update strategies took into account both the independence of nodes and manageable of domains. What's more, trust recommendation in this model was treated and managed as one type of cloud services.

This paper was constructed as follows: part 2 describes the main concept of trust and part 3 analyzes and compares several existing trust models. Part 4 introduces the new cross-clouds trust model. Part 5 shows results of our emulation experiments and the last part is conclusion and future work.

2 Definitions

2.1 Trust Relationship

The following are some correlative definitions:

* Definition1: *Trust* is referred to the recognition of entity's identity and the confidence on its behaviors. Trust is subjective behavior since entity's judgement is usually based on its own experiences. Trust is described by trust value.
* Definition2: *Trust value* or *trust degree* is used to measure the degree of trust. Trust value often depends on special time and special context.
* Definition3: *Direct trust* means trust that is obtained by entities' direct interaction.
* Definition4: *Indirect trust* or *recommended trust* means trust that is obtained from credible third party who has direct contact with the designated one. Recommended trust is one important way to obtain trust degree of unknown entities.

2.2 Classification of Trust

Trust can be classified into different categories according to different standards.

* According to attributes: identity trust and behavior trust
* According to obtaining way: direct trust and recommended trust
* According to role: code trust, third party trust and execution trust, etc.
* According to based theory: subjective trust and objective trust.

2.3 Features of Trust

In our opinion trust has the following main features:

* *Subjective, uncertainty and fuzzy*.
* *Asymmetry.* If A and B have to set up trust relationship, A's evaluated trust for B can be different from B for A
* *Inconstancy and context-sensitive*. Trust is changing along with special time and special context..
* *Condition based transitivity.* A's trust value for B is always unequal to the recommended trust that is received from C. There always exists a recommendation factor.

3 Trust Models in Distributed Environment [7-11]

With the widespread application of large scale and distributed systems such as Grid computing, Ubiquitous computing, P2P computing and Ad hoc networks, trust models fit for them have been in-depth researches. In this part we discuss the previous trust models designed for distributed systems.

3.1 PKI Based Trust Model

This trust model depends on a few leader nodes to secure the whole system. The leaders' validity certifications are signed by CA. GSI Security Infrastructure of Globus the most famous Grid toolkit is also based on PKI technology. GSI introduces the concept of user agent. PKI model may cause uneven load or a single point of failure since it rely on leader nodes too much.

3.2 Network Topology Based Trust Model

This trust model is constructed on the basis of network topology. Each entity's trust is evaluated according to its location in system topology and it usually uses tree or graph traversal algorithm. Trust management mechanism in this model is relatively simple. But due to the extremely complexity of network environment, trust values are often inaccurate which may cause system security risks.

3.3 Basic Behavior Based Trust Model

This model uses history trade records to compute trust. One entity's trust is gained by considering both former trade experiences and other nodes' recommendation. Trust value is relatively complete and reliable in this model while at the same time with large-scale computation and other burden.

3.4 Domain Based Trust Model

This trust model is mostly used in Grid computing. It divides Grid environment into several trust domains and distinguishes two kinds of trust. One is in-domain trust relationship and the other is inter-domain trust relationship. It establishes different strategies for them. The mechanism of this model is reasonable in that since nodes in the

same domain usually are much more familiar, they generally have higher trust degree for each other. This algorithm is low computational complexity because in-domain trust's computation only depends on the number of nodes in a domain and inter-domain trust only depends on the number of domains. Domain based model can be seen as a compromise between PKI and network topology. But just like PKI, it may cause network bottleneck and a single point of failure and it ignores the trust decision independence of entities.

3.5 Subjective Trust Model

Distributed applications are often faced with two major security scenarios. First, user programs may contain malicious codes that may endanger or weaken resources. Second, resources once infected by network attacks may damage user applications. So Subject logic based trust model divides trust into several subclass: execution trust, code trust, authority trust, direct trust and recommendation trust and so on. Also it designs different strategies for each kind of trust. Subjective trust is a subjective decision about specific level of entity's particular characters or behaviors. Entity A trusts entity B means A believes that B will perform certain action in some specific situation. Probability theory for example D-S theory or fuzzy mathematics is the basic tool to define trust. But generally speaking subjective trust cannot reflect fuzziness and is only reasoning on probability models which were over formalized and far away from real essence of trust management. Literature [9] proposed a new subjective trust model based on cloud model which can better describe the fuzziness and randomicity. There are other defects such as it cannot realize the integration of identity and behavior certification and the mechanism is so complex that it is difficult to realize the system based on it.

