Dynamic User-Role Assignment in Remote Access Control

Mohsen Saffarian[1], Qiang Tang[1], Willem Jonker[1,2], and Pieter Hartel[1]

[1] Faculty of EWI, University of Twente, the Netherlands
[2] Philips Research, the Netherlands

Abstract. The Role-Based Access Control (RBAC) model has been widely applied to a single domain in which users are known to the administrative unit of that domain, beforehand. However, the application of the conventional RBAC model for remote access control scenarios is not straightforward. In such scenarios, the access requestor is outside of the provider domain and thus, the user population is heterogeneous and dynamic. Here, the main challenge is to automatically assign users to appropriate roles of the provider domain. Trust management has been proposed as a supporting technique to solve the problem of remote access control. The key idea is to establish a mutual trust between the requestor and provider based on credentials they exchange. However, a credential doesn't convey any information about the behavior of its holder during the time it is being used. Furthermore, in terms of privileges granted to the requestor, existing trust management systems are either too restrictive or not restrictive enough. In this paper, we propose a new dynamic user-role assignment approach for remote access control, where a stranger requests for access from a provider domain. Our approach has two advantages compared to the existing dynamic user-role assignment techniques. Firstly, it addresses the principle of least privilege without degrading the efficiency of the access control system. Secondly, it takes into account both credentials and the past behavior of the requestor in such a way that he cannot compensate for the lack of necessary credentials by having a good past behavior.

1 Introduction

An access control system is a component of a multi-user system which mediates requests to resources of the system and makes decisions about whether or not they should be granted. Classical access control models , namely Discretionary Access control (DAC), Mandatory Access Control (MAC) and Role-Based Access Control (RBAC) have been mainly applied to single domains. The main characteristic of a single domain is that users are known to the administrative unit of that domain, beforehand. Among such models, RBAC has absorbed more attention, both in practice [17] and research [7]. This is due to the fact that firstly, RBAC can be well adapted to organizations where each role corresponds to a job function. Secondly, RBAC greatly simplifies the management of authorizations within a system, because a group of subjects are usually given the same permissions.

Contrary to the problem of access control in a single domain, in remote access control scenarios the access requestor is not registered in the provider domain and thus, the user population is heterogeneous and dynamic. Though the problem of access control has been an active research line for many years, we didn't find a clear description for different possible scenarios. We address this issue by presenting a systematic overview of the different possible access control scenarios. In particular, we are interested in a remote access control scenario where a stranger who is not registered in any domain requests for access from a provider domain. Despite the above-mentioned advantages, the application of RBAC for such a scenario is not straightforward. The main issue here is to dynamically assign appropriate domain-specific roles to strangers.

Trust management [2, 3, 9, 13] has been proposed as a supporting approach to solve the problem of remote access control. Here, the key idea is to establish a mutual trust between the requestor and provider based on credentials they exchange. Each credential, issued by a trusted authority, binds an attribute to its holder. Each domain regulates accesses to its resources and services by asking the requestor to present a set of credentials. Although credential based trust management systems are promising for the described remote access control scenario, they don't take into consideration the behavior of strangers using credentials. A credential doesn't give any information about the behavior of its holder during the past uses of that credential. In other words, existing trust management systems bind a binary trust to strangers regardless of their past behavior.

In order to benefit from the advantages of both RBAC and trust management systems in an open environment, a few models have been proposed [5, 10, 20]. The main issue that such models address is to automatically assign strangers to appropriate domain- specific roles, based on credentials they present. In particular, the TrustBAC model [5] tries to address the above-mentioned problem with the notion of credentials. In this model, automatic user-role assignment is based on not only credentials of a stranger but its past behavior and recommendations. However, this approach has two drawbacks. First, it doesn't respect the principle of least privilege. When a stranger sends his request for access, based on the presented credentials, his past behavior and user recommendations he is assigned a trust level which is associated with some roles. Consequently, the requestor becomes a member of all such roles and authorized to exercise all permissions associated to them. This exposes the system to the risk of granting strangers excessive permissions that they don't require for the requested transaction. The second drawback of this approach is that a user who lacks some necessary credentials to become qualified to assume a highly privileged role might still be able to become a member of that role. This may happen if an unqualified user has shown a good past behavior and/or received good recommendations.

