International Journal of Security and Its Applications
Vol. 10, No. 11 (2016), pp.187-200
http://dx.doi.org/10.14257/ijsia.2016.10.11.17

Blacklisting and Forgiving Coarse-grained Access Control for Cloud Computing

Khaled Riad [1,2]

[1] *School of Computer and Communication Engineering,*
University of Science and Technology Beijing, Beijing, 100083, China
[2] *Mathematics Department, Faculty of Science,*
Zagazig University, Zagazig, 44519, Egypt
khaled.riad@science.zu.edu.eg

Abstract

Cloud security is a shared responsibility between cloud providers and users. Reaching to an agreement about the dynamic policies considered for the access control decision-making process is not an easy task in cloud computing. Such dynamic policies can be built in a coarse-grained sharing manner between cloud providers and data owners. The trust notation can provide these dynamic policies, based on multiple factors that can accurately compute the user's trust level for the granting access entity. In this paper, we have introduced the formal trust definition, which imports a novel method to provide the basis for granting access. It is based on two factors and their semantic relations which investigate important measures for the cloud environment. Also, a new Blacklisting and Forgiving Coarse-grained Access Control (BF-CAC) model has been proposed. The proposed model supports changing the user's assigned permissions dynamically based on its trust level. In addition, BF-CAC ensures secure resource sharing between potential untrusted tenants. The proposed model has been implemented on our private cloud environment based on OpenStack. Finally, the experimental results have indicated that the trust level is decaying over time, thus no user can be trusted forever. Also, the number of assigned permissions for the same user is dynamically changing with the user's final trust level.

***Keywords**: Trust; cloud security; access control models; coarse-grained policy*

1. Introduction

Cloud computing is massively scalable, elastic, and inherently dynamic which pose several challenges for cloud access control. The cloud resources are provided by multiple service providers with different policies. Cloud computing is suffering from many security issues, which impacted its wide adoption for enterprises and organizations. Moreover the cloud resources are provided by different service providers, which may reside in another country with different regulations. As a shared environment, data may face issues like privacy and unauthorized access. Cloud providers offer flexible and always available access to users so, unfortunately, roles and access permissions are less controllable due to the distinguished nature of cloud computing [1]. Also, cloud users may need to acquire permissions from different domains based on the service they need. Cloud has to cooperate and spread across providers' boundaries to accomplish the requested service.

The distinguished cloud computing nature introduces novel challenges and authorization requirements, which are fully described in [1]. The basic authorization requirements, that any access control model is going to be applied for cloud computing must fulfill, are as follows: Least of Permissions (LoP), Delegation of Capabilities

ISSN: 1738-9976 IJSIA

(DoC), Separation of Duties (SoD), Integration of Policy (IoP), and Dynamic Policy Adjustment (DPA). Moreover, the recent research indicates that no comprehensive cloud access control model exists yet [2]. The detailed access control constraints are described in [3]. It should be mentioned that, for the best of our knowledge, there is no access control model has been dealt with the problem of building dynamic coarse-grained access control policies in cloud computing yet. Also, the proposed cloud access control models cannot handle the full authorization requirements.

The dynamic coarse-grained access control policies, which is a shared responsibility between the cloud provider and data owner, have to be provided on creditable basis that ensure its validity and usability. In this paper, we are proposing a new access control model that is based on building dynamic coarse-grained access control policies using trust. A trust level has to be calculated for each user using different factors, then if the user became trusted, it can be authorized for some permissions based on its trust level, otherwise it is still prohibited. In fact each user must has a trust level that consequently indicates its possessed permissions.

Our Motivation. Based on the aforementioned discussion, the traditional access control model such as Mandatory Access Control (MAC) [4], Discretionary Access Control (DAC) [5], Role Based Access Control (RBAC) [6], and Attribute Based Access Control (ABAC) [7], can no longer easily applied without course modifications. In these traditional models, the entity granting access must in advance know the identity of the entity requesting access. Looking for using ABAC for cloud computing environments in its normal state. ABAC introduces a set of hard difficulties such as the static attributes' nature, kind of attributes should be used, number of attributes considered for making access decisions is a complex task in cloud computing [8], and no proof for why the access should be allowed.

Therefore, an effective and dynamic access control model is needed. This access control model has to support the following features:

- Introducing the basis and proofs of why the access should be allowed;
- Dealing with all types of users under different circumstances and variable policies; and
- Defining a way whereby the cloud service provider can mitigate attacks and deny access to potentially dangerous users while protecting fair users.

The aforementioned requirements are behind our motivation to propose a trust-based access control model.

Our Contributions. The proposed model can deal with all types of users including new users and ensures granting access for trusted users from different service providers with different regulation policies, while protecting the users' identity. The model also supports a dynamically changing set of permissions based on the trust level for the same user. Also, the ABAC access policies are dynamically adjusted to consider the calculated trust level for the user. The last but not least, the model has a novel way to punish malicious users by blacklisting them for a specific period of time, then their trust level has to be evaluated again, while protecting fair users. Specifically in this paper:

- We state the formal trust model, based on two factors (attributes and observation) and the semantic relations between them. Through the formal trust defined in Section 2;
- A new access control model (BF-CAC) has been proposed, the model has a set of features that distinguish it from current models proposed for cloud computing:

I. It ensures granting access for trusted users while protecting the users' identity;

II. The users' permissions are dynamically changed based on the current trust level of the user;

III. Dynamically adjusting the ABAC access control policies to consider the users trust level and changing the permissions assigned based on the trust level; and

IV. The malicious users are blacklisted and cannot access the system for a specific time, until their trust level is improved and they are forgiven, then they can be assigned some permissions.

- The proposed model has been validated through experiments with an open-source OpenStack cloud platform as elaborated in Section 4.