3.6 Dynamic Trust Model

Dynamic trust mechanism is a new and hot topic of security research for distributed applications. Construction of dynamic trust relationship needs to solve the following mathematics issues.

* To decide trust degree space. Always it is defined by fuzzy logics.
* To design mechanism of acquirement of trust value. There are two kinds of methods: direct or indirect.
* To design mechanism of trust value evaluation or evolution.

The research of dynamic trust model is still at the initial stage with a lot of problems to be resolved.

* Definition confusion of dynamic trust relationship. Since trust is a subjective concept there is no universal definition that can be widely accepted.
* Diversity of trust model. Dynamic trust models are based on special application environment and lack universality.
* Difficulties in the evaluation of trust model performance.
* Lack of the realization or application of model.

4 Proposed Trust Model

We proposed a novel trust model that can be used in large-scale and completely distributed cross-clouds environment based on the previous research. The following is the detail of our model.

The model differentiates two kinds of cloud roles: client and server or customer and provider. Clients are enterprises or individuals who choose to use cloud services, while service nodes are resources of cloud providers. Resources that belong to the same providers will attend the same trust domain and each domain set trust agent.

4.1 Trust Relationship Table

Each client stores and maintains a customer trust table.

Table 1. Customer trust table

Domain name	Service type	Trust value/trust degree	Generation time

The key of the table showed above contains the first and second attributes. Domain name is one cloud provider's unique identity in uniform cloud platform. Service type can be computation, storage and so on. In our model, the most specific service type is trust recommendation. When customer uses certain provider's service for the first time, it will use recommended trust provided by the other familiar providers to compute original trust. After the trade, it updates the corresponding trust according to trade result, and also it updates the recommendation factor of corresponding providers. Recommendation factor here is the trust value of recommendation service. Trust value or trust degree is used for trade judgement. The last column "generation time" is used to update trust.

Providers rely on their domain trust agent to manage trust. Agent stores and maintains domain trust table which records other domains' trust. Domain trust is used when one provider cooperates with some others, turns over customer's services, recommends trust, etc. Each time when two providers cooperate for the first time, they can also request for trust recommendation from their familiar domains. In this case recommendation is also one cooperation type. Table2 below is domain trust table.

Table 2. Domain trust table

Domain name	Cooperation type	Trust value/trust degree	Generation time

4.2 Realization Mechanism

Figure 1 shows the basic realization framework of the new model.

