VNU Journal of Science: Comp. Science & Com. Eng, Vol. 34, No. 1 (2018) 19-29

VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

An Authorisation Policy Management Model in Federations

Vu Ngoc Cham[*], Nguyen Tuan Anh

Electric Power University, Hanoi, Vietnam

Abstract

A federation is usually an alliance of organisations where users from one organisation are trusted to access resources in another organisation. The membership of federations is diverse and continually changing. Federations require distributed and dynamic security policy management to meet these challenges. We propose an authorisation policy management model, FABACD, which simplifies the management of collaborations between organisations. It allows distributed and trusted administrators to adjust the authorisation policies in a resource holding organisation, whilst ensuring that the latter remains in ultimate control. The net result is that a resource's authorisation system is able to use user credentials built from preexisting attributes issued by any participating organisation, in order to determine a user's access rights to the various resources, without requiring credentials to be issued that are based on federation specific attributes. The model significantly simplifies the authorisation management process for the resource holding organisation.

Received 12 April 2018, Revised 18 June 2018, Accepted 18 June 2018

Keywords: Authorisation management, collaboration management, federations, distributed workflows, ABAC.

1. Introduction

A federation is an alliance of autonomous, diverse, and geographically dispersed organisations, where the participants pool resources, information and knowledge in order to meet common objectives. Since the membership of both is diverse, widely distributed, and subject to change, they require dynamic, distributed, security policy management in order to provide responsive authorisation management.

Authorisation management mechanisms are used for controlling users' permissions in federations. When users in one organisation (the issuing organisation) request access to some sensitive information in another (target) organisation, the authorisation system of the target organisation either needs to issue its own new credentials to these new users, or needs to be configured to recognise the issuing organisation and its credentials (and the attributes embedded in them), so that users with those credentials can gain access to its protected resources. In the latter case, the target organisation's authorisation system must validate and understand the meaning of the credentials issued by the issuing organisation, and may need to map them into their equivalent local ones, in order to allow them to grant access to its information and resources. In the general case, there may be many issuing organisations in a federation, in which case

[*] Corresponding author. Email: chamvn@epu.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.174

each target organisation may need to either issue new credentials to many different sets of users or recognise many collaborating organisations and the credentials they issue. A solution typically employed in federations today is to standardise a new set of federation specific attributes, and to require the issuing organisations to assign these to its users, and the target organisations to accept these from remote users. This solution employs the first approach, and is fully supported by the ANSI RBAC and ANSI ABAC models [1, 2]. However, these models do not separate issuing organisations from target organisations, issuing domains from target domains within one organisation and mapping of roles or attributes. Moreover, they do not provide any mechanisms to administer RBAC itself [3]. Therefore, these models in their current forms are not applicable to the second approach.

We believe it is more scalable and more manageable for a target organisation to reconfigure its existing authorisation system to accept credentials issued by other organisations that are based on its existing attributes, rather than to issue and revoke its own credentials to all groups of users. It is also the most cost effective solution for issuing organisations since their work is minimal. Consequently, our model is based on this paradigm. We call the mechanism that controls the way that the authorisation system of a target organisation is reconfigured to recognise the validity of users' credentials issued by other organisations the federated ABAC administrative model (FABACD).

Basically, the research work in this paper is to separate the concepts, issues and relations identified in our previous research [4], and to generalise and formalise those concepts, issues and relations. FABACD comprises two models: the federated attribute based access control (FABAC) model [5] and the FABACD model. In the first model, the separate administrative domains are recognised and modelled. While in the second model of FABACD, we are heading to the decentralization of resources administration between collabrated organisations without fading the original objects. Thank to our proposed model of FABACD model, administrators are granted limited permissions which enable them to modify components of the target organisation's FABAC policies. That enables users gain access rights to the protected resources in the target domain with minimal effort by the SoA even when their organizational attributes are quite discrepant.

As we are proceeding our implementation, this paper reports on our FABACD model's formalization and is organised as follows. Section 2 and section 3 introduce the FABAC and FABACD models respectively. Section 4 presents our empirical implementation of the models. Section 5 reviews related works whilst section 6 presents our discussion and concludes the paper.

2. The federated ABAC model

In this section we briefly present the FABAC model [5] which grants users access to federated resources based on their organisational attributes. This is shown pictorially in Figure 1.