The contribution of the paper is twofold. First, it gives a systematic overview of the different possible access control scenarios where the requestor can be inside or outside of the provider domain. The second and main contribution of the paper is that it gives a novel dynamic user-role assignment approach which addresses the identified shortcomings of the TrustBAC model. In particular, we

address the principle of least privilege by assigning a stranger the least privileged role to which the requested permission has been assigned. In addition, in our approach we require a stranger to be both qualified and trusted for the requested role (permission). In this way, a stranger cannot compensate for the lack of qualifications by having a good past behavior and/or receiving good recommendations.

The remainder of this paper is organized as follows. In Section 3 we describe different access control scenarios. In Section 4 we present our dynamic user-role assignment approach for the remote access control scenario, where a stranger requests for access from a provider domain. In Section 5 we discuss about the proposed approach. In Section 6 the state-of-the-art application of RBAC in other access control scenarios is given. Finally, Section 7 concludes the paper.

2 Related Work

In this section we represent the background work that has influenced our work. We start with representing the basic and hierarchical RBAC and then describe the dynamic user-role assignment and role-based trust management framework RT [11, 12, 14].

2.1 Role-Based Access Control

Role-based access control has been the subject of research for many years, resulting in several different models such as the RBAC96 model [18], the role graph model [16] and the ANSI RBAC standard model [19]. At the core of all these models is the concept of role, which connects a set of users to a set of permissions, in a way that users assigned to a role acquire permissions of that role. The main motivation behind RBAC is that it greatly simplifies the management of authorizations within a system, due to the fact that a group of subjects are usually given the same permissions. More elaborated RBAC models utilize the concept of role hierarchy, through which membership inheritance and permission inheritance are added to the basic model. In the hierarchical RBAC models, the role hierarchy is a partial order, in which a more senior role inherits all permissions associated to its junior roles. Furthermore, each member of a more senior role is also a member of all of its junior roles. Therefore, hierarchical RBAC even further simplifies the management of authorizations by reducing the number of explicit user-role and permission-role assignments. In the rest of the paper, we confine the interpretation of RBAC to the ANSI RBAC standard model.

2.2 Dynamic User-Role Assignment

There are a few existing techniques which address the problem of dynamic user-role assignment [1, 4, 10, 11, 12]. Such techniques try to automate the user-role assignment administrative task, which is motivated by the following two scenarios. The first scenario is a large-scale organization where the number of users

can be in the hundreds of thousands or millions. In such an environment, the manual user-role assignment becomes a cumbersome task. The second scenario is an open distributed system where the user population is dynamic and the identity of all users is unknown beforehand. In such systems, the manual user-role assignment is almost impossible. In this paper, we are dealing with the latter scenario.

In a single domain of control, dynamic user-role assignment is based on the policies defined by the administrative unit of that domain. In order to assign an appropriate role to a user, such policies require different information of the user which depends on the applied dynamic user-role assignment model. In [1] Al-Kahtani et al. proposed the Rule-Based RBAC or RB-RBAC. In this model, the administrative unit of an enterprise, defines a set of rules to automatically assign users to roles. Such rules, take into consideration two elements: the credentials of the user and the constraints on using roles. However, this model takes into account the trustworthiness of neither the credential issuers nor the users regarding their past behavior.

This is also the case for the existing credential-based trust management systems which try to dynamically assign roles to strangers [4, 10, 11, 12]. A credential may be issued by different authorities. In addition, a credential doesn't give any information about the behavior of its holder during the past uses of that credential. This issue has been addressed to some extent in the TrustBAC model [5] in which dynamic user-role assignment is based on not only credentials of a stranger but its past behavior and recommendations. Here, users are assigned to trust levels which are assigned to roles. The trust level of a user is obtained by the sum of values extracted from separate trust components for that user, e.g. his credentials, past behavior in the system and recommendations from other users. However, as we discussed in the previous section, this model has two drawbacks. First, it doesn't respect the principle of least privilege. Second, a stranger who lacks some necessary credentials to become qualified to assume a highly privileged role might compensate for those credentials by having a good past behavior and/or receiving good recommendations.

