Context-Aware Dynamic RBAC Model for Application Layer Multicast

FuJian Zhong* ; YuJiao LIU
Mianyang Normal University, Mianyang, Sichuan, China，621000
150980919@qq.com; 515392625@qq.com

Abstract— This paper presents application layer multicast system of the context-aware dynamic role, which need to real time tinker up the permissions owned by consumer based on the context information. We analyse key problems of layer multicast system in application layer multicast communication. In order to solve the problems, we provide the dynamic management of members' role and permission in application layer multicast, context-aware dynamic role based access control model is presented to solve the problem. The model is analyzed based on the definitions of user assignment and permission assignment matrix. The user and resource context are collected by U-agents and P-agents. With the information collected, the users' role and permission assignments are dynamically adapted. Finally, the dynamic role based access control model for application layer multicast is implemented, which solve the key problems and which meets our demand for safety of reality and provides a strong guarantee for the safe to us.

Keywords-ALM; RBAC; User Assignment; Permission Assignment; Context

I. INTRODUCTION

In 1988 Steve Deering proposed the IP Multicast architecture, which is the most efficient way to perform group data distribution, as it is able to reduce packet replication on the wide-area network to the minimum necessary. However, more than two decades after its initial proposal, deployment of IP Multicast has been limited and sparse due to a variety of technical and non-technical reasons. Therefore some researchers have proposed application layer multicast (short for ALM) as an alternate technique for multicasting. As the name suggests, in application layer multicast, the multicasting functionality is implemented at the application layer, i.e. at the end-hosts instead of the network routers[1,6,11,12]. A number of application layer multicast methods such as Narada, NICE, ALMI have been proposed in the literature in the recent past. Those methods mainly focus on the implement of ALM, to some extend efficiency and reliability of data transmission, only ignore the security problem. Role Based Access Control (short for RBAC) can be used to resolve the access control of ALM.

RBAC[2,4,5,9,13], which will reduce the complexity and cost of authorization managements, provides the role that is consistent with the structure in ALM. User indirectly access the resource with the assigned role. RABC which is regarded as better substitution of DAC and MAC, got extensive attention among researchers, users and software manufactures. However in the traditional RBAC, User Assignment (short for UA) and Permission Assignment (short for PA) is controlled and adjusted by system administrator without considering the dynamic change of users and resources attributes which is defined as Context.

II. KEY PROBLEMS IN ALM COMMUNICATION

In ALM approach multicast functionality is implemented at the end-hosts instead of network routers. Unlike network-layer multicast, ALM requires no infrastructure support and can be easily deployed in the Internet. Some questions need to be considered before RBAC applied in ALM system:

1. Security issue. Security issue in ALM is complex. ALM protocols do not provide permission control of members, which brings security problem. We focus on the RBAC applied in ALM system to control the permission assignment and make sure that only the legal user can join the communication.

2. Dynamic members. Members can random join or quit ALM system. At same time member attribute and network state are dynamic changing. User Assignment and Permission Assignment need to be changing to adapt this dynamic attribute.

3. Reliability. In ALM approach multicast functionality is implemented at the end-hosts. As the diversity of hosts in ALM communication, the system need to dynamically adjust the load of the hosts according to the context information.

4.data delay. In ALM approach, the data is not only transmitted through circuitry, but also throughout by the hosts. The RABC model applied in ALM need to consider the data delay. The context information contains the networks and hosts delivery delay.

In ALM approach, that user joins or quit the communication freely, data delay and user drop may cause the change of the user and communication attributes.

Literature [7] presents a RBAC Model Based on Centralized ALM to focus on the access control problem, but it lack considering the dynamic adjusting UA and PA. Literature [8，10] presents a model that dynamically changes the user assignment without permission assignment. Based the two papers, we extend a context-aware dynamic RBAC model for ALM. In the model the users' role and permission assignments are dynamically adapted.

III. A CONTEXT-AWARE RBAC MODEL

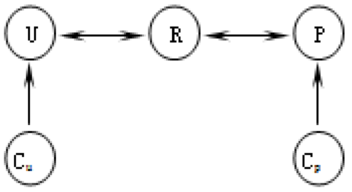

Figure 1. A Context-aware RBAC model

The definitions in our model are based on literature [3-5]. U represents a set of users. R represents a set of roles. P represents a set of permissions. S represents a set of sessions.