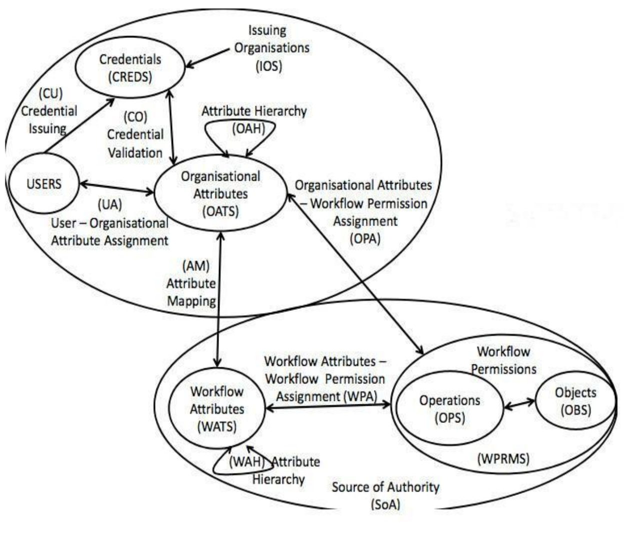
Issuing Organisations (IOS)
Credentials (CREDS)
(CU) Credential Issuing
(CO) Credential Validation
Attribute Hierarchy (OAH)
USERS
Organisational Attributes (OATS)
(UA) User – Organisational Attribute Assignment
Organisational Attributes – Workflow Permission Assignment (OPA)
(AM) Attribute Mapping
Workflow Attributes – Workflow Permission Assignment (WPA)
Workflow Permissions
Workflow Attributes (WATS)
Operations (OPS)
Objects (OBS)
(WPRMS)
(WAH) Attribute Hierarchy
Source of Authority (SoA)

Figure 1. The Federated ABAC Model

(FABAC).

In each target organisation which owns resources that are to be shared in a federation, there is an administrator who is trusted by the organisation to set the authorisation policy for the authorisation system controlling access to those resources. He is called the Source of Authority (SoA). The authorisation policy comprises of an access control policy, which is either a set of workflow permission-workflow attribute assignments (*WPA*) and attribute mappings (*AM*), or a set of workflow permission-organisational attribute assignments (*OPA*), or both; and a credential validation policy which contains the rules for validating a user's presented credentials (*CO*). Essentially, these are the rules for validating user-organisational attribute assignments (*UA*) in a distributed system. It specifies which issuing organisations are trusted to issue which kinds of organisational attributes to which users [6].

In our model, we recognise two types of user attribute: workflow attributes and organisational attributes. Both can form attribute hierarchies in a similar way to role hierarchies in RBAC [1]. Workflow attributes are assigned to workflow permissions by the SoA. Organisational attributes are assigned to users by various issuing organisations. In this way, application-level (workflow) security infrastructures are separated from organisational level security infrastructures [7]. Furthermore, in our model, credentials, which are attribute assertions digitally signed by the issuing organisation, take the place of sessions from the ANSI standard RBAC model [1]. Users are assigned credentials by the issuing organisation, usually when they start a session. Users present a subset of their credentials to the target domain when they start a session, and these contain a subset of the users' organisational attributes. Different sets of credentials can be presented in different sessions.

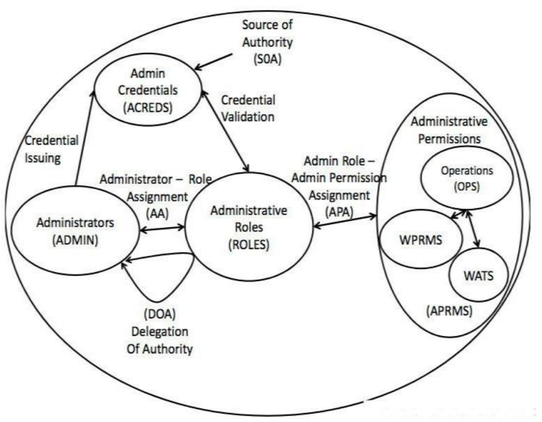
Source of Authority (SOA)
Admin Credentials (ACREDS)
Credential Validation
Credential Issuing
Administrative Permissions
Operations (OPS)
Administrator – Role Assignment (AA)
Admin Role – Admin Permission Assignment (APA)
Administrators (ADMIN)
Administrative Roles (ROLES)
WPRMS
WATS
(APRMS)
(DOA) Delegation Of Authority

Figure 2. The Federated ABAC Administrative Model (FABACD).