2.3 Role-Based Trust Management Framework (RT)

The main problem that we try to shed light on is also a variant of trust management system in which a stranger sends its request for access to a provider domain which applies Role-Based Access Control model. To do so, we apply the Role-Based Trust Management Framework (RT). The reason that we choose RT is that it is one of the most influential and successful credential based trust management systems [6]. RT includes a family of Role Based Trust Management (TM) languages which RT_0 is the simplest member and RT_1, RT_2, RT_3, RT^T, RT^D are the more sophisticated ones. In the following, we briefly describe RT_0 which we use in our dynamic user-role assignment approach.

The main constructs of RT_0 are entities, role names and roles. An entity, denoted by a name starting with an uppercase letter, can be an individual like Alice, or a domain of control like HospitalA. An entity can define roles, issue

credentials and make access requests. A role name is denoted by a name starting with a lowercase letter e.g. doctor, nurse and physician are role names. The concept of role is at the core of RT_0 and is identified by an entity followed by a role name, separated by a dot. For example, HospitalA.doctor is a valid role, indication that HospitalA is the owner of role doctor and is the only authority who can directly determine its members. Members of a role acquire the permissions associated to that role. A permission is also represented by a role in RT_0 e.g. HospitalA.readMedicalRecord which is the permission to read a medical record of hospital A. In RT_0, there are four types of credentials that entity A can issue. For each credential type, the membership of role A.r is defined in a different way.

- **Simple Member:** $A.r \leftarrow D$.
  This credential means A states that D is a member of $A.r$
- **Simple Inclusion:** $A.r \leftarrow B.r_1$.
  This credential means all members of $B.r_1$ are also members of $A.r$. This represents a delegation from A to B.
- **Linking Inclusion:** $A.r \leftarrow A.r_1.r_2$.
  This credential means $A.r_1$ includes $B.r_2$ for every B that is a member of $A.r_1$. This shows a delegation from A to all members of $A.r_1$.
- **Intersection Inclusion:** $A.r \leftarrow B_1.r_1 \cap B_2.r_2$.
  This credential means A states that $A.r$ includes any entity who is a member of both $B_1.r_1$ and $B_2.r_2$.

It should be noted that a policy is a finite set of credentials of the above form. For the complete introduction of RT_0 we refer the reader to [6].

3 The Scenarios

In this section, we present four different access control scenarios. We start with entities and assumptions on which the scenarios rely. We then continue with brief description of each scenario.

Entities

- Access Control Point (ACP)- The system entity that is in charge of the whole access control process, including definition of policies, evaluation of policies and enforcing the authorization decision.
- Past Behavior Authority (PBA)- The system entity that, in a specific application context, issues certificates of the past behavior for both subjects and domains.
- Attribute Authority (AA)- The system entity that issues attribute certificates for both individual subjects and domains.

Assumptions

- Each domain regulates accesses to its own resources using the hierarchical RBAC model, in which user-role assignment can be automatically performed based on the policies defined in RT_0.

– There are attribute authorities and past behavior authorities which, for both individual subjects and domains, issue attribute certificates and certificates of their past behavior, respectively.

Depending on the relationship between the requestor and the provider, there are four different access control scenarios which are depicted in Fig. 1 and described in the following.

