Interaction Modelling using Trust and Recommendation in Cloud Computing Environment

Chaitali Uikey
School of Computer Science & Information Technology
Devi Ahilya Vishwa Vidyalaya
Indore

D.S. Bhilare
Computer Center
Devi Ahilya Vishwa Vidyalaya
Indore

ABSTRACT
To ensure a secure computing in a cloud environment, recommendation and trust-based access control model is proposed. The proposed model allows calculation of direct trust and indirect trust based on recommendations. It handles cases where the requesting entity may have a past interaction experience or fresh entity without any past experience with the service. It includes the capability to cause human reasoning performance and can change by behavioral pattern modifications. Positive and negative threshold limits are used to handle malicious recommendation. The results of security mechanism so integrated with the proposed model against attacks such as bad mouthing attack, Sybil attack, and on-off behavior attack are verified.

General Terms
Cloud, Security, Algorithm.

Keywords
Interacting Model, Trust Management, Cloud Computing Environment.

1. INTRODUCTION
The purposes of cloud computing are to provide on-demand, convenient, network access to a shared group of configurable computing resources (e.g. services, application, servers, networks, storages and computing power) that can be quickly provisioned and free with minimal management effort on service provider interaction. The dynamic nature of cloud computing is extremely distributed and non-transparent. It represents a significant challenge for the acceptance and marketplace success of cloud services.

Cloud user in a cloud computing environment needs some mechanism or methods to protect data from unauthorized exposing. However, in the traditional approach to computing environment access to resources is forced by both secure authentication mechanism and physical boundaries. It is not suitable as they believe that the entities can be statically computed and allocated the suitable privileges in advance. The dynamic nature of the cloud computing means the entities (which provide services) often do not know each other and need to obtain services from an environment that are unaware a possible hostile. The cloud user has to store their sensitive data to the cloud. Cloud user feels that they are failure to control data, which are generally outside the same trusted domain as data owner. Cloud users obscure that whether cloud service providers can trust. It is an important issue that how to provide reliable computing in a cloud computing environment. To solve these issues, user access to resources must be based on trustworthiness rather than the traditional technique.

Trust is a central system of socializing with the human world. It is a human cognitive task that shoots social interaction. Recently, many researchers are working to replicate the way human evaluate trust. It is a network for socializing into a new security concept for cloud computing [1]. Due to the ambiguous requirement for access control in cloud computing the work in this paper is emphasizes on computational capability, with less concern for systematically trust infrastructure designed. This paper is based on real world characteristics of trust that develop a framework for access control in a cloud computing environment. The vision is to allow users to access resources and service from anywhere.

The user can access required service in the environment with the help of trust management. Trust to be transmitted between unknown entities, so it uses recommendations from trusted services. Open and dynamic nature of cloud computing, various malicious recommendations which provide biased recommendation to maximize their gain can also exist. This model used upper and lower threshold limit to filter out biased recommendation. Assuming that different recommender used same probability distribution. This model detects suspicious behavior and includes the concept of maximum achievable trust value that increase when an entity behaves continuously positively and decrease each time when entity behaves negatively. This model is adaptive recommendation and trust-based access control model to find calculated malicious behavior and upper and lower threshold limit to find a malicious recommendation to provide an unaffected attack model.

2. LITERATURE REVIEW
Security is the primary task in any computing environment. Trust plays a significant role in the interaction between known and unknown entities in a cloud computing environment. Trust based on human notation is widely applied to handle with new security concerns in the cloud. Blaze et al. [2] Proposed decentralized trust- management PolicyMaker. This model was based on secure application policies and credential verification to control access to resources and services. However, this model does not support the recommendation to select a suitable service. It is not feasible for cloud computing because of its complex calculation requirement. Sun et al. [3] proposed a framework to measure quantitatively trust, defend trust evaluation, and trust propagation systems against malicious attacks. Three trust model appeared as a policy based trust model, reputation-based trust model and social based trust model.

Khan KM et al. [4] give a framework of solutions using developing technologies for establishing trust in cloud computing. Tabaki H et.[5] discuss several security and privacy challenges in the cloud computing environment and trust-based framework for supporting adaptive integration

policy. Beth et al. [6] proposed a trust model for distributed networks and has distinguished recommendation trust from direct trust and gave their formal representations along with the rules to derive relationships and algorithm to compute direct trust values. Habib et al. [7] proposed a multi-faceted Trust Management system architecture for cloud computing marketplace. This system provides a means to identify the trustworthy cloud providers in term of different attributes (e.g. security, performance and compliance) assessed by multiple sources and roots of trust information. Sun Y et al.[8] investigated the benefits of introducing trust into distributed network, the vulnerabilities in trust establishment methods, and the defense mechanism. Five attacks against trust establishment methods are identified, and defense technique is developed.