UA is the mappings that assign roles to users. In the session, each user is assigned a set of roles, the context information is used to decide which role is active. The user will access the resource with the active role. In other words UA⊆U×R.

PA is the mappings that assign permissions to roles. Every role that has privilege to access the resource is assigned a set of permissions, and the context information is used to decide which permission is active for that role. In other words PA⊆P×R.

The model is illustrated in Figure 1. In the model UA and PA matrix are dynamically adjusted according to context information, from which the users can be always provided with ‘fit’ privileges.

DEFINITION 1. Let C represent a set of context information, C={c1,c2,…cn},n≥1, ∀ i,j,i≠j and 1≤i,j≤n,ci≠cj. Let Cu represent a set of USERS context information and Cp represent a set of PERMISSIONS context information, we say that C=Cu∪Cp.

UA is influenced by Cu which can be login time, login IP, trust grade, balance,etc. PA is influenced by Cp which can be bandwidth, package loss rate, host performance, etc.

In the model programmer decides the set of context information to fulfil the requirements and compiles the actual context collecting codes.

The context collecting code is used to dynamically compute the value of the parameter, and its implementation is system-dependent. The dynamic mechanism to compute parameter values such as in toll TV. In such case, the balance gets important. When the money left is over, the user should not be assigned corresponding roles to receive the video data. in such environment, the balance is constantly changing and need to be re-evaluated at certain intervals. Additionally, for dynamic access constraints, such as duration, context collecting function would be called periodically to ensure that the constraint is always satisfied.

A. Static parameters

Let USER set U={u_1,u_2,…u_x}, ROLE set R={r_1,r_2,…r_y}, PERMISSION set P={p_1, p_2,…p_z}, i=1,2,…,x, j=1,2,…,y, k=1,2,…,z, x, y and z are natural numbers, we have the following definitions：

DEFINITION 2. M is UA matix, which is used to describe the state of role-to-user assignment. ∀ u_i∈U,r_j∈R, and a corresponding boolean variable m_{ij}∈{1,0}, when m_{ij}=1, USER u_i is assigned ROLE r_j and otherwise, USER u_i is not assigned ROLE r_j. We have UA matix M:

$$M=\left[m_{ij}\right]_{x\times y}=\begin{bmatrix} m_{11}, & m_{12}, & \Lambda\ , & m_{1y} \\ m_{21}, & m_{22}, & \Lambda\ , & m_{2y} \\ \Lambda & \Lambda & \Lambda & \Lambda \\ m_{x1}, & m_{x2}, & \Lambda\ , & m_{xy} \end{bmatrix}$$

DEFINITION 3. N is PA matix, which is used to describe the state of permission-to-role assignment. ∀ r_j∈R,p_k∈P, and a corresponding boolean variable n_{jk}∈{1,0}, when n_{jk}=1, ROLE r_j is assigned PERMISSION p_k and otherwise, ROLE r_j is not assigned PERMISSION p_k. We have PA matix N:

$$N=\left[n_{jk}\right]_{y\times z}=\begin{bmatrix} n_{11}, & n_{12}, & \Lambda\ , & n_{1z} \\ n_{21}, & n_{22}, & \Lambda\ , & n_{2z} \\ \Lambda & \Lambda & \Lambda & \Lambda \\ n_{y1}, & n_{y2}, & \Lambda\ , & n_{yz} \end{bmatrix}$$

THEOREM 1. According to DEFINITION 2, we have :set {r_j | m_{ij}=1} is the ROLE set assigned to u_i, set {u_i | m_{ij}=1} is the USER set that r_j is assigned to.

The ith row rank can be denoted by |{ m_{ij} | j=1,2,…,y} | whose value is the number of roles owned by u_i. The jth column rank can be denoted by | { m_{ij} | i=1,2,…,x} | whose value is the number of users r_j assigned to.

The matrix M shows the state of ROLES assignment to USERS.

THEOREM 2. According to DEFINITION 3, we have : {p_k | n_{jk}=1} is the permission set of ROLE r_j. {r_j | n_{jk}=1} is the role set p_k assigned to.

From the two sets upside we can get the number of permissions owned by r_j and the number of roles p_k assigned to.

The matrix N shows the state of PERMISSIONS assignment to ROLES.