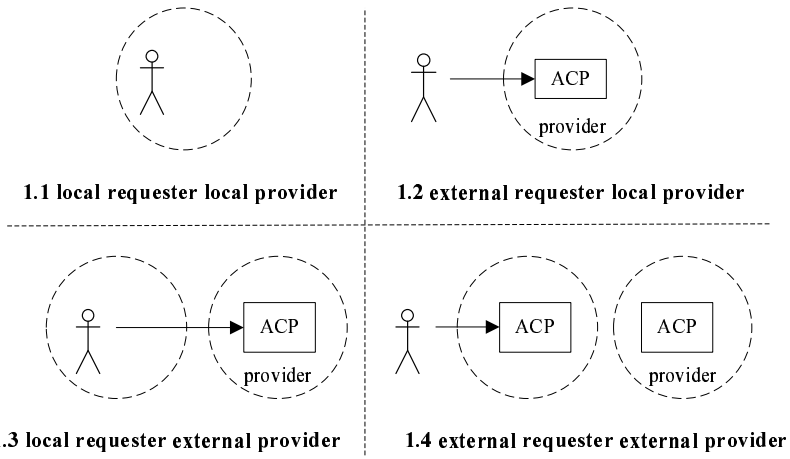

Fig. 1. Different access control scenarios

– **Local Requestor Local Provider**
The access requestor is registered in the provider domain.
– **External Requestor Local Provider**
The access requestor is registered neither in the provider domain nor in any other domain.
– **Local Requestor External Provider**
The access requestor is registered in a domain different from the provider domain.
– **External Requestor External Provider**
The access requestor who is not registered in any domain requests from domain D_1 for a permission provided by domain D_2.

In Section 4, we focus on the second scenario namely, external requestor local provider. This is because other scenarios are either straightforward or can be transformed into the second scenario. Later, we will briefly describe what is the state-of-the-art application of RBAC in other scenarios.

4 Dynamic User-Role Assignment in the External Requestor Local Provider Scenario

This scenario deals with the problem of remote access control where a stranger requests for access from the provider domain. Hence, we apply trust management techniques to issue the requestor a domain-specific timed credential. Using such a credential, the requestor is assigned to an appropriate role in the provider domain for a specific time interval. In the following, we describe the assignment process in detail. It is worth mentioning that in a real scenario the requestor has the right of choice between different providers. Consequently, the requestor may also ask the provider to present some credentials. The architecture of this scenario is depicted in Fig. 2.

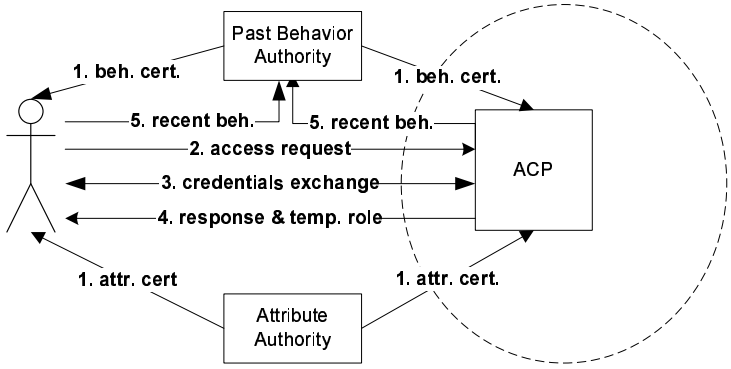

Fig. 2. External user local provider

4.1 Data-Flow Model

1. AAs and PBAs issue attribute certificates and past behavior certificates, respectively and make them available to both the access requestor and the provider domain.
2. The access requestor sends a request for access to the ACP.
3. The ACP and the requestor exchange the required credentials. To do so, The ACP traverses the hierarchical role graph in the provider domain, looking for the least privileged roles to which the requested permission has been assigned. Thus, The applied search algorithm is bottom-up Breadth-First Search (BFS). When one of such roles is found, the ACP asks the requestor to send the required credentials according to the corresponding user-role assignment policy defined in RT. Among such credentials is also the past behavior certificate of the requestor in the corresponding context. If such certificates satisfy all conditions encapsulated in the policy, the ACP issues

the requestor a domain- specific timed credential. Such a credential, defined in definition 2, indicates that the requestor is the holder of the corresponding role for a specific time interval. Otherwise, the ACP continues the search process in the role graph to find the next least privileged role to which the requested permission has been assigned. In case that there is no more roles left, the access request is rejected. It is worth mentioning that the ACP also sends certificates which may be asked for by the requestor. In the following, we also give the pseudo code for the operations performed by the ACP.