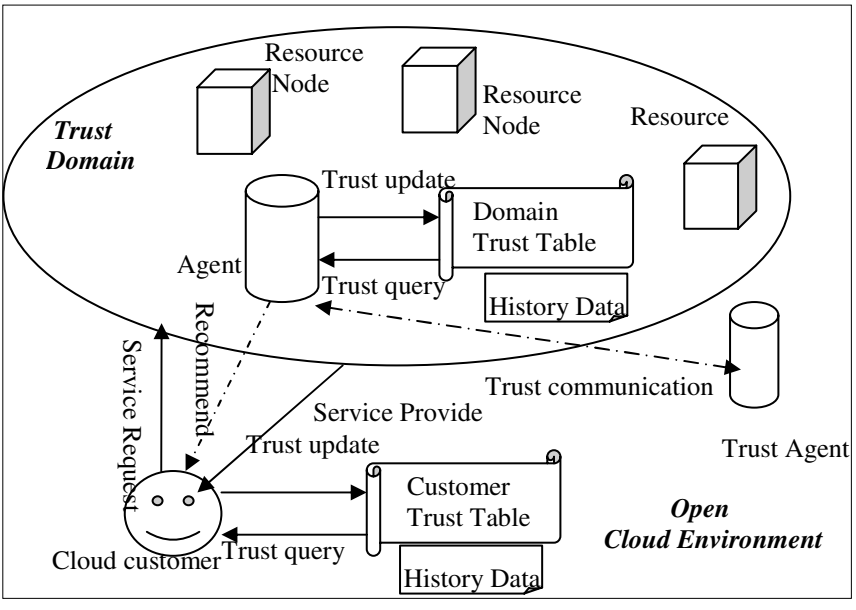

Fig. 1. Realization mechanism

4.2.1 Trust Decision

Safe transactions in cross-clouds environment are ensured by trust mechanism of which the key is trust decision. When customers want to use cloud services, they have to make trust decision. Also when service providers want to cooperate, they have to make trust decision. Explicit trust decision needs a *threshold*. Unless trust value is bigger than or trust degree is higher than the threshold, entities will not continue their transaction. In our model, threshold is customizable and cloud entity or trust domain can independently set its threshold according to its current level of security.

The general process of trust decision in our model is as follows: first of all, to search corresponding value of special trading partner in local trust table (for customer is customer trust table and for provider is domain trust table). If there exists the value and it exceeds the threshold, entity will agree to continue the transaction else transaction will be suspended. If no corresponding record is found, entity will broadcast trust request within familiar domains. And original trust for counterparty will be calculated using the received recommendation trust and corresponding recommendation factor.

Algorithm to Obtain Direct Trust for the familiar nodes.

```
  DirectSearch(nodeB,serviceType){/*example    of    nodeA
searches nodeB's trust in local trust table*/
       Boolean found=false;
       TargetNode *isTargetNode=nodeA.dir;
       While(!found||isTargetNode!=null) {
```

```
  if(isTargetNode->id==nodeB) {
                    found=true;
  for(;isTargetNode->context==serviceType||isTargetNode!=
nodeB;isTargetNode =isTargetNode->next)
  { if(isTargetNode->context==serviceType)
  {return isTargetNode->trustValue;}
  }
  } else { isTargetNode=isTargetNode->next;}
            }
        if(!found) {return -1;}
  }
```

Algorithm to Compute Recommendation Trust.

```
  RecommendSearch(nodeB,serviceType){/*example  of  nodeA
calculates nodeB's recommendation trust */
    Boolean found=false;
    float trustValue;
    RecomNode *isRecomNode=nodeA.dir;
    While(!found||isRecomNode!=null){
     if(isRecomNode->context==recommendation){
      trustValue=requestforRecom(isRecomNode,nodeB,
        serviceType);/*Agent  will  reply  recommendation
     trust for nodeB of the special trade type*/
     }
     if(trustValue>=0){
      found=true;
      return trustValue*isRecomNode->trustValue;
     }
     isRecomNode=isRecomNode->next;
     }
     if(!found) return -1;
  }
```

We suppose that rows in each trust table are already sorted by trust value in descending order. And for simplicity in recommendation circumstances, entity just chooses to use the recommendation trust of the node with highest recommendation

factor. Besides since trust is always context dependent, our algorithms take into account service types.

4.2.2 Trust Update

Two factors cause the update of trust: one is time and the other is re-evaluation of trust after each transaction. Time influence is continuous while transactions' are leaping. So the model adopts different strategies to evaluate them. It tends to use appropriate attenuation function to measure time influence. And in contrast it counts much on the evaluation of last time transaction rather than history cooperation data. Below is the different update policy for different cloud role.

* For customers:
  * To set a time-stamp and periodically delete expired records.

Example of Time Update.

```
ETimeUpdate(){
  DirNode *isDirNode=nodeA.dir;
  DirNode *p=nodeA.dir;
  IsDirNode= IsDirNode->next;
  While(p!=null) {
if(isDirNode->time>=MAXTIME) {
  p->next=IsDirNode->next;
    Delete IsDirNode;
    p=p->next;
    IsDirNode=p->next;
  }  else {
    p=p->next;
    IsDirNode=p->next;}
  }
}
```

* To re-evaluate trust after each transaction. If it is the first time, customer will increase one record in customer trust table to store the new provider's trust and at the same time update the recommendation service trust of providers who offered recommended trust. Else it just replaces the old trust with the new one.
* For agents:
  * To refresh trust using proper time attenuation function.
  * To update domain trust value after each cooperation with other domains.

5 Emulation Experiment and Results

We designed simulation experiments realizing the emulation of cross domain transactions based on proposed model and traditional domain-based model. The experiments set up two evaluation factors: trust accuracy and transaction success rate. *Trust accuracy* means the ratio of obtaining correct trust value through trust mechanism to the total number of evaluations. *Transaction success rate* means the ratio of success transactions to the ideal number of transactions.