DEFINITION 4. Let L be a U-P matrix which is the product of UA matrix and PA matrix., then

L=M×N，or

$$L=(l_{ik})_{XZ}\ l_{ik}\ =\ m_{i1}n_{1k}\ +\ m_{i2}n_{2k}\ +\Lambda\ +\ m_{iy}n_{yk}\ =\sum_{j=1}^{y} m_{ij}n_{jk}$$

L describes the state of that users own permissions in RBAC model. From U-P matrix L we can see that the set {p$_k$ | $\sum_{j=1}^{y} m_{ij}n_{jk}$ =1} is the permission set owned by u$_i$. The change of matrix L shows the change of permissions assigned to users in RBAC model.

According to THEOREM 1 and THEOREM 2, we have: when p$_k$∈{p$_k$ | n$_{jk}$=1} and r$_j$∈{r$_j$ | m$_{ij}$=1}, USER u$_i$ owns PERMISSION p$_k$.

We suppose same permission is not assigned to two or more roles owned by a same user to avoid redundancy. In other words, p$_k$∈{p$_k$ | n$_{j'k}$=1} and p$_k$∈{p$_k$ | n$_{j''k}$=1}, r$_{j'}$, r$_{j''}$∈{r$_j$ | m$_{ij}$=1}, then we have r$_{j'}$=r$_{j''}$.

THEOREM 3. Based on the condition beside, when

$\sum_{j=1}^{y} m_{ij}n_{jk} = 1,$ USER u$_i$ owns PERMISSION p$_k$,

$\sum_{j=1}^{y} m_{ij}n_{jk} = 0,$ USER u$_i$ does not own PERMISSION p$_k$,

$\sum_{j=1}^{y} m_{ij}n_{jk}\ \phi\ 1,$ Same permission is assigned to two or more roles owned by a same user. This situation should not happen according to the condition beside.

Here are two algorithms to avoid that same permission is assigned to two or more roles owned by same user.

Algorithm 1: assign PERMISSION p$_k$ to ROLE r$_a$

```
1.  procedure
     $n_{ak}$ =0   {before system assigns PERMISSION $p_k$ to ROLE $r_a$. $r_a$∈R, 1≤a≤y, a is natural number }
2.  While (1≤i≤x) {RBAC model contains x users}
     if $m_{ia}$=1{if USER $u_i$ owns $r_a$}   then s=$\sum_{j=1}^{y} m_{ij}n_{jk}$ {s is a temporary variable}
          if s=0 then $n_{ak}$ =1 { assign $p_k$ to $r_a$ }
     {if a ROLE is not owned by a USER, it will not assigned any PERMISSION}
```

Algorithm 1 shows, only when redundant assignment is avoided, can PERMISSION be assigned to ROLE.

Algorithm 2: assign ROLE r$_j$ to USER u$_a$

```
1.  procedure
     $m_{aj}$=0  {before system assigns ROLE $r_j$ to USER $u_a$}
2.  $m_{aj}$= $m_{aj}$ +1{if USER $u_a$ owns $r_j$}
While (1≤i≤x)
     s=$\sum_{j=1}^{y} m_{ij}n_{jk}$ {s is a temporary variable}
          if s＞1 then $m_{aj}$=0 { not assign $r_j$ to $u_a$ }
```

Algorithm 2 shows, if redundant assignment occur, ROLE will not be assigned to USER.

B. Dynamic parameters

DEFINITION 5. Role switch condition is SwitchR({m$_{ij}$})={m'$_{ij}$}，j=1,2,…,y. After role switch, set {r$_j$ | m'$_{ij}$ =1} is the ROLE set owned by u$_i$.

If m$_{ij}$= m'$_{ij}$, ROLE r$_j$ is still assigned to user u$_i$.

If m$_{ij}$=1，m'$_{ij}$ =0，the state of that u$_i$ owns r$_j$ changed. ROLE r$_j$ is no longer assigned to user u$_i$.

If m$_{ij}$=0，m'$_{ij}$ =1，the state of that u$_i$ owns r$_j$ changed. ROLE r$_j$ is assigned to user u$_i$.

DEFINITION 6. Permission switch condition is SwitchP({n$_{jk}$})={n'$_{jk}$}，k=1,2,…,z. After permission switch, set {p$_k$ | n'$_{jk}$ =1} is the PERMISSION set owned by r$_j$.

If n$_{jk}$= n'$_{jk}$, p$_k$ is still assigned to user r$_j$.