Organization. The rest of this paper is organized as follows: Section 2 describes the formal trust model defined in this paper, its basic factors and the relations between these factors. Section 3 presents the proposed blacklisting and forgiving coarse-grained access control model (BF-CAC) and its four stages. The model analysis, implementation, and experimental verification are presented in Section 4. The related trust-based access control models for different environments are summarized in Section 5. This is followed by the conclusion and future extensions in Section 6.

2. Formal Trust

We intend to develop a centralized formal trust model that considers the service providing entities do not completely trust its tenants and partners. In this formal model, the trust is calculated based on different factors and the semantic relations between them. In this paper, these factors could consist of the user attributes and system observation. Each of these factors has its own contribution to the calculated trust level. According to the calculated trust level, one of two possible options will be assigned to the user by the service provider: either allow or block access to services.

2.1. Trust Factors and Relations

The proposed formal trust model adapts the opinion model proposed by Jøsang in 1997 [9]. In his work, opinion is represented as a triple (b, d, u) where b represents belief, d represents disbelief and u represents uncertainty. Each of these components has a value between [0,1] and the sum of these three components is 1. Thus, an opinion (b, d, u) is represented as a point in the opinion space, which in turn is represented by a unit equilateral triangle. An entity (truster α) does not completely trust a new entity (trustee β). The truster needs to construct a trust relationship with the trustee in a specific role (r). Since users can be associated with multiple roles, there is an urgent need to determine the authorization between a user and a role. Hence, the user's trust level is evaluated separately for each role r.

We assume that the trust relationship between a truster α and a trustee β in the role r depends on two factors: attributes and observation. The formal trust relationship is represented as $[C_\beta^\alpha(r), I_\beta^\alpha(r), D_\beta^\alpha(r)]$, where: $C_\beta^\alpha(r)$ is α's credibility on β for a specific role r; $I_\beta^\alpha(r)$ is α's incredibility on β for a specific role r; and $D_\beta^\alpha(r)$ is α's doubt on β for a specific role r. Each item of the trust vector $[C_\beta^\alpha(r), I_\beta^\alpha(rs), D_\beta^\alpha(rs)]$ has a value between [0,1]. The contribution of each factor (attributes and observation) to the formal trust vector has to be calculated.

2.1.1. Attributes

It is based on the attributes possessed by the trustee and its importance to the organization. The trustee β discloses a set of attributes to be verified by the truster α.

Definition 1 (Attributes (A)). *They are characteristics of the user, subject, or object. Attributes contain information given by a name-value pair:* $A = \{\{UA\}, \{SuA\}, \{OA\}\}$ *- is a finite set of user, subject, and object attributes respectively. For each attribute* (A^*) *in* $UA \cup SuA \cup OA$*, there exists* $Range_{A^*}$*, a constant finite set of atomic values. Then each attribute can be either atomic or set of values. Hence the attribute values are mapped dynamically using the following function:*

$$A^* = \begin{cases} Range_{A^*} & if\ attType(A^*) = atomic \\ 2^{Range_{A^*}} & if\ attType(A^*) = set \end{cases}$$

Algorithm 1 Attributes' Contribution to Trust

```
Input: Trustee attributes $\beta A$
for $i = 1$ to $|\beta A|$ do
    if $a_i \in \beta A \cap P_A^r$ then                          ▷ $a_i \in \beta A$
        $w_{a_i} = w_{p_{a_i}^r} \in W_{P_A^r}$                    ▷ $a_i$ is positive attribute
    else if $a_i \in \beta A \cap \mathrm{N}_A^r$ then
        $w_{a_i} = w_{n_{a_i}^r} \in W_{N_A^r}$                    ▷ $a_i$ is negative attribute
    else
        $w_{a_i} = w_{m_{a_i}^r} \in W_{M_A^r}$                    ▷ $a_i$ is mild attribute
    end if
end for
$\mathrm{AC}_\beta^\alpha(\mathrm{r})$ , $\mathrm{AI}_\beta^\alpha(\mathrm{r})$ and $\mathrm{AD}_\beta^\alpha(\mathrm{r})$ are computed using equations 1, 2, and 3 respectively
Output: Trustee's attributes contribution to trust $AT_\beta^\alpha(r) = [AC_\beta^\alpha(r), AI_\beta^\alpha(r), AD_\beta^\alpha(r)]$
```

For a user to have a specific role in an organization, the user needs to possess a set of certain attributes. Assume that the attributes defining each role r are classified into three categories: Positive Attributes (P_A^r); Negative Attributes (N_A^r); and Mild Attributes (M_A^r). The three classes of attributes (P_A^r, N_A^r, and M_A^r) are associated with a weight determined by the organizational policy, this weight reflects the importance of each attribute with respect to the role r, as follows: $W_{P_A^r}$; $W_{N_A^r}$; and $W_{M_A^r}$. Where $\{w_{p_{a_i}^r}, w_{n_{a_j}^r}, w_{m_{a_k}^r}\} \in [0,1]$, and $\sum_{i=1}^{n} w_{p_{a_i}^r} = \sum_{j=1}^{m} w_{n_{a_j}^r} = \sum_{k=1}^{l} w_{m_{a_k}^r} = 1$. Algorithm 1, illustrates the full process for calculating the trustee attributes contribution to trust. The attributes trust vector components are computed as follows:

$$AC_\beta^\alpha(r) = \frac{\sum_{i=1}^{n'} w_{p_{a_i}^r} + \frac{\sum_{k=1}^{l'} w_{m_{a_k}^r}}{2}}{\sum_{i=1}^{n'} w_{p_{a_i}^r} + \sum_{j=1}^{m'} w_{n_{a_j}^r} + \sum_{k=1}^{l'} w_{m_{a_k}^r}} \qquad (1)$$

$$AI_\beta^\alpha(r) = \frac{\sum_{j=1}^{m'} w_{n_{a_j}^r} + \frac{\sum_{k=1}^{l'} w_{m_{a_k}^r}}{2}}{\sum_{i=1}^{n'} w_{p_{a_i}^r} + \sum_{j=1}^{m'} w_{n_{a_j}^r} + \sum_{k=1}^{l'} w_{m_{a_k}^r}} \qquad (2)$$

$$AD_\beta^\alpha(r) = \frac{\sum_{k=1}^{l'} w_{m_{a_k}^r}}{\sum_{i=1}^{n'} w_{p_{a_i}^r} + \sum_{j=1}^{m'} w_{n_{a_j}^r} + \sum_{k=1}^{l'} w_{m_{a_k}^r}} \qquad (3)$$

2.1.2. Observation

The observation factor is based on the history of the past behavior of the trustee. The truster α has to collect a set of past events within a certain period of time in which the trustee β is included.