Noor H et al.[9] presents a generic analytical framework that assesses existing trust management research prototypes in cloud computing and relevant areas using a set of assessment criteria. Yang Z et al.[10] a new dynamic trust approach for cloud computing is proposed where multi-level Dirichlet distribution is introduced to compute the value of trust degree. At the same tile confidence factor and time decay factor is calculated in trust evolution. Almenarez et al.[11] propose an evolutionary model of trust management that captures dynamic entities' behavior over time.

3. DEFINITIONS USED FOR PROPOSED MODEL

Here present the definition used for the proposed model.

Cloud computing environment: A model for delivering information technology services in which resources are retrieved from the internet through web-based tools and applications, rather than a direct connection to a server. Data and software packages are stored on servers.

Entity (E): Entities represent the set of the participating entities. Entities can be service owner, an account, a service, service provider node or any other entity on its behalf. Service owner that access services that are provided by the service provider. The service provider is a physical organization that provides resources and services in a cloud environment. Resources or services in a cloud environment may come from same or different service providers.

Service (S): Service is a software that provides some functionality and can be accessed /provided by the entities. Different service providers offer services. Services are exposed to the cloud environment along with their security policy and associated trust requirements.

Resources (R): Resource is an object that is accessed /provided by entities. It is a storage device, software, data, CPU, or any other devices. Entities access/provide resources through services. Entities access resources based on authorization, authentication and their conformance to established security policies. In other word, Resource is a service.

Trust (T): The mathematical definition of trust is given by Dimitrakos [12].

"Trust of an entity X in an entity Y for a service Z is the quantifiable belief of X in Y behaving consistently for a specified duration within a specified context about Z".

In this definition, an entity can be a service user, an account, a collection of processes/resources, or a system; the term service includes recommendations, issuing certificates, underwriting, transactions; consistently is used roughly to include security, privacy, trustworthiness, correctness, and maintainability; a duration may be the length of the service, refers to the past, future, or always; finally, the term context refers to the relevant service level agreements, service history, regulatory frameworks, technology infrastructure, legislative and that may apply.

Policy (P): policy is a set of rule, the requirement that associated with entities, service, and domain. Many policies exist, such as authentication policy, authorization policy, trust policy, privacy policy, security policy, management policy, application policy any other policy. Service, trust, application policy are the subset of policy.

Service Policy & Trust Policy (SP&TP): Service policy is the set of service rules and requirement associated with a particular service. Trust policy is a set of rules and requirements related to particular services and entities. Entities must conform to associated services and trust policy to access that service and these services fulfil the demands of entities.

Service Interface (SI): A service normally has a different phase, each directing a different circle of users. The service interface defines these stages within the same service.

Security Service Factor (SSF): Each service maintains a set of numeric values that define the security level of service interfaces. The security factor represents the charge of reward/penalty after each interaction.

Trustee Representation (TR): A trust value maps the level of trust a service can have an entity. In this approach, a higher trust value 1 corresponds to the total presence of trust, and lower value 0 corresponds to complete absence of trust. The occurrence of 0 means only if the service completely distrusts an entity. The table 1, shows the trust levels in continuous and discrete, their corresponding trust values and their explanation.

Level	Value	Meaning	Level in continuous	Level in discrete	Explanation for direct	Explanation for Recommendation
0	0	Distrust	Very low	No trust	Completely untrustworthy	Completely untrustworthy
1	$0 \leq$ value < 0.25	Ignorance	Low		Can't decide	Can't decide
2	$0.25 \leq$ value <0.5	Minimal	Neutral	uncertainty	Lowest trust	The participating entity itself judges the reliability of recommender's recommendation
3	$0.5 \leq$ value $<$ 0.75	Average	Mid		Mean trustworthiness	
4	$0.75 \leq$ value < 1	Good	High		Trust by major population	
5	1	complete	Very high	Trust	Fully Trusted	

Table 1. Trust Value and its Description

4. FEATURES OF PROPOSED TRUST MODEL

The proposed framework designed for the secure relationship between known and unknown entities in a cloud environment. The model calculates credibility of each entity, investigates the activity pattern of the entity and provides a service access decision in agreement with security policy. The framework has following features:-

a. Trust relationship: Creating trust relationships between entities and a service for a particular service interface with service policy within a cloud environment. It may be indicated the degree of trust a service has, in an entity, to authorize access to a particular service interface. Trust is the subjective probability depending upon time and context. The trust model of a cloud computing system are characterized by a six-tuple (Entity Trustor ET_i, Entity Trustee ET_j, Service S_K, Service Policy SP_l, Service interface SI_m, Time t)

b. In this model discrete levels of trust used, and so the trust degree or value can express equally.