If n$_{jk}$=1，n'$_{jk}$=0，the state of that r$_j$ owns p$_k$ changed. p$_k$ is no longer assigned to user r$_j$.

If n$_{jk}$=0，n'$_{jk}$=1，the state of that r$_j$ owns p$_k$ changed. p$_k$ is assigned to user r$_j$.

DEFINITION 7. If two or more elements in set {m$_{ij}$ | j=1,2,…,y} change, a role muti-switch happen.

DEFINITION 8. If two or more elements in set {n$_{jk}$ | k=1,2,…,z} change, a permission muti-switch happen.

DEFINITION 9. $M',M''\Lambda\ M^{(c)}$ are UA matrices，according to ROLE SWITCH CONDITION, let $M'\rightarrow M''\rightarrow\Lambda\ \rightarrow M^{(c-1)}\rightarrow M^{(c)}$ (c is natural number, and c>1) be a switch series, then series $\{r_j|m'_{ij}=1\}\rightarrow\{r_j|m''_{ij}=1\}\rightarrow\Lambda\ \rightarrow\{r_j|m^{(c-1)}{}_{ij}=1\}\rightarrow\{r_j|m^{(c)}{}_{ij}=1\}$ is role switch chain of u$_i$.

The switch series $M'\rightarrow M''\rightarrow\Lambda\ \rightarrow M^{(c-1)}\rightarrow M^{(c)}$ shows the dynamic change of ROLE assignments to USER in RBAC. In a similar way, we have a switch series $N'\rightarrow N''\rightarrow\Lambda\ \rightarrow N^{(c-1)}\rightarrow N^{(c)}$ which shows the dynamic change of PERMISSION assignments to ROLE in RBAC.

we can define permission switch chain $\{P_k|n'_{jk}=1\}\rightarrow\{P_k|n''_{jk}=1\}\rightarrow\Lambda\ \rightarrow\{P_k|n^{(c-1)}{}_{jk}=1\}\rightarrow\{P_k|n^{(c)}{}_{jk}=1\}$.

DEFINITION 10. RBAC state can be expressed with a two-tuples (UA,PA). In dynamic RBAC model, UA matrix and PA matrix are dynamically changed by context.

THEOREM 4. If UA matrix M or PA matrix N changes, RBAC state changes, vice versa.

THEOREM 5. If U-P matrix L changes, RBAC state changes. Whereas it is not always true.

Example: let USER u own ROLE r_1 and r_2，PERMISSION p is assigned to r_1. Then PERMISSION p is switched from r_1 to r_2, RBAC state changes and U-P matrix does not change.

IV. AN IMPLEMENTATION FRAME OF DRBAC FOR ALM SYSTEM

As in figure 2, the user and resource context are collected by U-agents and P-agents. With the information collected, the users' role and permission assignments are dynamically adapted.

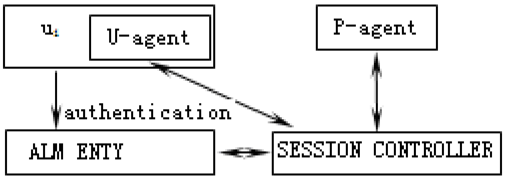

Figure 2. An implementation frame of DRBAC for ALM system

When a USER u_i login ALM system, after successfully authentication, u_i download a client program contain U-agent from SESSION CONTROLLER and install it. Then SESSION CONTROLLER assigns initialized role to u_i. With the context information collected by U-agent and P-agent, SESSION CONTROLLER changes the UA matrix M and PA matrix N of u_i.

A simple example: we have 3 users, 4 roles and 5 permissions in ALM system. The RBAC state can be expressed with M and N matrix.

$$M = \begin{array}{c|cccc} & r_1 & r_2 & r_3 & r_4 \\ \hline u_1 & 1 & 0 & 1 & 1 \\ u_2 & 0 & 0 & 1 & 0 \\ u_3 & 1 & 1 & 0 & 0 \end{array}$$

$$N = \begin{array}{c|ccccc} & p_1 & p_2 & p_3 & p_4 & p_5 \\ \hline r_1 & 0 & 0 & 1 & 1 & 0 \\ r_2 & 1 & 0 & 0 & 0 & 0 \\ r_3 & 0 & 1 & 0 & 0 & 1 \\ r_4 & 0 & 0 & 0 & 0 & 0 \end{array}$$

From M matrix, we know:
USER u1 owns ROLE r1, r3 and r4,
USER u2 owns ROLE r3,
USER u3 owns ROLE r1 and r2.
From N matrix, we know:
ROLE r1 owns PERMISSION p3 and p4,
ROLE r2 owns PERMISSION p1,
ROLE r3 owns PERMISSION p2 and p5,
ROLE r4 owns no PERMISSION.