Definition 2 (*Observation (O)*). *It is the observation of the truster α about a trustee β. It is defined by the contribution of the cumulative influence of a number of past events collected by the truster about the trustee for a specific role r and over a predefined period of time $[t_0, t_n]$ $(ET_{\beta}^{\alpha}(r))$. It is assumed that the truster has to audit the trustee behavior (events) within a predefined period of time $[t_0, t_n]$. Assume that the system define a set of events related to each role, these events are classified into three categories: Positive Events (P_E^r); Negative Events (N_E^r); and Mild Events (M_E^r).*

Algorithm 2 Observation's Contribution to Trust

```
Input: Trustee events $\beta E$ related to role $r$ at $[t_0, t_n]$ period of time
for $i = 1$ to $|\beta E|$ do                              ▷ $e_i \in \beta E$
    if $e_i \in \beta E \cap P_E^r$ then                    ▷ $e_i$ is positive event
        $w_{e_i} = w_{p_{e_i}^r} \in W_{P_E^r}$
    else if $e_i \in \beta E \cap N_E^r$ then               ▷ $e_i$ is negative event
        $w_{e_i} = w_{n_{e_i}^r} \in W_{N_E^r}$
    else                                                   ▷ $e_i$ is mild event
        $w_{e_i} = w_{m_{e_i}^r} \in W_{M_E^r}$            ▷ The time instance of
    end if                                                     event $e_i$
    $t_{e_i} = T(e_i)$
    $tw_{e_i} = \frac{t_{e_i}}{n}$
end for
$OC_{\beta}^{\alpha}(r)$, $OI_{\beta}^{\alpha}(r)$, and $OD_{\beta}^{\alpha}(r)$ are computed using equations 5, 6, and 7 respectively
Output: Trustee's observation contribution to trust $OT_{\beta}^{\alpha}(r) = [OC_{\beta}^{\alpha}(r), OI_{\beta}^{\alpha}(r), OD_{\beta}^{\alpha}(r)]$
```

The three classes of events $(P_E^r, N_E^r$, and $M_E^r)$ are associated with a weight determined by the organizational policy, this weight reflects the importance of each event with respect to the role r, as follows: $W_{P_E^r}$; $W_{N_E^r}$; and $W_{M_E^r}$. Where $\{w_{p_{e_i}^r}, w_{n_{e_j}^r}, w_{m_{e_k}^r}\} \in [0,1]$, and $\sum_{i=1}^{n} w_{p_{e_i}^r} = \sum_{j=1}^{m} w_{n_{e_j}^r} = \sum_{k=1}^{l} w_{m_{e_k}^r} = 1$. Algorithm 2, illustrates the full process for calculating the truster's observation contribution to trust about a trustee β. The components of the events' trust vector are computed as follows:

$$OC_{\beta}^{\alpha}(r) = \frac{\sum_{i=1}^{n'} w_i * w_{p_{e_i}^r} + \frac{\sum_{k=1}^{l'} w_k * w_{m_{e_k}^r}}{2}}{\sum_{i=1}^{n'} w_i * w_{p_{e_i}^r} + \sum_{j=1}^{m'} w_j * w_{n_{e_j}^r} + \sum_{k=1}^{l'} w_k * w_{m_{e_k}^r}} \tag{4}$$

$$OI_{\beta}^{\alpha}(r) = \frac{\sum_{j=1}^{m'} w_j * w_{n_{e_j}^r} + \frac{\sum_{k=1}^{l'} w_k * w_{m_{e_k}^r}}{2}}{\sum_{i=1}^{n'} w_i * w_{p_{e_i}^r} + \sum_{j=1}^{m'} w_j * w_{n_{e_j}^r} + \sum_{k=1}^{l'} w_k * w_{m_{e_k}^r}} \tag{5}$$

$$OD_{\beta}^{\alpha}(r) = \frac{\sum_{k=1}^{l'} w_k * w_{m_{e_k}^r}}{\sum_{i=1}^{n'} w_i * w_{p_{e_i}^r} + \sum_{j=1}^{m'} w_j * w_{n_{e_j}^r} + \sum_{k=1}^{l'} w_k * w_{m_{e_k}^r}} \tag{6}$$

2.2. Trust Formula and Dynamics

After calculating the trust vector of each trust factor (attributes and observation), the trust evaluation policy of the truster α on the trustee β for a specific role r at time t is $T_\beta^\alpha(r)_t$ which is represented by the matrix $[AT_\beta^\alpha(r); OT_\beta^\alpha(r)]_t$.

$$\begin{aligned} T_\beta^\alpha(r)_t &= [AT_\beta^\alpha(r); OT_\beta^\alpha(r)]_t \\ &= \begin{bmatrix} AC_\beta^\alpha(r) & AI_\beta^\alpha(r) & AD_\beta^\alpha(r) \\ OC_\beta^\alpha(r) & OI_\beta^\alpha(r) & OD_\beta^\alpha(r) \end{bmatrix}_t \end{aligned} \tag{7}$$

Since each truster may need to lay more importance to one of the trust factors. So, given the same set of values for the trust factors, two trusters may have two different trust levels for the same trustee. This characteristic is called propensity of trust [11], it is based on the weight assigned by each truster to each of the trust factors. A truster may lay more importance on observation than attributes in calculating trust level for a specific trustee. Another truster may lay more importance on a different factor and so on. Hence we have introduced the concept of trust weight vector.