$0 \leq T_v (ET_i, ET_j, S_K, SP_l, SI_m, t) < 1$ where $i \neq j$

c. There are three ways to establish trust. *Direct trust (T_{Dir})* which computed on the basis of interaction experiences the entity had with the requested service. *Recommended Trust (T_{Recom})* is when the system has no personal interaction with the entity, a corresponded view regarding the trustworthiness of an entity can be requested (also known as Indirect Trust). If new entities joining a cloud computing environment for the first time that have neither any evidence of past interaction experience nor any recommendation. In this case, these entities have assigned the *IgnoranceTrust* (T_{Ignor}) which can be updated as added information become available.

$\exists T_v (ET_i, ET_j, S_K, SP_l, SI_m, t) = T_v (ET_i, ET_j, S_K, SP_l, SI_m, t) \rightarrow T_{Dir}(ET_i, ET_j, S_K, SP_l, SI_m, t) \vee T_{Recom} (ET_i, ET_j, S_K, SP_l, SI_m, t) \vee T_{Ignor}(ET_i, ET_j, S_K, SP_l, SI_m, t)$

d. Trust is a service interface specific. Different trust values associated with the same service with same policies but different interfaces.

$T_v (ET_i, ET_j, S_K, SP_l, SI_m, t) \neq T_v (ET_i, ET_j, S_K, SP_l, SI_o, t)$ *where* $m \neq o$

e. Trust is variant time value; the entity has no new interaction with each other, value of direct trust decay with time. Trust has obtained at time t in a view of a particular service may not be the same as the trust assigned to him in the same view, at time $t+\Delta t$.

$T_v (ET_i, ET_j, S_K, SP_l, SI_m, t+\Delta t) < T_v (ET_i, ET_j, S_K, SP_l, SI_m, t)$

f. The trust value increases with good achievements and decreases with bad achievements.

$\{(C_{interaction}^{+} \rightarrow T_v (ET_i, ET_j, S_K, SP_l, SI_m, t) \geq T_{v-1} (ET_i, ET_j, S_K, SP_l, SI_m, t))\}$

$\{(C_{interaction}^{-} \rightarrow T_v (ET_i, ET_j, S_K, SP_l, SI_m, t) \leq T_{v-1} (ET_i, ET_j, S_K, SP_l, SI_m, t))\}$

g. Counters limited suspicious activities gain maximum possible trust value. This model also monitors the entity activity for constant positive actions to gain maximum achievable trust value.

$\{CP_n > CP_{n-1} \rightarrow T_{MATV} < T_{MATV-1}\}$

$\{CR_p > CR_{p-1} \rightarrow T_{MATV} > T_{MATV-1}$

h. The activities of the entity change the reward/penalty rate. Reward increases with successive positive activities. The penalty increases with successive negative activities.

$\Delta R_p = \{\Delta R_p | CR_n > CR_{n-1} \wedge C_{interaction}^{+} \rightarrow \Delta R_p \geq \Delta R_{p-1}\}$

$\Delta P_n = \{\Delta P_n | CP_n > CP_{n-1} \wedge C_{interaction}^{-} \rightarrow \Delta P_n \geq \Delta R_{n-1}\}$

5. ARCHITECTURE OF PROPOSED TRUST MODEL

Traditional security management fails to provide required elasticity for interaction between service providers and service users in a cloud environment. Proposed trust-based security architecture based on the human notion of trust to allow access to service and resources in a cloud environment. Entities try to access the services from the cloud. This model is establishing a trust relationship between entities and services within a cloud environment. Each service keeps a list of trustworthy entities, untrustworthy entities, trust value, the number of interactions, and time. An overview of the proposed security framework shown in figure 1.

Table2. Representative Computation of Trust Value

Iteration	Positive Behavior N_{pi}	Negative Behavior N_{ni}	Total no. of Interaction N_{ti}	Continuous Positive C_{pi}	Continuous Negative C_{pi}	Trusted value T_v
1	1	0	1	1	0	0.102
2	2	0	2	2	0	0.105
3	3	0	3	3	0	0.108
4	4	0	4	4	0	0.111
5	0	1	5	0	1	0
6	1	0	6	1	0	0.086
7	2	0	7	2	0	0.089

