Weighted Directed Graph-Based Authorization Delegation Model

Lei Jianyun
School of Computer Science, South-Central University for Nationalities, Wuhan 430074, China
Email: leijianyun@mail.scuec.edu.cn

Abstract—In this paper, a new authorization delegation model is proposed, which is based on weighted directed graph to present the trust relationship between the entities and the existing issues, such as the evaluation of the trust and the presentation of the trust among the entities are addressed in simple and visual way. The model and the corresponding algorithm of the compliance checking is analyzed and proved to be correct and efficient. Detailed description and theoretical analysis show the proposed scheme is effective to provide security authorization in actual system.

Keywords—Access Control; Trust Management; Authorization; Delegation; Weighted Directed Graph

I. INTRODUCTION

It becomes a basic requirement to do the data exchange in open multi-domain environments [1, 2]. So it is also a key factor in access control field that how to establish trust and do secure authorization and delegation between interoperability entities. Many scholars have done a lot of related research on this issue, and brought forward various solutions [3, 4]. Currently existed schemes [5, 6] that address security authorization can be divided into two categories: The first is to establish the map of the roles between the two interoperability administrative domains and the other category is trust management [6, 7].

Trust, as defined in dictionaries, is “the firm reliance on the integrity, ability, or character of a person or thing”. It describes a specific relationship between entities [8, 9, 10]. Trust relationships are normally established between entities after they have communicated and collaborated for a certain time.

Trust management has gained extensive research and application, its main feature is supporting unknown entities to access the system resources through the authorization and delegation of the other entities [11, 12], and using the appropriate trust delivery mechanism to support application scalability in distributed systems [13, 14]. To support the application of distributed systems scalability, it is necessary to use distributed transitive authorization mechanism to achieve the corresponding requirements. Because of the distributed transitive authorization mechanisms, also some of the characteristics of trust itself, there is always some limitation in current trust management systems:

1) It is hard to achieve the quantification of trust and the expression of quantitative trust.

2) It is not easy to control the transfer of trust. Even if A trusts B, but A itself is not sure that B will or not abuse the privilege, such as B authorize and delegate to C whom is not trusted by A.

3) Compared with centralized authorization, distributed authorization has great flexibility, and because of the dispersion of the authorization, it always leads to conflict authorization and ring authorization problems.

II. MODEL DESIGN

A. Elements In Weighted Directed Graph-Based Authorization Delegation Model

1) Entity
   Entity is an object like system user, all entities can be used to uniquely represent a subject or object. Entity will be represented using nodes in the weighted directed graph.

2) Role
   Role is the name of the given entity name space, it usually represent a set of access privilege.

3) Subject
   Subject is an object who gives delegation to others in the process of authorization and delegation. Subject can not only be an entity, but also be a role. If the subject is a role, then the subject represents the set of all entities of the role. Subject will be represented using the beginning of a directed section in the weighted directed graph.

4) Object
   Object is authorized object in the authorization and delegation. Object can not only be an entity, but also be a role. Similar to subject, if the object is a role, then the object represents the set of all entities of the role. Object will be represented using the end of a directed section in the weighted directed graph.

5) Trust Value
   Trust value is used to indicate the trust level of a subject to an object. A non-negative real number is used to express the trust value. The greater the trust value is, the higher the trust level is. To facilitate the calculation below, the range of [0,100] is prescribed to the trust value.

6) Authorization Root
   Authorization root is the origin point of the authorization, it is also the access control list (ACL) announced by the resource owners. ACL entry is a triad which contains privilege, subject and trust threshold. This trust threshold is a minimum that decide the access...
permission can be granted or not, that is to say, only when the direct or indirect trust value owned by the requester of the authority is greater than or equal to this value, requester can access the corresponding resources.

7) Delegation

Delegation is the process that authority passes from subject to object. Authorization certificate is used to represent a delegation. Authorization certificate and the issuer of authorization certificate must be digitally signed using its own private key. Weighted directed sections from one node to another node in the weighted directed graph will be used to represent delegation.

In this model, the concept of role follows Sandhu’s definition in RBAC 96. Delegation is the most important concept of the model because it makes all the elements referred above related together, such as authorized source, trust threshold, subject, object, trust value, and so on.