Definition 3 *(Trust Weight Vector). It is represented as $w_\beta^\alpha(r)_t = [Aw_\beta^\alpha(r), Ow_\beta^\alpha(r)]_t$. It is the weight of each trust factor considered for computing trust of a truster α on a trustee β for a certain role r at time t. Where $Aw_\beta^\alpha(r)$ and $Ow_\beta^\alpha(r)$ $\in [0,1]$ and $\sum(Aw_\beta^\alpha(r), Ow_\beta^\alpha(r)) = 1$.*

Therefore, the trust formula has to be modified to be weighted trust $wT_\beta^\alpha(r)_t$, which considers that the truster lays more importance on a specific trust factor, by including the trust weight vector $w_\beta^\alpha(r)_t$ for the trust of the truster α on the trustee β for a certain role r at a specific time t:

$$\begin{aligned} wT_\beta^\alpha(r)_t &= [CwT_\beta^\alpha(r), IwT_\beta^\alpha(r), DwT_\beta^\alpha(r)]_t \\ &= w_\beta^\alpha(r)_t \times [AT_\beta^\alpha(r); OT_\beta^\alpha(r)]_t \\ &= [Aw_\beta^\alpha(r), Ow_\beta^\alpha(r)]_t \\ &\quad \times \begin{bmatrix} AC_\beta^\alpha(r) & AI_\beta^\alpha(r) & AD_\beta^\alpha(r) \\ OC_\beta^\alpha(r) & OI_\beta^\alpha(r) & OD_\beta^\alpha(r) \end{bmatrix}_t \end{aligned} \tag{8}$$

Where $CwT_\beta^\alpha(r) = Aw_\beta^\alpha(r).AC_\beta^\alpha(r) + Ow_\beta^\alpha(r).OC_\beta^\alpha(r)$, $IwT_\beta^\alpha(r) = Aw_\beta^\alpha(r).AI_\beta^\alpha(r) + Ow_\beta^\alpha(r).OI_\beta^\alpha(r)$, and $DwT_\beta^\alpha(r) = Aw_\beta^\alpha(r).AD_\beta^\alpha(r) + Ow_\beta^\alpha(r).OD_\beta^\alpha(r)$.

Moreover, still the last but not least thing to find the final trust. Since trust should not only dependent on the present time calculated trust using the previous factors, but also it should consider another trust value calculated by the truster for the trustee at a specific time t_k in before. Consider that the current time trust is $T_\beta^\alpha(r)_t = [CwT_\beta^\alpha(r), IwT_\beta^\alpha(r), DwT_\beta^\alpha(r)]_t$, and the trust previously calculated at time t_k is $T_\beta^\alpha(r)_{t_k} = [C_\beta^\alpha(r), I_\beta^\alpha(r), D_\beta^\alpha(r)]_{t_k}$.The truster has to give each of $wT_\beta^\alpha(r)_t$ and $T_\beta^\alpha(r)_{t_k}$ a specific weight which reflects laying more importance to which one.

$$Let: \begin{cases} \rho \in [0,1] & is\ the\ curent\ time\ trust\ \left(T_\beta^\alpha(r)_t\right) weight \\ \varsigma = 1 - \rho & is\ the\ previous\ time\ trust\ \left(T_\beta^\alpha(r)_{t_k}\right) weight \end{cases}$$

The final trust level can be calculated based on the current time trust and the previous time trust. Thus, the weights assigned to them by the truster as follows:

$$\begin{aligned} T_\beta^\alpha(r)_t &= [C_\beta^\alpha(r), I_\beta^\alpha(r), D_\beta^\alpha(r)]_t = \rho \times wT_\beta^\alpha(r)_t + \varsigma \times T_\beta^\alpha(r)_{t_k} \\ &= \rho \times [CwT_\beta^\alpha(r), IwT_\beta^\alpha(r), DwT_\beta^\alpha(r)]_t \\ &\quad +\varsigma \times [C_\beta^\alpha(r), I_\beta^\alpha(r), D_\beta^\alpha(r)]_{t_k} \end{aligned} \tag{9}$$

3. Blacklisting and Forgiving Coarse-grained Access Control (BF-CAC)

There are several novel access control models that have been already proposed to satisfy the cloud computing access requirements. However, you cannot find a single model that satisfies all the requirements. Also, most of the proposed models do not explicitly express the basis for access control decision-making. The decision-making is one of the most important features required for adapting access control decisions dynamically in order to support elasticity and on-demand self-services. In the real world, decisions are frequently made to share sensitive information under not ideal security conditions. These decisions are made when the benefit of sharing the information overcomes the risk of sharing. The basis for these decision-makings is represented in clear understanding of the environment requirements, the expected security risks and laying more attention to the effects of similar decisions in the past. The aforementioned considerations motivate us to propose a trust-based access control model for service access control. The proposed BF-CAC model is defined in terms of a set of factors and relations between those factors with trust-based constraints defined on these relations.