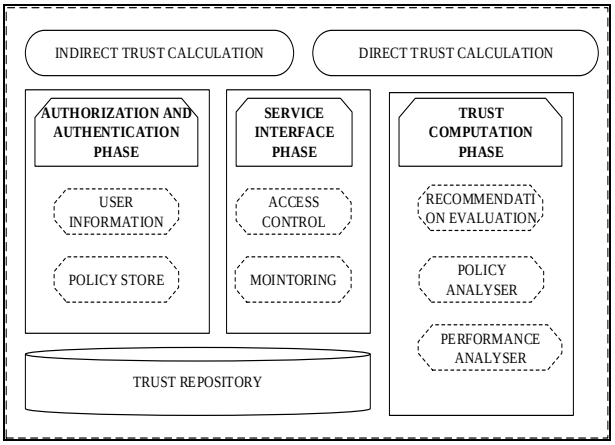

Figure1. Architecture of Proposed Trust Model

The framework consists of three phases. Authorization and authentication phase check the identification and authorization of the entity with defined policies that stored in the policy store. The model allows the service user to access shared resource or service interface in the cloud on the basis of trust value stored in trust repository of each service. If no previous trust value information exists, recommendation evaluator component takes a recommendation from peer service users or providers in cloud environments. The recommended trust value computed with indirect trust calculation components for a new trust relationship. If no recommendation is available for the entity, the ignorance trust value is assigned on the basis of the security level of entity requesting service interface. The policy analysis component in trust computation phase is to process the request; to ascertain whether the user is allowed to do the requested action in the presence of defined policies for that service interface. Performance analysis handles the growth process; it evaluates entity activities pattern involved in interaction according to its activity. It's associated with trust repository and monitoring component of the system. Direct trust calculation takes place after the result of an interaction and finding some observation from monitoring component.

5.1 Role of Policy Analyzer

The requirement of policies depends on the behaviour of associated entities. Policy-based management approach handles the malicious behavior of entities. It includes a set of rules for planned attack discovery, with suitable behavior and manage to counter these attacks. Policies are defined in this model are:

a. Different trust levels and its values described in table 1.

b. The rate of reward or penalty determined by service that is checked by security factor.

c. Suspected entity is permitted to interact again after absolution time.

d. T_{maxtru} of the entity decreased each time when an entity changes its behaviour and is produced equal to current trust value.

e. If the continuous positive behaviour occurs, then T_{maxtru} is incremented.

f. An entity is suspected when current trust value is 0.

5.2 Direct Trust Calculation

The performance analysis component handles the direct trust calculation. According to action and availability of additional evidence, it evaluates the activity sample of the entity involved in an interaction. Each service keeps the subsequent information for each entity that updated in trust evaluation.

- N_{ti} total number of interactions of entity
- N_{pi} number of positive interaction with the entity
- N_{ni} number of negative interaction with the entity
- C_{pi} number of continuous positive interaction with the entity
- C_{ni} number of continuous negative interaction with the entity
- T_{maxtru} Maximum trust that can achieve by an entity during interaction
- $countSO_{+}^{-}$ number of times the entity has swing between positive and negative activities
- *isBanned* entity constantly being banned
- *isSuspected* entity constantly being suspected
- *SF* security factor defined between $1 \leq SL \leq 5$.

All services in cloud environment do not require an equivalent stage of security. An entity can use a service have even low trust value, and with repeated positive interactions entity can grow into the trusted user of the service. However, cloud services are more responsive and require positive activities before declaring an entity entirely trusted. Similarly, negative interaction with cloud services is expected to refuse the trust at a higher rate as compared to the given service. The security factor value associated with this model. The numeric security value is assigned to each service to control the rate of reward and penalty after each interaction.

Trust evaluation occurs after finishing an interaction. If the entity has positive activities during the interaction, its positive interaction is increased otherwise the negative interaction is increased. Continuous positive and negative interaction is increased, and it's set to 0 when entity shows a change in activities. Depending on the result of the interaction, positive activity is rewarded, and a negative activity penalized. Rewarded is increasing the service trust and penalized is decreased the service trust in the entity.

The updated trust value is calculated using last trust value and current interaction in the form of reward and penalty using given equation. $C_{interaction}$ represents current interaction, $L_{interaction}$ represents last interaction, T_v and T_{v-1} represent new and old trust value, ΔR and ΔP are reward and penalty for each type of activities. All are depending on above notation.

$T_v = T_{v-1} + \Delta R$ *for* $C_{interaction}$ = *positive interaction*
$T_v = T_{v-1} - \Delta P$ *for* $C_{interaction}$ = *negative interaction*

$$\Delta R = \alpha \left(\frac{N_{pi}}{N_{ti}}\right) * 2^{\sigma * sf * c_{pi}} \qquad (i)$$

$$\Delta P = \alpha \left(\frac{N_{ni}}{N_{ti}}\right) * 2^{\frac{\sigma * c_{pi}}{sf}} \qquad (ii)$$

Where σ is a constant, and its value is 0.04, security factor value is considered as 1, α is 0.1. Table 2 represent computation of trust values.