Simulation experiments simulated cloud platform that contained 2000 nodes and 10 trust domains. In initial time, node randomly became a customer node or joined a domain and became a resource node. Each node should complete 100 times transactions. For each customer node each time, it randomly chose a certain domain to provide download service. For each resource node each time, the specific domain it belonged to randomly chose another domain to cooperate. So the total number of transactions was 200,000. Since each time before nodes began transaction they made trust decisions, the total number of trust evaluations was also 200,000. Malicious node or bad nodes in the experiments were referred to those who refused to provide services or deliberately cheat in trust recommendation. The following two figures show the results.

The results show the proposed model can ensure higher transaction success rate on the basis of relative higher trust accuracy compared to simple domain-based model in cross-domain environment with transaction fraud and malicious recommendation.

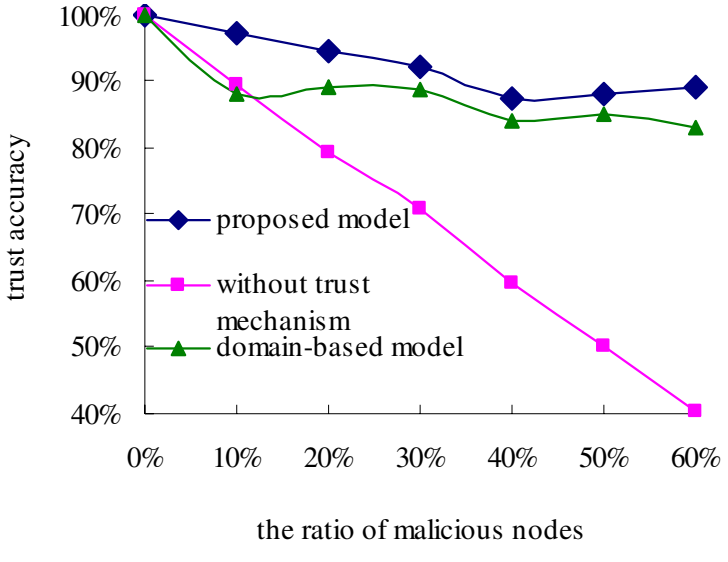

Fig. 2. Result of Trust Accuracy. X-axis represents the ratio of malicious nodes and Y-axis represents trust accuracy.

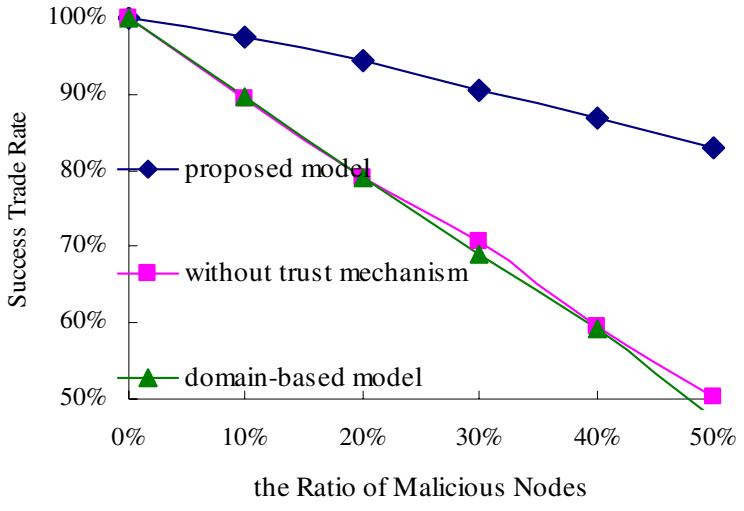

Fig. 3. Result of Success Transaction Rate. X-axis represents the ratio of malicious nodes and Y-axis represents success trade rate.

6 Conclusion and Future Work

This paper introduces a novel trust model that can be used in cross-clouds environment. We distinguished two different roles in cloud: customer and provider. Resources that belong to the same provider will be managed in the same trust domain. In each trust domain we set up a trust agent to charge domain's trust. What's more the model treats recommendation as one type of cloud service. Simulation experiments show the proposed model can establish trust relationship between customer and provider and between different cloud platforms fast and safe.