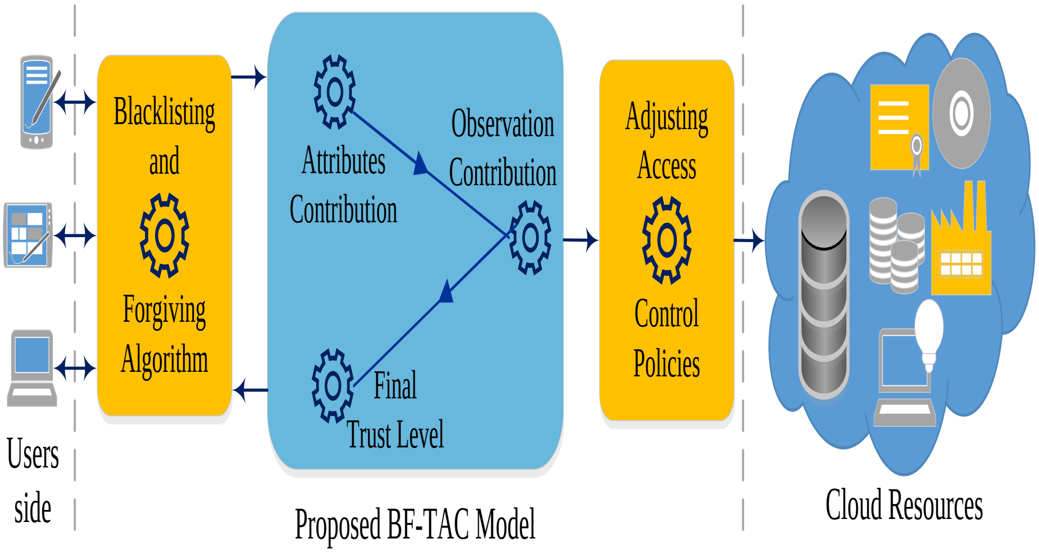

Figure 1. The Proposed Coarse-Grained Access Control (BF-CAC) Model for the Cloud Environment

BF-CAC ensures that authorized user has the access control rights (permissions) required to do subsequent security-sensitive operations. In BF-CAC the authorization component requires that the user authentication to be completed first, it is assumed that the trust level is calculated after authentication between the user (the trustee) and the service provider (the truster). Also, it is assumed that each service provider has its own access policies. These policies define sets of services that a user with a particular trust level can access. The proposed model uses ABAC [7] as its backbone and the formal trust definition provided by us in Section 2, it operates in four stages, as shown in Figure 1:

3.1. Stage One: Computing Trust Level

The service provider has to consider the provided information by the subject and compute its trust level. This can be done in three steps:

1. Calculating the attributes' contribution to trust ($AT_\beta^\alpha(r) = [AC_\beta^\alpha(r), AI_\beta^\alpha(r), AD_\beta^\alpha(r)]$) described in Section 2.1.1, using Algorithm 1;

2. Calculating the observation's contribution to trust ($OT_\beta^\alpha(r) = [OC_\beta^\alpha(r), OI_\beta^\alpha(r), OD_\beta^\alpha(r)]$) presented in Section 2.1.2, using Algorithm 2; and

3. Finally, calculating the final trust vector after considering the trust dynamics ($wT_\beta^\alpha(r)_t = [AwT_\beta^\alpha(r), OwT_\beta^\alpha(r), RwT_\beta^\alpha(r)]_t$) described in Section 2.2.

3.2. Stage Two: Adjusting ABAC Access Policies

The attribute based access control policies have to be adjusted to consider the calculated trust level. In the proposed model the service provider access polices have to be adjusted dynamically before granting a new user the right to access the system. This is useful because over time may the permissions minimum trust level is changed based on the positive and negative events encountered by the service provider during a period of time $[t_0, t_n]$ for the current system users. $[t_0, t_n]$ is the period of time considered to encounter the events of the trustee (the new user to be trusted) by the truster.

3.3. Stage Three: Blacklisting and Forgiving

Here, each user/subject can have one of three states based on its trust level and the adjusted access control polices:

- **Whitelisted,** if its trust level overcomes the predetermined trust threshold to be a trusted user;

- **Blacklisted,** if the user's trust level cannot overcome the predetermined trust threshold to be a trusted user. So the user will be in the black list and cannot access the system for a specific period of time. After that blacklisting time vanishes the user trust level will be calculated again, and the new trust level will be considered, so the user may be forgiven or blacklisted again;

- **Forgiven,** this user was blacklisted and cannot access the system for a predetermined period of time. After that penalty period vanishes, the user can apply again to the system. If its trust level for this time overcomes the predetermined threshold to be trusted, it is forgiven.

3.4. Stage Four: User/Subject Authorization

Finally, if the user/subject is not blacklisted and become trusted, it will be assigned to a set of permissions based on its trust level and the adjusted access policies. Then this user became a partner to the truster and may contribute in the recommendation process for a new user to be trusted.

4. Implementation and Analysis

The proposed BF-CAC model uses the Attribute Based Access Control (ABAC) as the backbone and integrate the formal trust model introduced in this paper with ABAC to form an effective cloud access control model.