Definition 1. *Let T denote a time interval of type $[t_1, t_2]$, where $t_1, t_2 \in \mathbb{R}$ and $t_1 \leq t_2$.*

Definition 2. *A domain-specific timed credential is an expression of the form $c\ [T]$, where c is an RT credential of the $SimpleMember$ type and T is a time interval.*

pseudo code for the operations performed by the ACP:

$eaten - list = \{\}$, $eating - list = \{leafnodes\}$
Until $eaten - list = vertices - list$
...remove first v and add to $eaten - list$
...IF the requested permission has been assigned to v
.....add all parents and grand parents of v to $eaten - list$
.....ask the requestor to present the required credentials
.....IF presented credentials satisfy the corresponding policy
.....assign v to the requestor
.....Break
...ELSE
.....add parents of v at the end of $eating - list$
END UNTIL

4. The ACP returns the decision response and issued credential, if there is any, to the requestor. The requestor can use this credential for the next access requests.
5. Both the requestor and the provider domain (ACP) inform the PBA about the behavior of the other party in the recent transaction. Based on its local policies, The PBA uses such information to update the past behavior certificates of the involved parties.

Example 1. Assume an external access requestor *Bob* requests to read the history of the diabetes of patient p_1 for which he is receiving cure from hospital A. The role hierarchy of hospital A is depicted in Fig. 3. In this hierarchy role $nurse$ is allowed to read the general healthcare information of every patient's health record. Roles $primaryCarePhysician$ and $highlyQualifiedNurse$ are authorized to read the history of diseases of every patient's health record for which

he has received healthcare from hospital A. We have the following set of credentials (HAB is HospitalAccreditationBoard, MPB is MedicalProfessionBoard and MBA is MedicalBehaviorAuthority):

(1)$HAB.accredited \leftarrow HospitalB$
(2)$HospitalB.experienced \leftarrow Bob$
(3)$MPB.doctor \leftarrow Bob$
(4)$MBA.highTrust \leftarrow Bob$

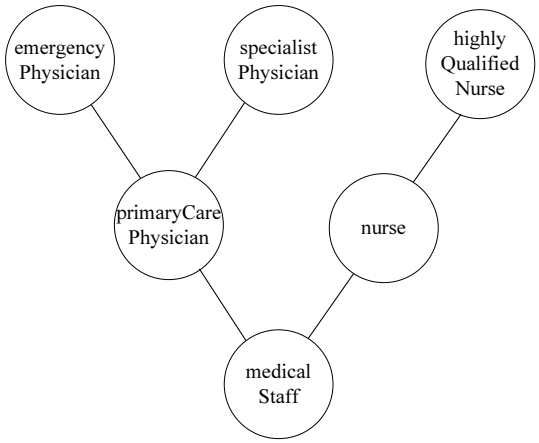

Fig. 3. Role hierarchy in hospital A

When the ACP in hospital A receives the request for access it starts traversing the role hierarchy, looking for the least privileged roles to which the requested permission has been assigned. The search algorithm is bottom-up Breadth-First Search (BFS). In this case, the first role which is found is $primaryCarePhysician$. Assume membership in such a role is determined by the following policy:

$HospitalA.primaryCarePhysician \leftarrow MPB.doctor \cap$
$HAB.accredited.experienced \cap MBA.highTrust$

Thus, the ACP asks Bob to present the required credentials for role $primary-CarePhysician$. When Bob presents all required credentials the ACP matches them to those in the policy. If all required credentials have been presented, the ACP issues credential $(HospitalA.primaryCarePhysician \leftarrow Bob)[T]$. Otherwise, the ACP repeats the procedure for the next role found in the search process namely, $highlyQualifiedNurse$. The time interval for which a domain-specific timed credential is issued particularly depends on the amount of trust hospital A puts on access requestor Bob. As long as this credential is valid Bob can present

it for his next requests for access. At the end of this transaction, both *Bob* and hospital *A* inform the MBA about the recent behavior of the other party.

5 Discussion of the Proposed Dynamic User-Role Assignment

In this section, we discuss the proposed dynamic user-role assignment approach. In particular, we describe the advantages of the proposed approach compared to the TrustBAC model.