Workflow permissions are assigned to a user's organisational attributes only at the time the user accesses the target resource of the application/workflow. The user's credentials are validated in the target system using the credential validation policy and the valid credentials assign a subset of the user's organisational attributes to the user. Workflow permissions are assigned to the user's organisational attributes in one of two ways: permission attribute assignments (*OPA*), in which the workflow permissions are assigned to the organisational attributes or attribute mappings (*AM*), in which the organisational attributes are mapped into workflow attributes which already have workflow permissions assigned to them (*WPA*).

3. The federated ABAC administrative model

The administrative model grants remote administrators rights to update the target's authorisation policy in order to determine which users under their control should have which access rights to the target's federated resources. Without this model, remote administrators either have no permissions to update the target's authorisation policy, or they need to be added as local (target) SoAs in order to be able to assign target workflow attributes to remote organisational atrributes (remote users). In both cases, fine-grained control over the target workflow attributes would not be achieved.

An administrative permission is a consent (for an administrator) to perform either organisational attribute workflow permission assignments (*OPA*) that is to assign one or more workflow permissions to a set of (one or more) organisational attributes, or to perform attribute mappings (*AM*) that is to map a set of organisational attributes to a set of workflow attributes. When specifying the administrative permissions only the operations (such as assign permission or add mapping) and the workflow permissions (*WPRMS*) or workflow attributes (*WATS*) respectively need to be specified, since the organisational attributes (*OATS*) are not constrained by the model. The administrator is free to specify any *OATS*, whether new or existing. This is shown in Figure 2.

The authorisation policy in the target domain needs to be updated with these administrative permissions by the SoA. These administrative permissions are then assigned to various administrative roles (*APA*) by the SoA. Finally, administrators are created by the SoA assigning administrative roles to users from the federation (AA). This allows an administrator to dynamically update limited parts of the authorisation policy in the target domain, specifically, to either assign a subset of the workflow permissions (*WPRMS*) to organisational attributes or to map the latter into workflow attributes (*WATS*), and to provide the user credential validation policy (*CO*) for validating the user credentials that contain these organisational attributes. In this way, the administrator can grant controlled access to the workflow resources in the target domain to a set of users which are under his control and have the appropriate credentials, as he best determines. This mechanism further ensures that central control (by the SoA) is not lost over which workflow permissions can be administered by which administrative roles, and that the authorisation policy does not drift away from its original objectives, as no new workflow permissions can be created by anyone except the SoA.

If an administrator holds administrative permission over a workflow permission for example. “read file X” then he is authorised to assign that permission to either a single organisational attribute, for example. “employer=Org X”, or to a combination of organisational attributes for example. “employer=Org X” and “member Of Collaboration=A”. In attribute mapping, the administrator can map one or more organisational attributes into one or more workflow attributes. It is a many to many mapping relationship. For example, if “read file X” is assigned to the combination of workflow attributes “role=Y” and “User=Assigned” then the administrator can map this attribute pair into any set of organisational attributes.

The set of workflow permissions that an administrative role can assign to organisational attributes either directly or indirectly is called its administrative scope. The SoA’s administrative scope is all the workflow permissions under his control in the target domain. If an SoA’s administrative scope changes, this will usually require an update to the authorisation policy to adapt to the new set of workflow permissions. The SoA may delegate a subset of his administrative scope to administrators in the same or other domains in order to decentralise authorisation administration. This is achieved by the SoA defining a set of administrative roles (*ROLES*) which each control a subset of his administrative scope. The SoA can determine whether each administrative role may directly assign workflow permissions (*WPRMS*) to organisational attributes, or indirectly assign them via mapping a subset of the workflow attributes (*WATS*) into organisational attributes. The SoA may assign these administrative roles to any administrators on demand as the need arises, so that the administrators can control subsets of the workflow permissions. Note that our concept of administrative scope is different from the concept of administrative scope in [8]. In [8], the administrative scope of a role is defined in terms of a role hierarchy and changes dynamically as the hierarchy changes. In our model, the administrative scope is defined as a set of administrative permissions, which is always independent of the organisational attribute hierarchy, and also of the workflow attribute hierarchy if the administrative permissions are derived directly from the workflow permissions set (that is for *OPA*). If they are derived from the workflow attributes set (that is for *AM*) then the administrative scope may change if the workflow attribute hierarchy changes.

Administrators are allowed to delegate their administrative roles to other people at the discretion of the SoA. Administrative roles can therefore be delegated to other administrators, recursively, thereby adding administrators to the system. When an administrator starts a session, he is assigned (or selects) a subset of his administrative credentials. These credentials are validated in the target system using the SoA's credential validation policy (*CO*). Valid credentials assign a subset of the administrator's roles to the administrator, from which the administrative permissions are inherited. This fixes the administrator's administrative scope for the session.