In future, there are still a lot of issues to be studied. We will establish a cross-clouds security prototype system and implement the proposed model in the test-bed. Since in reality entities behaviors are more complex and there are many other potential security risks in ultra-large-scale cross-clouds environment, we will perfect our model and improve its performance when in use and so on.

References

1. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, San Francisco (1999)
2. Chinacloud.cn, `http://www.chinacloud.cn`
3. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration. Technical report, Global Grid Forum (2002)
4. Xu, Z., Feng, B., Li, W.: Cloud Computing Technology. Publishing House of Electronics Industry, Beijing (2004)
5. Gartner.: Security Issue of Cloud Computing, `http://cio.ctocio.com.cn/12/8723012.shtml`

6. Urquhart, J.: The Biggest Cloud-Computing Issue of 2009 is Trust (2009), http://news.cnet.com/8301-19413_3-10133487-240.html
7. Li, W., Wang, X., Fu, Y., Fu, Z.: Study on Several Trust Models in Grid Environment. Journal of Fuzhou University Natural Science Edition 34(2), 189–193 (2006)
8. Blaze, M., loannidis, J., Keromytis, A.D.: Experience with the KeyNote Trust Management System. Applications and Future Directions. In: iTrust 2008, pp. 284–300 (2003)
9. Meng, X., Zhang, G., Kang, J., Li, H., Li, D.: A New Subjective Trust Model Based on Cloud Model. In: ICNSC 2008, 5th IEEE International Conference on Networking, Sensing and Control Sanya China, April 6-8, pp. 1125–1130 (2008)
10. Xiao-Yong, L.I., Xiao-Lin, G.U.I.: Research on Dynamic Trust Model for Large Scale Distributed Environment. Journal of Software 18(6), 1510–1521 (2007)
11. Song, S., Hwang, K., Macwan, M.: Fuzzy Trust Integration for Security Enforcement in Grid Computing. In: Jin, H., Gao, G.R., Xu, Z., Chen, H. (eds.) NPC 2004. LNCS, vol. 3222, pp. 9–21. Springer, Heidelberg (2004)
12. Altman, J.: PKI Security for JXTA Overlay Network, Technical Report,TR-I2-03-06, Palo Alto; Sun Microsystem (2003)
13. Perlman, R.: An Overview of PKI Trust Models. IEEE Network 13, 38–43 (1999)
14. Dou, W., Wang, H., Jia, Y., Zou, P.: A Recommendation-Based Peer-to-Peer Trust Model. Software Journal 15(4), 571–583 (2004)
15. Gan, Z., Zeng, G.: A Trust Evaluation Model Based on Behavior in Grid Environment. Computer Application and Software 22(2), 63–64 (2005)
16. Zhu, J., Yang, S., Fan, J., Chen, M.: A Grid&P2P Trust Model Based on Recommendation Evidence Reasoning. Journal of Computer Research and Development 42(5), 797–803 (2005)
17. Li, X., Michael, R., Liu, J.: A Trust Model Based Routing Protocol for Secure Ad Hoc Network. In: Proceedings of the 2004 IEEE Aerospace Conference, vol. 2, pp. 1286–1295 (2004)
18. Lin, C., Varadharajan, V., Wang, Y.: Enhancing Grid Security with Trust Management. In: Proceedings of the 2004 IEEE International Conference on Service Computing, pp. 303–310 (2004)
19. Azzendin, F., Maheswaran, M.: Evolving and Managing Trust in Grid Computing Systems. In: Proceedings of the 2002 IEEE Canadian Conference on Electrical & Computer Engineering, vol. 3, pp. 1424–1429 (2002)
20. Abdul-Rahman, A., Hailes, S.: Supporting Trust in Virtual Communities. In: Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii, vol. 1 (2000)
21. Wang, L., Yang, S.: A Trust Model in Grid Environment. Journal of Computer Engineering and Application 40(23), 50–53 (2004)
22. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A Security Architecture for Computational Grids. In: The 5th ACM Conference on Computer and Communication Security, pp. 83–92 (1998)
23. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud Computing and Grid Computing 360-Degree Compared. In: Grid Computing Environments Workshop, GCE 2008. IEEE, Los Alamitos (2008)