5.3 Indirect Trust Calculation

Reputation evaluator module computes indirect trust computation. It gets a recommendation for more information when the amount of inspection is not enough for the service to see the credibility of the entity requesting services. The

reputation evaluator module evaluate the recommendation trust value of entity E_i to E_j for service S_K with service policy SP_l and service interface SI_m at time t as $T_{Recom}(ET_i, ET_j, S_K, SP_l, SI_m, t)$ respectively. Then T_{Recom} can be defined as

$$T_{Recom}(ET_i, ET_j, S_K, SP_l, SI_m, t) = \beta T_{i,j\,(p)}(ET_i, ET_j, S_K, SP_l, SI_m, t) + (1-\beta) T_{i,j(o)}(ET_i, ET_j, S_K, SP_l, SI_m, t)$$

The $\beta\, T_{i,j\,(p)}$ is a recommended trust value of peer entity in the own domain and (1-β). $T_{i,j(o)}$ is a recommended trust value of other services recommended by an entity in another domain of the cloud environment. β is a positive constant that can be set to have trust for an entity between 0 and 1. The trust value of peer-recommended to determine as

$$T_{i,j(p)}(ET_j, S_k, SP_l, SI_m, t) = \frac{\sum_{i=1}^{N} T_v(ET_j, S_k, SP_l, SI_m, t) * CF}{N_{i,j\,(p)}} \quad (iii)$$

$$T_{i,j(o)}(ET_j, S_k, SP_l, SI_m, t) = \frac{\sum_{i=1}^{N} T_v(ET_j, S_k, SP_l, SI_m, t) * CF}{N_{i,j\,(o)}} \quad (iv)$$

Where $N_{(i,j)p}$ is total peer recommendation, and similarly $N_{(i,j)o}$ is a total number of recommendations from another cloud environment.

5.3.1 Confidence Factor

The confidence factor (CF) is a way to measure the trustworthiness of recommending the service. CF = ζ * γ * SF $0 \leq CF \leq 1$ Where, ζ is normalized interaction value, γ is a time decay factor, and SF is a security factor of the recommendation service interface.

5.3.2 Time Decay Factor

Trust decay with time. The trust an entity has acquired at time t in a perspective of specified service might not be same as the trust attributed to him in the same perspective at time t [13]. Let t_c and t_l denote the current time and the last time of interaction then decay function γ is defined as

$$\gamma(t_c, t_l) = \alpha(1-\beta)^{\Delta t^k} = \alpha(1-\beta)^{(t_c - t_l)^k} \quad (v)$$

Where, k = {1, 2, 3,…….}, $\gamma(t_c, t_l) = [0,1]$

And *k* determines the rate of decay of the trust value with time Δt. α and β are adjustable positive constant that can be found accordingly to define the rate of decay. The trust decay factor starts decaying, as there are no interactions between the entities for a certain period. The trust decay factor is inversely proportional to the time. As the time increases the value of trust decay factor decreases. When the value of trust is 0, the trust is formed again. Figure 2 shows the effect of time decay factor on recommendation.

5.3.3 Effect of Confidence Factor and Security Factor on Recommendation

Recommended trust is depending on the knowledge of entity among the requesting service. The knowledge is the outcome of the interaction with an entity. The number of interactions between entities increases the confidence factor. Hence, the confidence factor is directly proportional to the number of interaction between entities. The given value of trust is $0 \leq T_v \leq 1$. The normalization function is required that can limit the

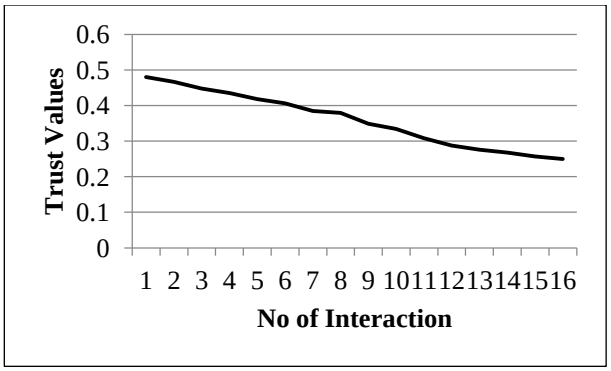

Figure 2. Effect of Time Decay Factor

number of interaction values lies between 0 and 1. The function for normalize interaction value (N_{ti}) given as

$$\zeta = \frac{N_{ti} - N_{ti}^{min}}{N_{ti}^{Max} - N_{ti}^{min}} \quad \text{where } 0 \leq \zeta \leq 1 \quad (vi)$$

Where N_{ti}^{min} = 1 and $1 \leq N_{ti}^{max} \leq \infty$ are minimum and maximum number of interaction.

Assuming, $\boldsymbol{N_{ti}^{max} = 100}$, the figure 3 show that the recommended trust value is directly proportional to number of interactions.