5.1 Principle of Least Privilege

As we mentioned earlier, in the TrustBAC model strangers are assigned to trust levels which are assigned to roles. Consequently, a stranger with a certain trust level becomes a member of all roles assigned to that trust level. This exposes the system to the risk of granting strangers excessive permissions that they don't require for the requested permission. A restrictive approach to address this issue is not to assign a role to a stranger, but decide about each access request separately. However, this approach highly degrades the efficiency of the access control system, as for each request the whole credential exchange phase should be repeated. Therefore, we apply the open world policy model [8] to temporarily assign the requestor an appropriate role. This means whenever there is not any restricting policy, for a specific time interval, the authorized requestor is assigned to the least privileged role to which the requested permission has been assigned. This is implemented through a notion of domain-specific timed credential defined in the previous section. Such a credential can then be used by its holder for the next access requests. It is worth mentioning that in this way even if the requested permission is not associated with the presented domain-specific timed credential, such a credential can simplify both the search process in the role graph and the credential exchange phase. The search process is simplified because the requested permission is not associated with any role beneath the one encapsulated in the presented credential. The credential exchange phase may also be simplified due to the fact that the requestor is not asked to resend the credentials that he has already sent to be assigned to the presented domain-specific timed credential.

5.2 Requiring the User to Be Both Qualified and Trusted

In our approach we require the requestor to be both qualified and trusted for the requested permission (role). This is achieved by composing each user-role assignment policy in the following form:

$$A.r \leftarrow (c_1 \odot c_2 \odot .. \odot c_n) \wedge PBA.trustLevel \text{ where } \odot \text{ is logical And/Or.}$$

In this way, the requestor has to present both a set of credentials issued by different attribute authorities (entities) and the past behavior certificate issued

by the past behavior authority in the relevant context. It is worth mentioning that past behavior authorities apply their own policies to update the trust level of both domains and individuals. A past behavior authority in a specific context updates each party's trust level either at specific time intervals or after certain number of transactions.

5.3 Choosing between Different Providers

As we described in the data-flow model of our approach, the access requestor has the right to choose between different providers. We make this possible by letting the requestor to ask a provider to send the required credentials, e.g. its past behavior certificate. If during a running session, the provider domain cannot meet the criteria of the requestor, the latter can terminate the session and try another provider domain.

6 Other Access Control Scenarios

In this section we briefly describe the state-of-the- art application of RBAC in other access control scenarios.

6.1 Local Requestor Local Provider

This scenario deals with the problem of access control in a single domain. Here, the access requestor is registered in the provider domain and thus, before sending his request for access he should be properly authenticated. The architecture of the scenario is depicted in Fig. 4.

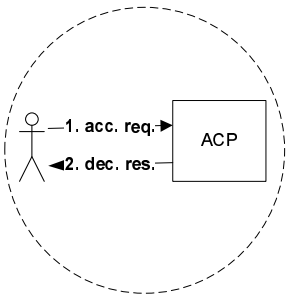

Fig. 4. Local user local provider

Data-Flow Model

1. The access requestor sends a request for access to the ACP.

2. The ACP evaluates the request against applicable local RBAC policies and returns the authorization decision to the requestor.

Although in the above scenario the user-role assignment can be performed automatically, it is more intuitive to use user-role assignment commands issued by administrators. What makes this scenario different from the external user local provider scenario is that, regardless of the applied user-role assignment mechanism, users are registered and identified in the provider domain. Consequently, whenever a new user is assigned to a role, it is considered both qualified and trusted to hold that role and perform its associated permissions. However, based on its behavior in the system, each user's qualifications and trust may change in the course of time. In order to exercise permissions for which a user is authorized, it should first be authenticated. Hence, each single activity performed by a user can be logged in the system. Such logs can then be used by the administrative unit to change the user's qualifications and/or adjust the amount of trust given to that user, which may result in the change of his assumed roles.

Example 2. Consider the role hierarchy of hospital A depicted in Fig. 3. When *Alice* is assigned any role senior to role $primaryCarePhysician$ she is authorized to read both the general information and the history of diseases of every patient's health record for which he has received healthcare from hospital A. To do so, she has to first authenticate herself and then activate one of such roles in a running session with the ACP.

6.2 Local Requestor External Provider

This scenario also deals with the problem of remote access control. However, in contrast to the scenario described in Sect. 4, here the requestor is registered in a domain which has a sort of agreement with the provider domain. This means that the requestor is already a member of some roles in a domain which is recognized by the provider domain. Here, an existing technique is to use a role mapping table which maps the requestor's role to a role in the provider domain [15]. The architecture of the scenario is depicted in Fig. 5.