B. Definition of Weighted Directed Graph-Based Authorization Delegation Model

Definition 1. Authorization weighted directed graph: 
\[ G = (G_r) \]
\[ G_r = (V,E,w) \]
represents all authorization which associated with \( r \), in which, \( V \) is a node set, each node corresponds to a certain object’s related authorization entities, \( E \) is a section set, each section is an ordered two-tuples that consisted of authorization publisher and authorization receiver, and each section is also the privilege that conferred from authorization publishers to authorization receiver. If \( w \) \( \langle V, s \rangle \neq 0 \), then there is a weighted directed section in \( G_r \) that direct from \( g \) to \( s \) and its weight is \( w \).

Fig. 1 is an example of authorization weighted directed graph.

![Example of authorization weighted directed graph](image)

The figure can be represented using triads:

- \( \langle V_1, V_2, 80 \rangle \)
- \( \langle V_2, V_3, 70 \rangle \)
- \( \langle V_3, V_4, 75 \rangle \)
- \( \langle V_3, V_4, 85 \rangle \)
- \( \langle V_2, V_3, 60 \rangle \)
- \( \langle V_4, V_3, 80 \rangle \)
- \( \langle V_3, V_4, 60 \rangle \)

It represents:

- The trust value of node \( V_1 \) to node \( V_2 \) is 80
- The trust value of node \( V_2 \) to node \( V_3 \) is 70
- The trust value of node \( V_3 \) to node \( V_4 \) is 85
- The trust value of node \( V_3 \) to node \( V_4 \) is 75
- The trust value of node \( V_4 \) to node \( V_3 \) is 60
- The trust value of node \( V_4 \) to node \( V_3 \) is 60

III. COMPLIANCE CHECKING PROCESS IN THE MODEL

In authorization delegation models based on weighted directed graph, it is just as same as all authorization delegation models, resource access requests can be passed or not depends on “whether the certificate set \( C \) provided by the requester is able to demonstrate that the request set \( r \) is consistent with the local security policy \( P \)”.

It is the so-called compliance checking problem. Certificate chain searching and trust transmission controlling and judging the trust value from the authorization source to the end whether it is greater than the specified threshold are the key points of compliance checking problem. As long as the calculation of the trust path value from the authorization source to every other vertex is realized, we can judge the trust value whether it is greater than the required threshold, and the compliance checking process is completed.

In authorization delegation model based on weighted directed graph, the algorithm of calculating the trust path value from node \( V_0 \) to every other node can be described as below:

Directed graph can be expressed with adjacency matrix \( adj \) if \( \langle V_j, V_i \rangle \) is a section, the value of \( adj[i, j] \) equal to the directed section’s weight, otherwise the value of \( adj[i, j] \) will be 0. Initially \( adj[i, i] = 0 \), and \( adj[i, i] \) is assigned with 1 to mark that \( i \) node has entered the first group in processing. Each element of array \( dist[1..n] \) consists of two fields: \( trust \) field contains the trust value of node, and \( pre \) field is the serial number of the previous node on the path from \( V_0 \) to this node. When the algorithm ends, we can attain the maximum trust path from \( V_0 \) to \( V_i \) by tracing along with the \( pre \) field corresponding to the node, and the maximum trust value will be in the \( trust \) field.

A. Data Structure Definition

```plaintext
TYPE path=RECORD
  trust: real;
  pre: 0..n
END;
VAR
  adj: ARRAY[1..n,1..n] OF real;
  dist: ARRAY[1..n] OF path;
  max: real;
  k,i,u:1..n;
```

B. Algorithm Description

1) Input Items

The serial number of the starting node \( V_0 \) is given in the variable \( k \), and the value of the \( trust \) field of the element in \( dist \) array indexed \( V_0 \) is assigned with 100.

2) Output Items

The maximum trust path from node \( V_0 \) to every other node.

3) Steps of the Algorithm

1) The initial nodes are divided into two groups
A. Idea

When a node is added into the first group, it is necessary to compare and modify the trust value of node in the second group. If there is a node added into the first group, go on until all nodes of the graph have been added into the first group.