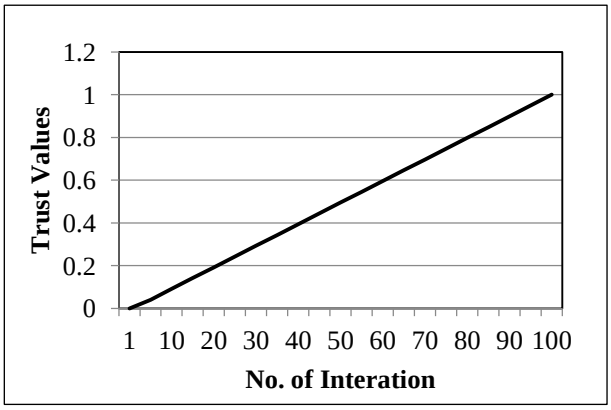

Figure 3. Effect of Confidence Factor and Security Factor on Recommendation

5.3.4 Evaluation for Recommendation

Evaluation of recommendation based trust value depends on the services that are used by entities. For example, in cloud computing a distributed file sharing service has multiple interfaces based on the type of service that is provided, and each service interface is associated with its security factor.

Malicious entities Send fake recommendations to increase the recommended trust value of requesting entity. The false recommendation can extremely manipulate the access control mechanism. In this model, a simple mechanism is used to determine whether the given recommendation is accurate or false. Recommendation evaluator collects all the recommendation former to calculate the indirect trust. Assume all recommendation values calculated from a normal

distribution. The normal distribution gives two limits one is Upper Positive Threshold Limit (UPTL) and Lower Negative Threshold Limit (LNTL). Figure 4 shows the result of detecting malicious recommendations using bad mouthing attack.

$$\text{UPTL} = \mu + \frac{5\sigma}{\sqrt{\text{N}_{\text{tr}}}} \qquad (vii)$$

$$\text{LNTL} = \mu - \frac{5\sigma}{\sqrt{\text{N}_{\text{tr}}}} \qquad (viii)$$

N_{tr} is total no of recommendation

Algorithm 1: UpdateTrustValue
Input: Entity E, SecurityFactors SF
Output: NewTrustValue
if $C_{interaction}$ = Positive Interaction then
if $L_{interaction}$= Negative Interaction then
C_{ni} = 0
else
C_{pi}++
endif
if $C_{pi} \geq C_{Thres}$ then increment T_{maxtru} endif
N_{pi} ++
ΔR= calculate increment
T_v=Min(T_{v-1} +ΔR, T_{max})
else
if $L_{interaction}$ = Positive Interaction then
C_{pi} = 0
$countSO_{+}^{-}$ ++
If $countSO_{+}^{-} \geq countThresSO_{+}^{-}$ *then*
suspected(EntityE)
endif
if $countSO_{+}^{-}$ *> 1 then decrement T_{maxtru}*
else
C_{ni}++
endif
N_{ni} ++
ΔP= calculate increment
T_v=Min(T_{v-1} - ΔP, 0)
If T_v < = 0 then suspected(EntityE) endif
endif
return T_v

Algorithm 2: Recommendation
Inputs: Recommendations
Output: RecommendationTrust
If entityE_i is outsider then Distribute recommendation request
For(Entity isRespond)
T_v= getRecommendation(ET_i, ET_j, S_K, SP_l, SI_m, t)
v++
end for loop
n_{tr}=v
compute UPTL and LNTL for recommendation threshold limit for (compute for each recommendation T_v)
if (T_x > LNTL and T_x < UPTL)
Calculate ζ_x , γ_x(t_c, t_l) and CF_x
if service is peer service in the domain then

$$T_{i,j(p)} = \frac{(T_{i,j\,(p)} + (T_X * CF_X))}{N_{i,j\,(p)} + 1}$$

endif
if service is another domain then

$$T_{i,j(o)} = \frac{(T_{i,j(o)} + (T_X * CF_X))}{N_{i,j(o)} + 1}$$

endif
endif
x++
end for loop
T_{Recom} = β$T_{i,j\,(p)}$ +(1-β)$T_{i,\,j(o)}$
endif
return T_{Recom}

Algorithm 3 ServiceAccessPermission
Input: Entity E, TrustValue T
Output: AccessPermission
if (E is not new arrival) then{
if (entityE isBanned) then AccessPermission not grant
elseif (entityE isSuspected) then
if(Δt < AbsolutionTime) then AccessPermission not grant
else
T_{Dir} =find Trustvalue of entityE
Δt= t_c-t_l
endif
endif
else
call Recommendation algorithm
if(n_{pi}+n_t ≠ 0)then compute T_{Recom}
else
allocate T_{Ignor}
endif
endif
recordTrustLevel(E, T_v)
grant ServiceAccessPermission

6. PROTECTION AGAINST MALICIOUS ATTACKS AND SOLUTION

The open environment of cloud computing creates the access control models considered for this environment vulnerable to attackers. In this section, various types of attacks against trust and reputation are investigated and how the proposed model protects against malicious attacks to provide an attack resistant model.