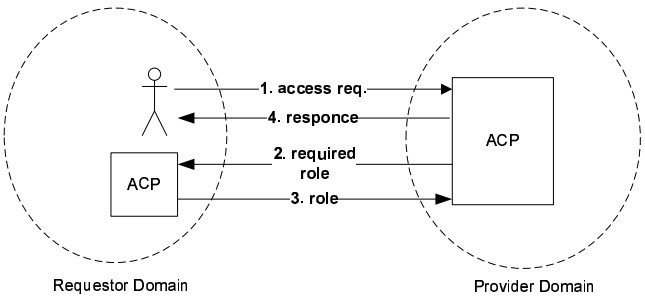

Fig. 5. Local user external provider

Data-Flow Model

1. The access requestor sends the request for access to the ACP of the provider domain.
2. The ACP of the provider domain asks for one of the required roles in the requestor domain that the requestor must be a member of. The ACP finds such roles by traversing its hierarchical role graph, looking for the least privileged roles to which the requested permission has been assigned. When one of such roles is found, the ACP looks up its role mapping table to determine the corresponding role(s) in the requestor domain. If the requested permission is associated with none of the roles in the provider domain the request is rejected.
3. The ACP of the requestor domain returns one of the roles asked by the ACP of the provider domain.
4. The ACP returns the decision response to the access requestor.

Example 3. Assume an access requestor *Bob* registered in hospital B requests to read brain MRI images of patient p_2 taken in hospital A. *Bob* has been assigned role *emergencyPhysician* in the role hierarchy of hospital B, depicted in Fig. 6. Based on the local policies of hospital A this permission is associated to roles *emergencyPhysician* and *specialistPhysician* (Fig. 3). Assume that Table 1 shows a part of role mapping table in hospital A. The ACP of hospital A searches this table looking for the roles in hospital B which correspond to *emergencyPhysician*. Hence, the ACP of hospital A asks the ACP of hospital B to send credential $HospitalB.emergencyPhysician$ for *Bob*. The ACP of hospital B then sends credential $HospitalB.emergencyPhysician \leftarrow Bob$ which results in granting access to *Bob*. If *Bob* is not a member of role $HospitalB.emergencyPhysician$ the ACP of hospital A asks for the role corresponding to *specialistPhysician* which is $HospitalB.surgeon$.

Table 1. Role mapping table in hospital A

External Role	External Domain	Local Role
$nurse$	$HospitalB$	$nurse$
$physician$	$HospitalB$	$primaryCarePhysician$
$headNurse$	$HospitalB$	$highlyQualifiedNurse$
$emergencyPhysician$	$HospitalB$	$emergencyPhysician$
$surgeon$	$HospitalB$	$specialistPhysician$

6.3 External Requestor External Provider

This scenario is similar to the one described in Sect. 4, external user local provider. However, here an external user asks a local provider to access a resource which belongs to another domain. Consequently, the first domain D_1

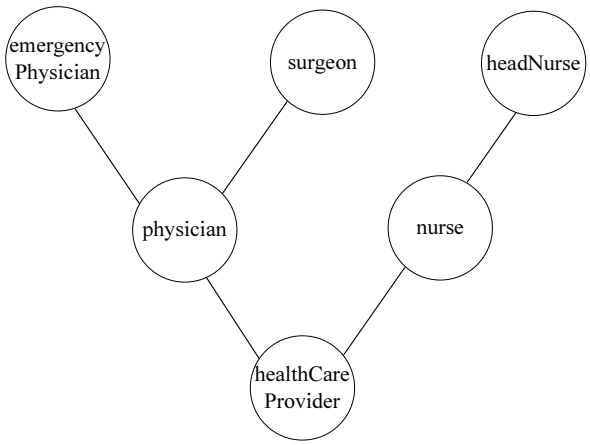

Fig. 6. Role hierarchy in hospital B

asks the requestor to send his request to the actual provider domain D_2 which is then handled in the same way as the one presented in example 1.