Formally we have:

- $APRMS = 2^{OPS \times (WPRMS \cup WATS)}$, the set of administrative permissions - that is, every permission is a set of pairs consisting of an operation plus one of the two kinds of assignments.
- *ADMIN and ROLES* (administrators and ad-ministrative roles respectively).
- $AA \subseteq ADMIN \times ROLES \times N_\infty$, a many-to-many mapping administrator-to-administrative role assignment relation, where each administrator has been given a maximum delegation level *n* for each role. A value of zero means no delegation capability. A value of infinity means unbounded delegation capability.
- *Assigned-administrators:* $(r : ROLES) \rightarrow 2^{ADMIN}$, the mapping of administrative roles onto a set of administrators. Formally, $assigned\text{-}administrators(r) = \{a \in ADMIN \mid (a,r,n) \in AA\}$.
- $APA \subseteq APRMS \times ROLES$, a many-to-many mapping adminstrative permission-to-administrative role assignment relation.
- *FAA* which is the complete set of adminstrative role assignments and delegations. Formally, $FAA = AA \cup DAA$ where *DAA* is the projection of delegations which omits the delegator, that is, $DAA = \{(de; r; n) \mid (dr; de; r; n) \in DEL\}^*$.
- $DEL \subseteq ADMIN \times ADMIN \times ROLES \times N_\infty$, is the delegation relation from delegators to delegates of administrative roles where $(dr; de; r; n) \in DEL$ implies $dr \neq de \wedge (dr; r; n+1) \in FAA^*$.
- The administrative scope (*ASCOPE*) is the set of administrative permissions that an administrator has been assigned. Formally: $ASCOPE \subseteq ADMIN \times APRMS$ where $ASCOPE = \{(a; p) \mid \exists r \in ROLES : (a; r; n) \in FAA \wedge (r; p) \in APA\}^*$.

4. Empirical implementation

In OpenStack, Keystone, the centralised identity service, assigns roles to users, and each cloud service assigns its own permissions to these roles. In this way users are authorised to access different OpenStack services. In our empirical federated OpenStack implementation, attributes are assigned to users by federated Identity Providers (IdPs - *IOSes*). A key component of our implementation is therefore the attribute mapping service, which maps between the user's IdP assigned organisational attributes, and the OpenStack roles assigned by Keystone. OpenStack roles are equivalent to the workflow attributes of our FABAC model. In this initial implementation of federated OpenStack, we implemented the FABAC model as follows:

- The OpenStack/Keystone administrator determines the set of workflow attributes (*WATS*) that will be used by the OpenStack cloud services.

- Each cloud service provider administrator determines the workflow permission assignments (*WPA*) for the given set of OpenStack roles (*WATS*) by setting the policy for its RBAC policy decision point.

1. The OpenStack/Keystone administrator deter-mines the organisational attributes (*OATS*) that are acceptable from each Identity Provider, that is the credential validation rules (*CO*). This determines the valid user-organisational attribute assignments (*UA*). The Keystone administrator also specifies the

valid attribute mappings (*AM*) from organisational attributes to workflow attributes.

2. The Identity Provider determines the credentials that will be issued to its users (*CREDS* and *CU*), but the Keystone administrator determines which of these are valid.

3. The current implementation does not support organisational attribute to workflow permission assignments (*OPA*).

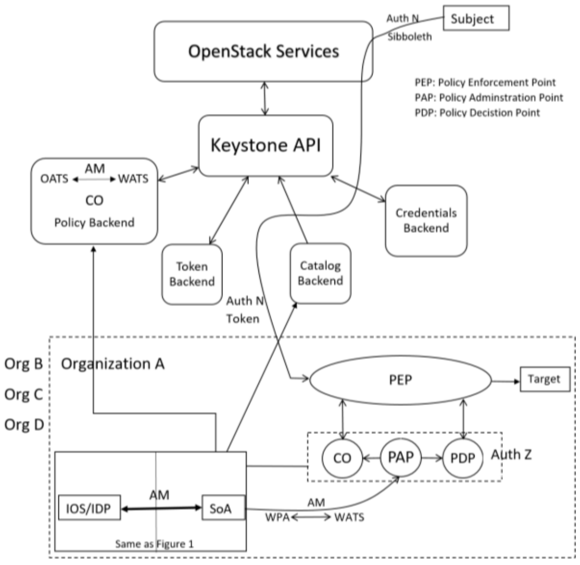

Figure 3. Keystone and FABAC integrated architecture.