4.1. Implementation

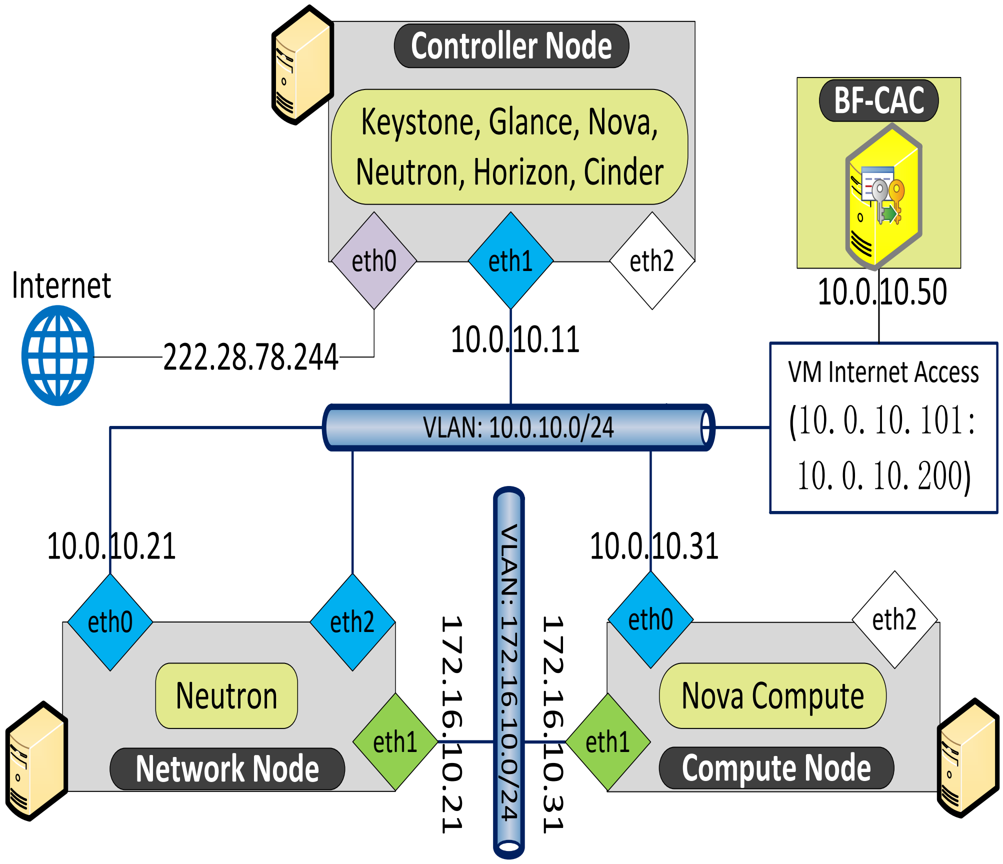

Figure 2. Our Private Openstack Cloud Environment Which Is Based On Three Physical Servers and the Proposed BF-CAC Is Implemented as a Virtual Machine

The proposed model has been implemented in Java based on the eXtensible Access Control Markup Language (XACML)[1]. The proposed model has been integrated with the Policy Decision Point (PDP) to take part in the decision-making process. The integration of the proposed BF-CAC access control model and the XACML-PDP, can be summarized here starting with policy injection to the PDP, then access requests can be initiated, until the finial obligations are enforced by the Policy Enforcement Point (PEP). The internal steps represent the communication of the different entities of the model. For simplicity reasons, the detailed information of these internal steps and its basic components are beyond the discussion of this paper.

4.2. Experimental Verification

In order to verify the proposed BF-CAC model in cloud IaaS, our next motivation is to integrate the proposed BF-CAC model with our private cloud environment which is built using the prominent IaaS platform OpenStack2 using three physical servers, as shown in Figure 2. The configuration of controller node and network node is: 48 cores CPU, 128 GB RAM and 5 TB disk and the configuration of the Nova compute node is: 24 cores

[1] http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-specos-en.html
[2] https://www.openstack.org

CPU, 128 GB RAM and 2 TB disk. Finally, the proposed BF-CAC model is implemented on a separate virtual machine which has dual core CPU, 4 GB RAM and 20 GB disk.

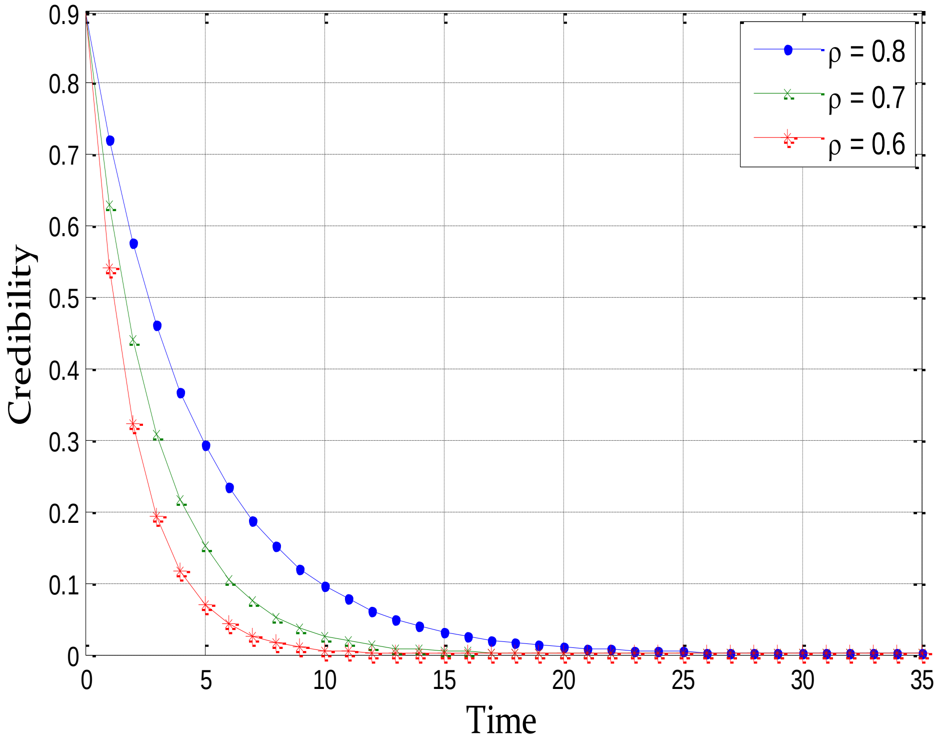

Figure 3. The Decay in the Same Credibility Value (0.9) For the Same User under the Same Role For Different Values of ρ

The proposed BF-CAC model has a considerable impact on controlling access to the cloud IaaS. Each user has to provide its possessed attributes, after the authentication process, then the user's trustworthiness is calculated based on the user attributes and system observation. The calculated trust level implies the permissions which assigned to that user during that session. It should be mentioned that the ABAC access control policies are adjusted after calculating the user trust level, hence the assigned permissions are always dynamic and mostly based on the users' trustworthiness. Since that the general tendency is to forget about the past and the final trust vector depends on the trust calculated at the current time t and the trust previously calculated at time t_k. This indicates that trust vector tends towards zero as the time increases, as shown in Figure 3.