Bad Mouthing Attack: In bad mouthing attack, malicious users can provide dishonest recommendations to improve the trust value of malicious entities or to reduce the trust value of honest entities. Firstly, this method to avoid and detect the malicious attacks. This model uses charts with upper and lower limit to sort out a dishonest recommendation; this recommendation provided by different recommenders follows the same probability distribution. Assuming the data set of 30 recommendations in which 27% of the recommenders are providing malicious recommendation i.e. trust value < 0.5.

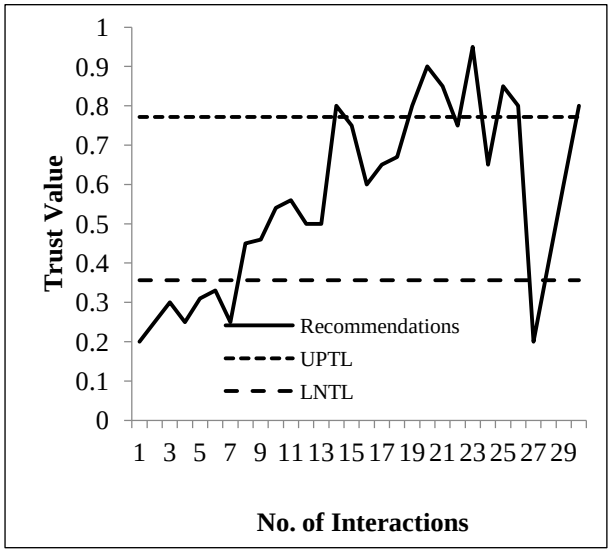

Figure 4. Detecting Malicious Recommendation

The given graph shows the particular interval UPTL and LNTL are judged as honest, dishonest and unused entities. 100% malicious recommendations were able to detect by this method. Some valid recommendations are also unused. Secondly, this method also computes the confidence factor on the recommended trust value to reduce the effect of false recommendation. It does not keep away from bad mouthing attack, but it may reduce its effects. Confidence factor based on recommended is dependent on the experienced size, last interaction time and also evaluation of recommender. The experience size calculated by the number of times when the two entities interacted. The size of experience to give more importance to the services that know the entity in question for a long time. Accordingly, assuming that with the maximum number of previous experience the trust level of the entity has already converted to a stable trust value. Thus, its conclusion must be more appropriate than the ending of an entity that has less number of interactions with the entity. This model easily distinguishes between old and new interaction, providing less weight to the valid but old recommendation. Thirdly, trust propagation chain is considered; recommendation of peer services in the own domain gives more weight in the recommendation, trust calculation as compared to a recommendation for services in other cloud domain environment.

Sybil Attack: A proper way of identity verification does not survive; a malicious entity creates many false identities to manipulate the overall behavior of the system is called a Sybil attack. In this attack, a new user can easily register as an entity, and then this is called beginner attack. These attacks completely related to the identity management system, but its influence trust management systems. The prevention of this attack is by using trust rules according with the security factor. Enrolment of the new entity takes over only if the environment experienced. Now, initially assign an ignorance trust value which gives permission to enter the cloud domain, but it requires many positive interactions to reach the trust threshold. The low trust value of the entity could not move the trust management. In a cloud environment, increase the security factor if malicious entities are present. If security factor is increasing, unknown entities will be accepted only with good recommendations. Figure 5 shows the increase and decrease in trust value.

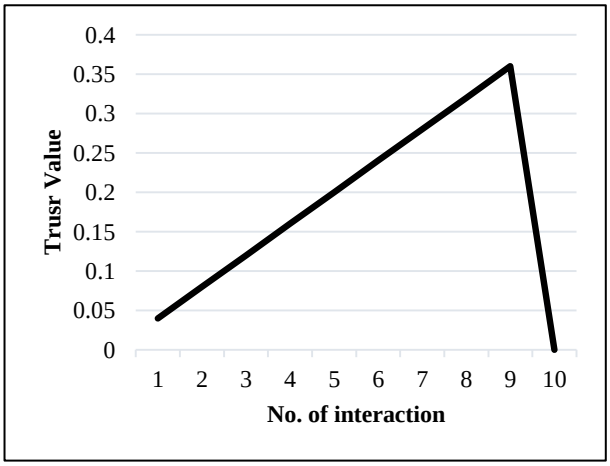

Figure 5. Increase/Decrease in Trust Value using Sybil attack

On-off Behavior Attacks: Malicious entities behave good and poorly alternatively, hoping that they will remain undetected, and their trust value will rise while causing damage. This attack attempts to develop the dynamic properties of trust through unpredictable behavior. Performance analyzer in this model calculates the behavior of entity and according to negative activity it decreases the trust value and SO_{+}^{-} count. The initial trust value is always greater than final trust value. Each time maximum achievable trust is decrease an entity shows a swung behavior. Figure 6 show the entity are banned or suspected if trust value continuous decreases. In this number of negative interactions is less than positive interaction.