The above clearly places a lot of burden on the Keystone administrator, since he has to set the majority of the authorisation policy, and this burden increases as additional IdPs (typically cloud using organisations) are added to the federation. In many cases, the Keystone administrator won't know the internal organisational structure of these IdPs, nor what organisational attributes they already assign to their users, so this places a significant burden on him. For this reason, many federations today simply define a standard set of attributes and require each IdP to map from its organisational attributes into the standard federation ones, and only release these to the service provider. But this model is typically too constraining and does not provide the fine grained access controls that are needed. This is the reason we develop the FABACD model, and it makes the attribute releases more flexible and the administration of them distributed to the local administrators who are familiar with the IdPs.

By applying the FABACD model to federated OpenStack, the administrators of the federated IdPs are recognised as being trusted to specify all the organisational attributes (*OATS*) that are issued, all the credential validation rules (*CO*) for these, and hence all the user organisational attribute assignments (*UA*) that Keystone will accept. Because they are trusted, no fine grained access control rules are needed to control this. However, they are not trusted to perform attribute mappings to all workflow attributes. These are tightly controlled by the Keystone administrator, so that each administrative role can only map a subset of the workflow attributes, consistent with its attribute scope. In this way each IdP administrator is controlled in the number of permissions that he can assign to end users. We have not yet finalized the implementation design for the FABACD model, but our initial thought as follows.

In order to configure the system, the Keystone administrator, being the SoA, determines which administrative roles he will issue (*ROLES*). He adds these to the existing set of Keystone roles (*WATS*) stored in the backend database, using the existing Keystone APIs. He then determines which permissions each role will have, in terms of which workflow attributes they can control (*APA*). This is done by providing the SoA with a new set of commands for creating, deleting and reading entries from a new table in the Keystone backend database that maps *ROLES* to *WATS*. Next the SoA assigns the various administrative roles to the various IdP administrators (*AA*), using the existing Keystone mechanism for assigning end users to roles. Finally, he creates a new set of authentication credentials for each administrator (typically a username and password) that each administrator must present in order to update the Keystone authorisation policies. In this way, we utilise the existing Keystone (nonfederated) access control mechanisms to authenticate the IdP administrators and assign to them their administrative roles. Note that delegation of roles is already implemented in Keystone, so the administrator will be able to delegate his administrative role to someone else using the in-built Keystone mechanism.

A new administrative client has been created for the IdP administrators, along with supporting policy enforcement point (PEP) functionality in Keystone. An IdP administrator uses this client to log into Keystone, using his newly acquired authentication credentials. Keystone validates the credentials, looks up the administrator's roles (*ROLES*) in its database, and returns a scoped token to the client which grants the user permission to use these roles for a limited period of time (this is standard Keystone procedure). The administrator now has a set of commands to choose from: create organisational attribute, delete organisational attribute, read organisational attributes, assign organisational attribute to IdP, remove organisational attribute from IdP, read IdP's organisational attributes, create new organisational attribute mapping, read organisational attribute mappings, and delete organisational attribute mapping.

The only access control needed to create, delete or read organisational attibutes (*OATS*) is that the administrator must have at least one administrative role (*ROLE*). Other than that, the administrator is trusted to create any organisational attribute he wishes. The PEP checks that the administrator has a *ROLE*, and then creates an entry in a new backend table that records the *OATS*. Note that any *ROLE* holder is authorised to create, read and delete any *OAT* since the model recognises that administrators are collectively trusted to manage the organisational attributes without any fine grained controls from the SoA (Keystone administrator). When a read operation is issued, all the *OATS* are returned, regardless of which administrators actually created them. This is actually what is needed in today's federations, where multiple IdPs typically issue the same standard set of attributes (often the eduPerson schema set). Consequently, there is little point in making each IdP administrator redefine the same set of organisational attributes. Once an organisational attribute has been defined by one administrator, it can be used by all. This provides them with backwards compatibility with existing federation procedures, until they no longer all need to issue a standard set of attributes.