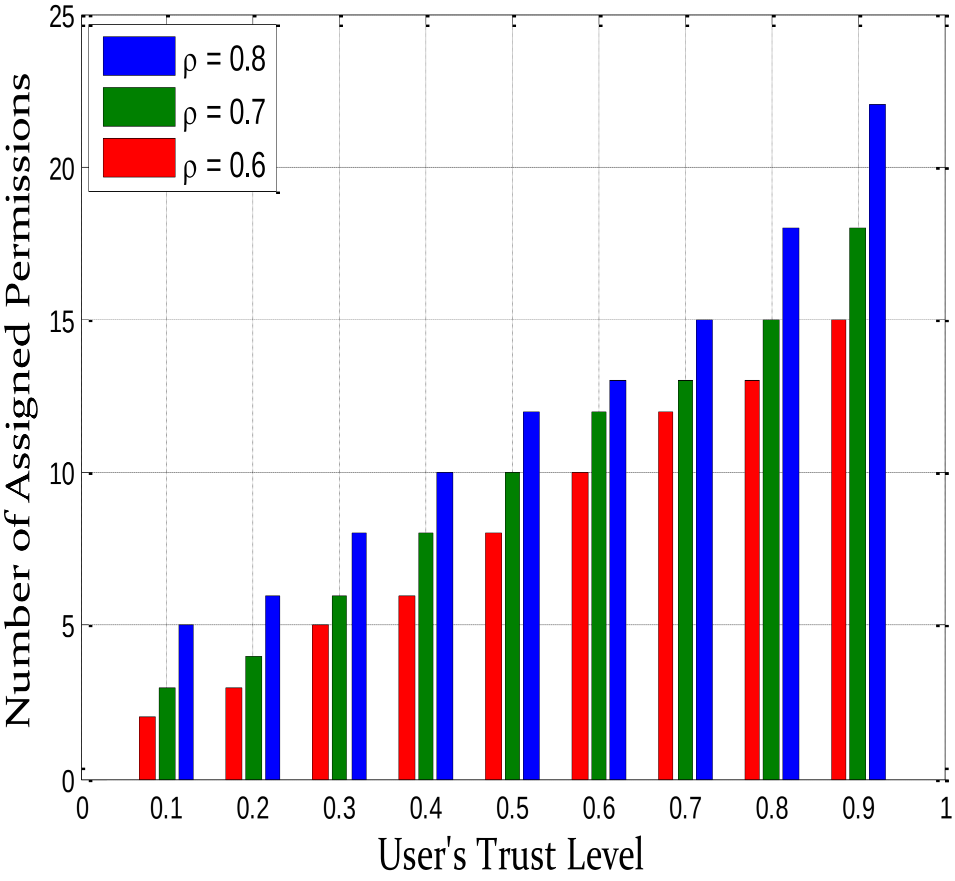

Figure 4. The Number of Assigned Permissions for the Same User According to the User's Trust Level at Different Values Of *ρ*

Still how fast the trust vector will decay over time, we have found that by laying more importance to the current time calculated trust (the value of ρ is near to one), the decay is slow. But with laying more importance to the previously calculated trust level at a specific time (the value of ρ is near to zero), the decay is very fast. Figure 3 shows the decay of the same credibility value (0.9) for the same user under the same role for different values of ρ (0.8, 0.7, and 0.6). The figure indicates that at first the value does not change much; after a certain period of time the change is more rapid; finally the change becomes stable as the value approaches to zero. Also, it is clear that the decay at $\rho = 0.6$ is more rapid than the decay at $\rho = 0.8$; and this asserts that when ρ is near to one, the decay is slow; but when ρ is near to zero, the decay is very fast. While, Figure 4 indicates that the number of assigned permissions for the same user is dynamically changing with the user's final trust level.

5. Related Work

There are some prior work looked forward for incorporating the trust notions to access control in order to address some of the dynamic systems challenges. There are various proposed cloud access control models each of them is trying to achieve some of the cloud authorization requirements. The role-based access control model and its extensions for cloud computing are introduced in [12, 13, 14]. The attribute-based access control model and its extensions for cloud computing are introduced in [7, 8, 15, 16]. Khaled Riad and Zhu Yan [17] have introduced a book, which secures the cloud environment based on the contributions of Software Defined Networking (SDN) and ABAC. There are some few contributions for introducing trust into access control models for different environments except cloud computing environment.

Khaled Riad [18] has introduced the revocation basis and proofs access control model for multi-authority cloud storage, based on the semi-formal trust model introduced in [19]. Bhatti *et al*. [20] presents a trust-enhanced version of theirs XML-based Role Based Access Control (X-RBAC) framework for web services that incorporates context-based access control. In their model the role assignment process is mainly based on trust. They define trust as the level of confidence associated with a user based on certain certified attributes. Jameel *et al*. [21] provides a trust model based on the vectors of trust values of different entities. The trust evaluation depends only on the recommendation of peer common entities to the interacting entities. The model require all entities to keep the trust values for all entities in a ubiquitous system which represents a big challenge.

Different contributions have been introduced to incorporate the trust notation with the RBAC model. In [22] the authors proposed the trust based access control model as an extension of the traditional RBAC for ubiquitous computing applications. In this model, the access privileges of a user depend on its trust level, which depends on the contextual information. Another trust based authorization model has been proposed by Chakraborty and Ray [23]. The model is an extension for the hierarchical RBAC model. The user can activate a role and invoke the assigned permissions to that role based on the user's trust level. However this model fails to capture the essential semantics of RBAC system.