To express the RBAC state, we calculate U-P matrix L(Mechanism must be set to avoid the element of L greater than 1):

$$L = M \times N = \begin{array}{c|ccccc} & p_1 & p_2 & p_3 & p_4 & p_5 \\ \hline u_1 & 0 & 1 & 1 & 1 & 1 \\ u_2 & 0 & 1 & 0 & 0 & 1 \\ u_3 & 1 & 0 & 1 & 1 & 0 \end{array}$$

From L matrix, we know:
USER u1 owns PERMISSION p1, p2,p3 and p4,
USER u2 owns PERMISSION p2 and p5,
USER u3 owns PERMISSION p1,p3 and p4.
In the application, M and N matrices are dynamically adjusted based on the context information collected by U-agents and P-agents.
In DRBAC implementation frame, we should pay attention to detail：

(1)The system should guarantee the accuracy of context and security of session control communication. The accuracy of context directly relates to permission assignment of u_i.

(2)Due to the dynamic of UA and PA, mechanism must be launched to avoid redundancy assignment (same permission assigned to two or more roles) or wrong assignment (assign permission p to a role, and at same time abrogate permission p from another role but owned by same user).

V. CONCLUSION

That ALM system is open to every user brings the security problem. To meet the security and dynamic permission control in ALM, the system need to real time adjust the permissions owned by users based on the context information. We present a dynamic RBAC model, with UA and PA matrix, permissions are dynamically adjusted. The matrix M maintains the state of ROLES assignment to USERS. The matrix N maintains the state of PERMISSIONS assignment to ROLES. From the UA matrix M and PA matrix N, we easily get RBAC state.

REFERENCES

[1] Yeo, C.K., Lee B S, Er M H. "A survey of application level multicast techniques."Computer Communications 27.15(2004):1547–1568.

[2] Sandhu RS, COYNE E J, FENSTEIN H L, et al. "Role-Based Access Control Models." 29.2(1999):38 - 47.

[3] Ferra Iolo D F, Sandhu RS, GAVR ILA S, et al. “Proposed NIST standard for role-based access control”. ACM Transactions on Information and System Security, 2001, 4(3):224-274.

[4] ANSI American national standard for information technology-role based access control[S], 2004.

[5] Li, Ninghui, et al. "A Critique of the ANSI Standard on Role-Based Access Control." Security & Privacy, IEEE 5.6(2007):41 - 49.

[6] Jun-sheng LI, Yu Zhenwen, Pan Yun. "A Survey of the Application-level Multicast."Application Research of Computers 21.11(2004):14-17.

[7] Yu-jiao, LIU, and LU Zheng-fu. "Research on RBAC Model Based on Centralized ALM." Journal of Mianyang Normal University (2007).

[8] Zhang, Xuewang, et al. "Study on an Improved Extended-RBAC Model.." Knowledge Discovery and Data Mining, 2009. WKDD 2009. Second International Workshop on (2009):640 - 643.

[9] Zan Yang, Lin Yang, Xiangyang Luo, Linru Ma, Baosheng Kou, and Kun Zhang, "Model of domain based RBAC and supporting technologies," Journal of Computers, 2013, 5（8）: 1220-1229,

[10] Li, Mengmeng, and Baode Fan. "The modeling of RBAC model based on UML and XACML." Systems and Informatics (ICSAI), 2012 International Conference on. IEEE, 2012:1533 - 1537.

[11] Chuah, Mooi‐Choo 1, and Yang, Peng 2. "Context-aware multicast routing scheme for Disruption Tolerant Networks." INTERNATIONAL JOURNAL OF AD HOC AND UBIQUITOUS COMPUTING 4.5 (2009):269-281.

[12] Li-jv, GUO, et al. "The Research of Application Layer DDoS Attack Detection based the Model of Human Access." Computer Security(2014).

[13] Bhattacharjee, Shiladitya, et al. "A multibit burst error detection and correction mechanism for application layer." Computer and Information Sciences (ICCOINS), 2014 International Conference on (2014):1 - 6.