When assigning an organisational attribute to an IdP, the administrator specifies the user-friendly name of the IdP that was created by the Keystone administrator when he added the IdP to the federation (by adding the IdP to the service catalog). In this way the administrator cannot add new IdPs to the federation, since he is not trusted to do this. He can only manage existing IdPs. The PEP records the IdP to organisational attribute assignment in a new backend table, which will subsequently be used for credential validation (*CO*) of end users. Again, the only access control needed to assign, remove or read an organisational attribute to/from/at an IdP is that the administrator must have at least one recognised administrative role (*ROLE*). Consequently, each administrator is trusted to assign or remove any organisational attribute to/from any IdP in the federation, as each administrator is recognised to have this authority. This feature (along with *create new organisational attribute mapping* – see below) is often needed in federations where there is one federation administrator in charge of assigning workflow permissions to all the different members of the federation. Remember that the Keystone administrator only controls which workflow permissions these administrators are authorised to map to, so he ultimately controls what end users may do with OpenStack services.

The *create new organisational attribute mapping* operation displays a list of workflow attributes (*WATS*) that the administrator is authorised to map organisational attributes into, by virtue of his valid *ROLES*. This is his administrative scope for the session. Consequently, different administrators will be shown different lists of workflow attributes. The administrator is shown the complete list of organisational attributes, and his picks one or more of these to map from. He then picks one or more workflow attributes to map these into. He is therefore effectively choosing which set of end users is to be given which particular workflow (cloud) permissions under his control. Note that the attribute mappings are from valid organisational attributes (according to *CO*) and not from the credentials issued by IdPs. Thus it is possible for multiple sets of users from different IdPs who share the same set of valid organisational attributes to be given the same set of workflow permissions. Through this interface, many to many attribute mappings are supported. This access control mechanism is much more flexible than standard RBAC, in which each role is assigned one or more permissions. In FABACD, an organisational attribute on its own may have no permissions assigned to it, and only in combination with one or more other organisational attributes does the end user obtain one or more workflow permissions. For example, a mapping may state that users with organisational attributes "*organisation=kent*" and "*status=staff*" and "*organisationalUnit=CS*" map into the workflow attributes "*role=user*" and "*tenant=KentCS*". Each newly created mapping is given a unique ID by Keystone, which references it in the backend database. This ID must be used by the administrator to delete an organisational attribute mapping. The Keystone PEP ensures that the read organisational attribute mappings operation only returns the mappings that are within the administrative scope of the administrator, that is where the *WATS* in the mapping have been assigned to the administrator's *ROLES*. Administrators therefore cannot see either the full set of workflow attributes or the permissions that have been granted to administrative roles that they do not possess.

We conduct a simple performance evaluation of the implementation. Because we do not have any statistical values in the number of organisations involve in a federation, the number of users in each organisation, how often is there an organisation that opts out from a federation and whether this organisation is replaced by another one, so we would not be able to introduce a statistical comparison between the two approaches specified in Section I. Instead, we use a simple but illustrative case for this purpose. In this case, there are three organisations that join their work and form a federation. Each organisation has 100 users. In the middle of their collaboration, for some reason, one organisation leaves and is replaced by another organisation that has 80 users. In the first approach, initially Keystone issues 300 tokens (certificates) to the users, then revokes 100 tokens and issues 80 new tokens. Note that the revocation of 100 tokens and the issuance of 80 new tokens normally happen in a relatively short time frame, and are therefore error-prone. In the second approach, neither Keystone nor the IdPs need to issue or to revoke any certificates, but the Keystone administrator has to rewrite its credential validation policy (step 3 and 4 in the FABAC model). We believe that if the number of users is large and/or in a highly dynamic environment, the second approach is better in terms of issuing/revoking time and effort, and the storage of certificates.

In terms of computation cost for validating user credentials, the first approach requires the PDP in a target organisation to validate the credential against the Keystone's policy. For the second approach, the PDP validates the credential against its IdP's policy, then against the Keystone's policy and finally checks the mapping of roles in the Keystone's policy. The difference in the two approaches is that in the second approach, the system has to validate the credential against its IdP's policy and check whether or not the user role is mapped to a workflow role in the Keystone's policy. In our test environment, the average time for checking a mapping of two roles is 0.25ms on a core i5 1.6GHz computer with 8G RAM. The process of validating the credential against the IdP's policy depends on the infrastructure of each organisation. In our previous research [9], the average computation time for this process was 6.93ms on a Intel P4 processor 2.4GHz and 2G

RAM. It is worth noting that these values largely depend on implementation and our implementation is just a proof of concept.