6. Conclusion and Future Extensions

We have introduced the formal trust model based on two factors (attributes and observation) and the semantic relations between these factors. Also, a new blacklisting and forgiving coarse-grained access control (BF-CAC) model for cloud computing IaaS based on ABAC and the formal trust model has been proposed. The proposed BF-CAC model can fulfill a set of mandatory requirements for various access control models being deployed in cloud computing, especially cloud IaaS. In this model, the users' permissions are dynamically changed based on the user's trust level. As well as, the ABAC access control policies are dynamically adjusted to consider the users' trust level. The last but not least, the model supports a new feature for blacklisting the malicious users for a specific period of time, until their trust level improves, then they are forgiven and can be assigned a set of permissions. The future extension of this work is extending the model to include more factors for the trust calculation process.

Acknowledgments

We would like thank the anonymous reviewers for their helpful comments and suggestions. Also, we thank the National Natural Foundation of China for partly supporting this work (Grant No. 61472032).

References

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," ser. Special Publication 800-145. U.S. Department of Commerce: National Institute of Standards and Technology, October 2012.

[2] I. Ray and I. Ray, High Performance Cloud Auditing and Applications. Springer-Verlag New York, 2014, ch. Trust-Based Access Control for Secure Cloud Computing, pp. 189–213.

[3] G.-J. Ahn and R. S. Sandhu, "Role-Based Authorization Constraints Specification," ACM Transactions on Information and System Security (TISSEC), vol. 3, no. 4, pp. 207–226, November 2000.

[4] D. Bell and L. LaPadula, "Secure Computer Systems: Mathematical Foundations," Bedford, MA. Retrieved February 04, 2013, from: Secure computer systems: mathematical foundations; 1973.

[5] B.W. Lampson and P. Alto, "ACM SIGOPS Operating Systems Review," SIGOPS ACM Special Interest Group on Operating Systems, ACM New York, NY, USA, vol. 8, no. 1, pp. 18–24, 1974.

[6] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, “Proposed NIST Standard for Role-Based Access Control,” ACM Transactions on Information and System Security, vol. 4, no. 3, pp. 224–274, 2001.

[7] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, “Guide to Attribute Based Access Control (ABAC) Definition and Considerations,” ser. Special Publication 800-162. U.S. Department of Commerce: National Institute of Standards and Technology, January 2014.

[8] X. Jin, R. Krishnan, and R. Sandhu, Data and Applications Security and Privacy XXVI, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7371, no. 0302-9743, ch. A Unified Attribute-Based Access Control Model Covering DAC, MAC and RBAC, pp. 41–55.

[9] A. Jøsang, “Artificial Reasoning with Subjective Logic,” 1997.

[10] T. Grandison and M. Sloman, “A survey of Trust in Internet Applications,” IEEE Communications Surveys Tutorials, vol. 3, no. 4, pp. 2–16, 2000.

[11] R. Sandhu, D. Ferraiolo, and R. Kuhn, “The NIST Model for Role-Based Access Control: Towards a Unified Standard,” in 5th ACM Workshop on Role-Based Access Control. ACM, July 2000, pp. 47–63.

[12] Z. Tianyi, L. Weidong, and S. Jiaxing, “An Efficient Role Based Access Control System for Cloud Computing,” in 11th International Conference on: Computer and Information Technology (CIT), 2011 IEEE, Augest 2011, pp. 97–102.

[13] L. Sun, H. Wang, J. Yong, and G. Wu, “Semantic Access Control for Cloud Computing Based on e-Healthcare,” in 16th International Conference on: Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE, May 2012, pp. 512–518.

[14] K. Riad, Z. Yan, H. Hu, and G.-J. Ahn, “AR-ABAC: A New Attribute Based Access Control Model Supporting Attribute-Rules for Cloud Computing,” in 2015 IEEE International Conference on Collaboration and Internet Computing (CIC 2015), October 2015, pp. 28–35.

[15] K. Riad and Z. Yan, “EAR-ABAC: An Extended AR-ABAC Access Control Model for SDN-Integrated Cloud Computing,” International Journal of Computer Applications, vol. 132, no. 14, pp. 9–17, December 2015.

[16] K. Riad and Z. Yan, “ABAC and SDN Role in Securing the Cloud Environment,” LAP LAMPART Academic Publishing, March 2016.

[17] K. Riad, “Revocation Basis and Proofs Access Control for Cloud Storage Multi-Authority Systems,” in Proceedings of the 3rd IEEE International Conference on Artificial Intelligence and Pattern Recognition. IEEE, September 2016, pp. 118–127.

[18] K. Riad, “Multi-Authority Trust Access Control for Cloud Storage,” in Proceedings of the 4th IEEE International Conference on Cloud Computing and Intelligence Systems. IEEE, August 2016.

[19] R. Bhatti, E. Bertino, and A. Ghafoor, “A Trust-Based Context-Aware Access Control Model for Web-Services,” in IEEE International Conference on Web Services, July 2004, pp. 184–191.

[20] H. Jameel, L. X. Hung, U. Kalim, A. Sajjad, S. Lee, and Y.-K. Lee, “A Trust Model for Ubiquitous Systems Based on Vectors of Trust Values,” in Proceedings of the Seventh IEEE International Symposium on Multimedia, Washington, DC, USA, 2005, pp. 674–679.

[21] G. Ya-jun, H. Fan, Z. Qing-Guo, and L. Rong, “An Access Control Model for Ubiquitous Computing Application,” in 2nd International Conference on Mobile Technology, Applications and Systems, November 2005, pp. 1–6.

[22] S. Chakraborty and I. Ray, “TrustBAC-Integrating Trust Relationships into the RBAC Model for Access Control in Open Systems,” in the ACM Symposium on Access Control Models and Technologies, Lake Tahoe, California, USA, 2006, pp. 1–6.