Loop until all nodes in the second group has been added into the first group

1) [Find the node with the maximum trust value in the second group]
i) $\text{max} \leftarrow 0; u \leftarrow 0$

if $\text{adj}[i,i] \neq 0$ and $\text{dist}[i].\text{trust} > \text{max}$
then $u \leftarrow i; \text{max} \leftarrow \text{dist}[i].\text{trust}$
else $\text{dist}[i].\text{pre} \leftarrow 0$

2) [add the nodes in the second group to the first group one by one]

(1) loop $i$ step by 1, from 1 to $n$
   i) $\text{dist}[i].\text{trust} \leftarrow \text{adj}[k,i]$
   ii) if $\text{dist}[i].\text{trust} \neq 0$
       then $\text{dist}[i].\text{pre} \leftarrow k$
   else $\text{dist}[i].\text{pre} \leftarrow 0$
   (2) $\text{adj}[k,k] \leftarrow 0$

B. Proof of the Algorithm

Division of the two groups and judging the trust value of nodes in the algorithm are clearly in accordance with the basic ideas above. The only thing to prove the correctness of this approach is to prove the division of the two groups and the trust value still meet the requirements after adding a node into the first group. That is to prove the trust value of the node $V_m$ is the maximum in the second group and the trust value is the greatest trust path value from the node $V_0$ to $V_m$ and $V_m$ is the node of the greatest trust path value in the second group.

1) If the trust value of node $V_m$ is not the maximum trust value from $V_0$ to $V_m$ and there is another path from $V_0$ through some nodes in the second group to $V_m$ whose trust value is greater than that of node $V_m$ and we assume the first passed node in the second group is node $V_s$, then the trust value of $V_s$ must be greater than that from $V_0$ through $V_s$ to $V_m$ and greater than that of $V_m$, this is conflict with the assumption that trust value of $V_m$ is the greatest in the second group. So the trust value of node $V_m$ is the maximum trust path value from node $V_0$ to $V_m$.

2) Assume node $V_s$ is any other node in the second group, and if only nodes in the first group can be the middle nodes in the maximum trust path to $V_s$, then its trust value can’t be greater than any other trust path value from node $V_0$ to $V_m$. This can be known by the definition of trust value. If the maximum trust path value from node $V_0$ to $V_s$ through middle nodes in the second group and assume that is a node named $V_s$, then the trust path value from $V_0$ to $V_s$ must be the trust value of $V_s$, it is less than or equal to the maximum trust path value from $V_0$ to $V_m$, so it is obvious that the maximum trust path value from node $V_0$ to node $V_s$ can’t be greater than the maximum trust path value from node $V_0$ to node $V_m$ so the node $V_m$ is the maximum trust path value in the second group.

C. Demonstration of the Algorithm

Take Fig. 1 for example to find the maximum trust value from node $V_i$ to every other nodes, values in array dist will be changed as the algorithm runs and the results are shown in Fig. 2.

![Figure 2. Algorithm simulated running instance](image_url)

IV. ANALYSIS OF THE COMPLIANCE CHECKING ALGORITHM

The time efficiency of the algorithm in calculating the trust path value from node $V_0$ to every other node in the model is $O(n^2)$. Because the algorithm has to add $n$ nodes into the first group, it is necessary to compare and modify each element in array dist with $n$ elements one by one when a node is added into the first group.

A. Idea

At first, there is only $V_0$ in the first group, the second group contains all the other nodes. The trust value of the node $V_0$ is 100, and the trust value of the nodes in the second group can be assigned in this way: if there is a section $\langle V_0, V_i \rangle$ in the graph, then the trust value of the node $V_i$ is the weight of the section $\langle V_0, V_i \rangle$, otherwise the trust value of the node $V_i$ is assigned with 0. Secondly, select a node $V_m$ with maximum trust value in the nodes of the second group to the first group every time. There is an amendment to the trust value of every node in the second group when the node $V_m$ is added to the first group. If added node $V_m$ is a middle node which makes the trust path value from $V_0$ to $V_i$ greater than that before $V_m$ is added as a middle node, then the modification of the trust value of the node $V_i$ is necessary. After amendment, chose the maximum trust value node and add it into the first group, go on until all nodes of the graph have been added into the first group or there is no node to be added into the first group.

© 2013 ACADEMY PUBLISHER
two shortcomings. The proposed model is simple and intuitive to understand. Algorithm of compliance checking is proved to be correct and efficient after analysis and demonstration.

ACKNOWLEDGEMENT

This work was supported by the Natural Science Foundation of Hubei Province, China (No. 2013CFB445).

REFERENCES