5. Related works

The RT model [10, 11] is a very powerful framework for representing policies and credentials in distributed authorisation system, and provides the capability of role mapping. In the XACML version 3.0 model [12-14], the SoA in one organisation can delegate an administrative policy to the administrator in another organisation in order to set up a collaboration between two organisations. This allows permissions or roles in one organisation to be mapped into attributes in the second organisation. Our model has one advantage over the RT and XACML models in that it supports both credentials and credential validation rules (*CO*). Whilst the RT model does have credentials it does not have a concept of a component that validates user credentials. In Kagal and others' model [15], the authorisation policy in a target domain is only modified and updated by the security officer in that domain, so that the model is not appropriate for dynamic and large environments like federations. In [16], the policy of an organisation is only updated by its administrator and does not have a mechanism to separate collaborations from each other. Furthermore, the policy for role mapping is statically set by the administrator in a system-site. The PERMIS infrastructure [17] supports the dynamic assignment of roles to users in different domains but does not have the capability of dynamically adjusting the authorisation policy. The CAS model [18] is used for authorisation in Grid environments but the policy of a CAS server is only modified and updated by its predefined administrators. Furthermore, it can not separate the workflow security infrastructure from the organisation level security infrastructure or explicitly deal with multiple collaborations. If there is a change of participant in the collaboration, the CAS server has to be reconfigured with a new set of users and users' permissions. The framework proposed by Firozabadi and others in [19] does not separate inter-organisational workflows from organisation-level changes. Furthermore, the framework has no mechanism to separate collaborations from each other. In [20], Mukkamala and others proposed the dynamic coalition-based access control (DCBAC) model that facilitates the formation of dynamic coalitions through the use of a registry service, where available services can be advertised by potential coalition members. This model does not consider the decentralised administration of collaborations, so that only the SoA in an organisation can register the organisation's services to coalitions. Furthermore, the workflow security infrastructures are not separated from the organisation level security infrastructures. Finally, the g-SIS model [21], [22] focuses on operational aspects that bear on group membership and is not an administrative model. Therefore, this model is orthogonal and complementary to administrative models, including ours.

6. Discussion and conclusion

One might think that our administrative control model is too lax, in that remote administrators are fully trusted to map any organisational attributes from any issuing organisation into the workflow permissions of their administrative scope. One might think this should be restricted so that each administrator can only administer the organisational attributes of one issuing organisation, his own. However we do not think this poses a significant risk or vulnerability, since we presumes that the workflow permissions that have been assigned to an administrator, via his administrative role, are because his organisation is somehow responsible for them. For example, the permissions are needed in order to participate in a collaboration, or the resource they grant access to is being paid for on a per use basis. Furthermore, we assume that the organisation trusts the administrator to act responsibly and has nominated him for the role to the SoA.

Thus, if the administrator were to give these permissions to users from another issuing organization, either by mistake or maliciously, this would soon be detected and he would then have to explain this to his superiors. The other argument is that in some collaborations it may be desirable for users from different organisations to access the same set of workflow resources, and our model allows one administrator to control this.

Workflow roles may need to be adjusted in order to accommodate workflow changes or new collaborations. Our model only allows the SoA to do this, which might be a limitation in some situations. We suggest that the administrative models proposed in [3], [8] might be useful to solve this. Another limitation is that a revocation model that allows SoA (and his/her subordinates) to revoke credentials (that contain both organisational and administrative attributes) is not presented in this paper, and has not been fully investigated in our current research.

To conclude, administration of an authorisation system is very important and must be carefully controlled to ensure that the authorisation policy does not drift away from its original objectives. Decentralizing the administration of an authorisation system without losing central control over broad policy is a challenging goal for system designers and architects. Our work provides a significant and practical advance towards this goal by proposing a federated ABAC administrative model. This management model allows administrators to be created by assigning administrative roles to them on demand. Administrative roles grant permissions to (possibly remote) administrators to dynamically update the authorisation policy of a target resource but only within the constraints of their administrative scope. Whilst the administrators are free to determine who the users should be (via their organisational attributes) they can only determine the users' permissions in a tightly controlled way, since the resource administrator (SoA) assigns the workflow permissions to the administrative roles. The result is that a wide set of users with greatly varying organizational attributes can easily gain access rights to the protected resources in the target domain with minimal effort by the SoA.

References

[1] American National Standards Institute, "Role based access control," 2004. ANSI/INCITS 359-2004.

[2] Bill Fisher, Norm Brickman, Santos Jha, Sarah Weeks, Ted Kolovos, Prescott Burden, Attribute Based Access Control, NIST, 2016

[3] R. Sandhu, V. Bhamidipati, and Q. Munawer, "The ARBAC97 Model for Role-Based Administration of Roles," ACM Transactions on Information and System Security, vol. 2, no. 1, pp. 105-135, 1999.