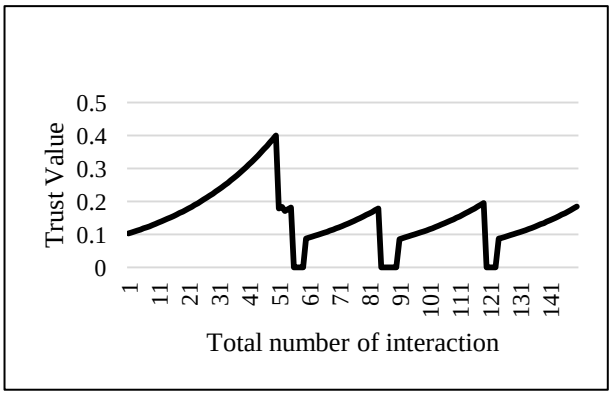

Figure 6. Swing between Positive and Negative activities using On-off Behavior attack

7. CONCLUSION

In this paper, recommendation and trust-based access control model is proposed. The proposed model handles both situations in which the requesting entity has experience with service and suspected entity without any experience and identity requesting to access the service. This paper is to define a trust evolution algorithm that dynamically adjusts the trust value according to entity behavior, thus minimizing human involvement for the security mechanism. To filter the malicious recommendation, positive and negative threshold limits are used to show the effectiveness of this model. Thus the proposed model successfully mitigates attacks such as bad mouthing attack, on-off behavior attack and Sybil attack. The future research will also be focused on implementation of the proposed model in a cloud environment.

8. REFERENCES

[1] Hang, Chung-Wei, Yonghong Wang, and Munindar P. Singh. "An adaptive probabilistic trust model and its evaluation." In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pp. 1485-1488. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[2] Blaze, Matt, Joan Feigenbaum, and Jack Lacy. "Decentralized trust management." In *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*, pp. 164-173. IEEE, 1996.

[3] Sun, Yan Lindsay, Zhu Han, Wei Yu, and KJ Ray Liu. "A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks." In *INFOCOM*, vol. 2006, pp. 1-13. 2006.

[4] Khan, Khaled M., and Qutaibah Malluhi. "Establishing trust in cloud computing." *IT professional* 12, no. 5 (2010): 20-27.

[5] Takabi, Hassan, James BD Joshi, and Gail-Joon Ahn. "Security and privacy challenges in cloud computing environments." *IEEE Security and Privacy* 8, no. 6 (2010): 24-31.

[6] Beth, Thomas, Malte Borcherding, and Birgit Klein. *Valuation of trust in open networks*. Springer Berlin Heidelberg, 1994.

[7] Habib, Sheikh Mahbub, Sebastian Ries, and Max Muhlhauser. "Towards a trust management system for cloud computing." In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, pp. 933-939. IEEE, 2011.

[8] Sun, Yan, Zhu Han, and KJ Ray Liu. "Defense of trust management vulnerabilities in distributed networks." *Communications Magazine, IEEE* 46, no. 2 (2008): 112-119.

[9] Noor, Talal H., Quan Z. Sheng, Sherali Zeadally, and Jian Yu. "Trust management of services in cloud environments: Obstacles and solutions." *ACM Computing Surveys (CSUR)* 46, no. 1 (2013): 12.

[10] Zhongxue, Yang, Qin Xiaolin, Yang Yingjie, and Li Wenrui. "A New Dynamic Trust Approach for Cloud Computing." In *1st International Workshop on Cloud Computing and Information Security*. Atlantis Press, 2013.

[11] Almenárez, Florina, Andrés Marín, Daniel Díaz, Alberto Cortés, Celeste Campo, and Carlos García-Rubio. "Trust management for multimedia P2P applications in autonomic networking." *Ad Hoc Networks* 9, no. 4 (2011): 687-697.

[12] Dimitrakos, Theo, David Golby, and Paul Kearney. "Towards a trust and contract management framework for dynamic virtual organisations." *eAdoption and the Knowledge Economy: eChallenges 2004* (2004): 27-29.

[13] Uikey, Chaitali, and D. S. Bhilare. "A Broker Based Trust Model for Cloud Computing Environment." International Journal of Emerging Technology and Advanced Engineering Volume 3, Issue 11, November 2013, pp. 247-252.