Example 4. Assume an external access requestor Bob sends his request to hospital B to read the history of the diabetes of patient p_1 for which he is receiving cure from hospital A. In this case, hospital B, tells Bob to send his request to hospital A, which is then processed in the same way as the one given in Sect. 4.

7 Conclusion

In this paper first we give a systematic overview of the different possible access control scenarios where the requester can be inside or outside of the provider domain. Then, integrating RBAC with RT, we present a new automatic user-role assignment approach for the scenario where a stranger requests for access from a provider domain. Our proposed approach has two main advantages compared to the existing dynamic user-role assignment techniques. First, we introduce and address the principle of least privilege. Second, we assign a stranger to a role in such a way that he must be both qualified and trusted for that role.

References

[1] M. Al-Kahtani and R. Sandhu. A model for attribute-based user-role assignment. *Computer Security Applications Conference, 2002. Proceedings. 18th Annual*, pages 353–362, 2002.

[2] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The KeyNote Trust-Management System Version 2, 1999.

[3] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. Technical report, 1996.

[4] D. Chadwick, A. Otenko, and E. Ball. Role-based access control with x.509 attribute certificates. *Internet Computing, IEEE*, 7(2):62–69, Mar/Apr 2003.

[5] S. Chakraborty and I. Ray. TrustBAC: integrating trust relationships into the RBAC model for access control in open systems. In *SACMAT '06: Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 49–58, New York, NY, USA, 2006. ACM.

[6] M. Czenko, S. Etalle, D. Li, and W. H. Winsborough. An Introduction to the Role Based Trust Management Framework RT. In A. Aldini and R. Gorrieri, editors, *FOSAD*, volume 4677 of *Lecture Notes in Computer Science*, pages 246–281. Springer, 2007.

[7] R. C. D. Ferraiolo and R. Kuhn. *Role-based Access Control*. Artech House, Apr, 2003.

[8] S. De Capitani di Vimercati, S. Foresti, and P. Samarati. Recent advances in access control. In M. Gertz and S. Jajodia, editors, *Handbook of Database Security: Applications and Trends*. Springer-Verlag, 2008.

[9] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI Certificate Theory, 1999.

[10] A. Herzberg, Y. Mass, J. Michaeli, Y. Ravid, and D. Naor. Access control meets public key infrastructure, or: Assigning roles to strangers. In *SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy*, page 2, Washington, DC, USA, 2000. IEEE Computer Society.

[11] N. Li and J. Mitchell. RT: a Role-based Trust-management framework. volume 1, pages 201–212 vol.1, April 2003.

[12] N. Li, J. Mitchell, and W. Winsborough. Design of a role-based trust-management framework. pages 114–130, 2002.

[13] N. Li and J. C. Mitchell. Datalog with constraints: A foundation for trust management languages. In *PADL '03: Proceedings of the 5th International Symposium on Practical Aspects of Declarative Languages*, pages 58–73, London, UK, 2003. Springer-Verlag.

[14] N. Li, W. H. Winsborough, and J. C. Mitchell. Distributed credential chain discovery in trust management. *Journal of Computer Security*, 11(1):35–86, Feb. 2003.

[15] L. Martino, Q. Ni, D. Lin, and E. Bertino. Multi-domain and privacy-aware role based access control in ehealth. pages 131–134, 30 2008-Feb. 1 2008.

[16] M. Nyanchama and S. Osborn. The role graph model and conflict of interest. *ACM Trans. Inf. Syst. Secur.*, 2(1):3–33, 1999.

[17] R. Ramaswamy and R. S. Role-based access control features in commercial database management systems. In *In Proceedings of 21st NIST-NCSC National Information Systems Security Conference*, pages 503–511, 1998.

[18] Ravi S. Sandhu and Edward J. Coyne and Hal L. Feinstein and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

[19] R. Sandhu, D. Ferraiolo, and R. Kuhn. The nist model for role-based access control: towards a unified standard. In *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*, pages 47–63, New York, NY, USA, 2000. ACM.

[20] B. B. Yuhui Zhong and M. Mahoui. Trustworthiness based authorization on www. Published in IEEE workshop on.