[4] T.-A. Nguyen, D. Chadwick, and B. Nasser, "Recognition of Authority in Virtual Organisations," in Proceedings of the 4th International Conference on Trust, Privacy & Security in Digital Business, (Regensburg, Germany), pp. 3–13, Springer Berlin/Heidelberg, September 3-7 2007.

[5] T.-A. Nguyen and C. N. Vu, "A federated abac model for collaboration management in federations," Journal of Science and Technology for Energy, vol. 11, pp. 69–76, 11 2016.

[6] D. Chadwick, S. Otenko, and T.-A. Nguyen, "Adding support to XACML for multidomain user to user dynamic delegation of authority," International Journal of Information Security, vol. 8, pp. 137-152, April 2009.

[7] M. H. Kang, J. S. Park, and J. N. Froscher, "Access Control Mechanisms for Inter-Organizational Workflow," in Proceedings of the sixth ACM symposium on Access control models and technologies, (Chantilly, Virginia, USA), pp. 66-74, ACM Press, May 2001.

[8] J. Crampton and G. Loizou, "Administrative Scope: A Foundation for Role-Based Administrative Models," ACM Transactions on Information and System Security, vol. 6, no. 2, pp. 201-231, 2003.

[9] T.-A. Nguyen, Delegation and Recognition of Authority in Virtual Organisations. PhD thesis, University of Kent, UK, 2010.

[10] N. Li, J. C. Mitchell, and W. H. Winsborough, "Design of a Role-based Trust-management Framework," in Proceedings of the

2002 IEEE Symposium on Security and Privacy, (Berkeley, California, USA), pp. 114-130, IEEE Computer Society Press, May 12-15 2002.

[11] N. Li, J. C. Mitchell, and W. H. Winsborough, “Distributed Credential Chain Discovery in Trust Management,” Journal of Computer Security, IOS Press, vol. 11, no. 1, pp. 35-86, 2003.

[12] E. Rissanen, “eXtensible Access Control Markup Language (XACML) Version 3.0,” tech. rep., OASIS, August 2010.

[13] E. Rissanen, “XACML v3.0 Administration and Delegation Profile Version 1.0,” tech. rep., OASIS, August 2010.

[14] E. Rissanen, “XACML v3.0 Core and hierarchical role based access control (RBAC) profile Version 1.0,” tech. rep., OASIS, August 2010.

[15] L. Kagal, T. Finin, and Y. Peng, “A Delegation Based Model for Distributed Trust,” in Proceedings of the IJCAI-01 Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents, (Seattle, USA), pp. 73-80, August 2001.

[16] J. S. Park, K. P. Costello, T. M. Neven, and J. A. Diosomito, “A Composite RBAC Approach for Large, Complex Organizations,” in Proceedings of the ninth ACM symposium on Access control models and technologies, (Yorktown Heights, New York, USA), pp. 163–172, ACM Press, June 02-04 2004.

[17] D. Chadwick, GansenZhao, S. Otenko, R. Laborde, L. Su, and T. A. Nguyen, “PERMIS: A Modular Authorization Infrastructure,” Concurrency And Computation: Practice And Experience, vol. 20, pp. 1341-1357, August 2008.

[18] L. Pearlman, V. Welch, I. Foster, C. Kesselman, and S. Tuecke, “Community Authorization Service for Group Collaboration,” in Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, (Monterey, CA, USA), pp. 50-59, IEEE Press, June 5-7 2002.

[19] B. S. Firozabadi, O. Olsson, and E. Rissanen, “Managing Authorisations in Dynamic Coalitions,” tech. rep., Swedish Institute of Computer Science, 2003.

[20] R. Mukkamala, V. Atluri, J. Warner, and R. Abbadasari, “A Distributed Coalition Service Registry for Ad-Hoc Dynamic Coalitions: A Service-Oriented Approach,” in Proceedings of the 20th Working Conference on Data and Applications Security (DBSec), (Sophia Antipolis, France), pp. 209-223, 2006.

[21] R. Krishnan, J. Niu, R. Sandhu, and W. H. Winsborough, “Group-centric secure information-sharing models for isolated groups,” ACM Trans. Inf. Syst. Secur., vol. 14, pp. 23:1-23:29, Nov. 2011.

[22] R. Krishnan, R. Sandhu, J. Niu, and W. H. Winsborough, “A conceptual framework for group-centric secure information sharing,” in Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, ASI-ACCS ’09, (New York, NY, USA), pp. 384-387, ACM, 2009.