Reputation trust mechanism under the organizational-based access control model

Khalifa Toumi, Hela Sfar, Joaquin Garcia-Alfaro

SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2015; **00**:1–13
DOI: 10.1002/sec

RESEARCH ARTICLE

Reputation Trust Mechanism under the Organizational-based Access Control Model

Khalifa Toumi, Hela Sfar, Joaquin Garcia-Alfaro

SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay,
9 rue Charles Fourier, 91011 Evry, France
E-mail: {Khalifa.Toumi, Hela.Sfar, Joaquin.Garcia_Alfaro}@telecom-sudparis.eu

ABSTRACT

The spread of high-speed networks changes the way in which organizations manage information. Distributed environments, such as multi-cloud environments, can be exploited by users belonging to different organizations. Companies are realizing that they can achieve significant cost savings by outsourcing some of their IT environments to specialized service companies. This rapid transition has introduced a number of security risks and challenges. The resulting environment cannot succeed at addressing them without the use of access control policies and the definition of trust mechanisms.
Access control ontologies, as a structured way to represent real word elements, are widely employed for making the security interoperable and understandable. Ontologies that have been built for this aim, suffer from the lack of crucial elements for distributed environments. In this paper, we tackle the problem of trust based access control models. We define a list of trust elements that should be integrated into any access control ontology. We also provide a mapping technique that permits the exchange of trust information. Based on these two contributions, our reputation mechanism, that builds upon the Organization Based Access Control model (OrBAC), is created. To prove the efficiency of our proposal, we test it in a multi-cloud environment. Then, we conduct a set of experiments that show the high accuracy level of our system.

Received . . .

1. INTRODUCTION

Interactions between heterogeneous systems become crucial. The widespread of service based networks offers the possibility to share resources among different organizations. However, these collaborations cannot succeed without addressing some security challenges. Examples include the specification of the interoperability security policies and the definition of trust between the participants.

Access Control (AC) models are one of the important solutions that may enhance the security of a distributed system. They permit to define high level descriptions of security requirements and to design interoperability policies. Different challenges are addressed by security experts to improve the efficiency of such systems. Among them, several AC ontologies are proposed (1) to provide a precise definition of the security concepts, (2) to facilitate the collaboration between entities and (3) to decrease the misunderstanding between the different partners. An ontology lays the ground rules for modeling a domain by defining the basic terms and relations to make up the vocabulary of this topic field[37]. Actually, several AC ontologies are defined in the related literature [17, 19, 18]. However, these solutions suffer from the lack of the trust concepts' integration [20, 21, 23]. Trust is the concept that permits to change access decision and to produce reaction based on the real time behavior of the entities. Organizations are reluctant to share their resources without trusting other entities. One important solution to guarantee trust in these environments is the use of reputation mechanisms. Their aim is to gather and aggregate feedback about an entity with regard to other participants. In other words, a reputation mechanism defines methods to collect the different trust beliefs.

In this paper, we address several challenges related to reputation mechanisms, such as: (1) how to share trust levels between the entities? (2) what are the trust elements to define into an AC ontology? (3) how to integrate them? (4) how to understand the goal of trust relationships?. Our solution is based on two main contributions. First, we extend an existent AC ontology that was designed for the Organization Based Access Control model (OrBAC) [24, 25, 26, 33]. The extension consists on the definition of trust elements, relationships, and classes that should be added into any AC ontology. Second, we present an

enrichment algorithm to automatically integrate the new concepts to the AC ontology. Third, we propose a new mapping algorithm that permits understanding the goal of trust relationship.

Section 2 surveys related work. Section 3 introduces the ontology elements and the OrBAC model. Section 4 presents our new trust classes and our enrichment algorithm. Section 5 provides details about the main components of our assumed reputation mechanism. Section 6 provides our experiments and results. Section 7 closes the paper.

2. RELATED WORK

In this section, we discuss the relationship between our approach with regard to other works in the literature on access control models, trust framework and mapping techniques for ontologies.

2.1. Access Control Solutions

The first challenge, access control solutions, is one of the major security issues for any organization.

This aims to protect the use of the resources by the definition of rules that determine whether a user can perform an action or can access to a resource.

In the literature, there are four basic security models [33] which are Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role Based Access Control (RBAC) and Organization Based Access Control (OrBAC). Many derivatives have been deduced from these models in order to resolve a specific need. Lately, solutions for distributed systems based on OrBAC has gain a lot of interest. This is due to the fact that this model can manage easily larger number of users. It has also various advantages such as its high description level of rules and its administration models. Several studies based on OrBAC have been proposed [25, 29, 26, 23]

For instance, in [25] authors present the basic elements needed to define contracts between two organizations. This proposal allows to *automatically* derive a set of interoperability security policies having the local one already defined. The work presented in [29] focuses on the definition of an OrBAC context ontology. This is useful and interesting to make easier the security rules definition and derivation during interoperability sessions.

In [26], two new concepts: *virtual user* and *image web service* are defined in order to use OrBAC with web service technologies. Finally, in [23] a new trust framework was defined: the evaluation of the trust levels and its integration into OrBAC was presented. However, this solutions does not focus on the reputation process. Indeed, this paper can be an extension of this solution.

This paper is a new attempt to improve this access control model. As the best of our knowledge, it is the first approach that deals with the reputation mechanism with OrBAC.

2.2. Trust Framework

The definition of trust in distributed systems [34, 35, 28, 10, 32, 30, 31] has been widely studied. The definition of trust parameters depends on the application, the environment and the requirements of the administrator, etc. For instance, In [10], a trust model based on recommendation and experience is proposed. This paper presents these concepts with more interest to the optimism, tendency, forgetting factors and their integration in a trust model. Moreover, different challenges are studied by the existing trust frameworks. In [30], authors propose a challenge response protocol to identify malicious peers in P2P systems based on a trust framework. The solution uses challenge-response operations in each trust evaluation phase and validates every contacted peer along with recording their corresponding trust value. In [32], a general reputation model for collaborative computing system was proposed. The impact of applying reputation in virtual organisation has been also studied. This approach permits to rate users according to their service usage and service providers and their services according to the quality of service they deliver.

These solutions are useful and interesting, however, the problem of trust objective meaning is not present and discussed in these approaches. Finally, there are some frameworks [34, 35, 28] that have designed trust ontologies. In [34], the authors propose a solution that is an extension of FOAF schema. They allow to express trust in people, statements, other content of information sources. [34] was extended in [35] fusing on how the messages should be exchanged in the context of a communication environment.

According to [28], all trust ontologies convey as same objective the representation of trust relationships and goal. The existent solutions in literature helped us to extract the different trust elements that depends essentially on the environment. Therefore, in our approach, we will focus first on the definition of the trust concepts for the OrBAC ontology. And then, we will specify a new enrichment methodology to add these concepts in the OrBAC ontology.

2.3. Mapping Techniques

Ontology mapping is an active area of research in the last decade. Several approaches have been developed, among them some are of particular interest due the specific techniques they employ. AgreementMaker [9] is a framework that is based on a layered approach for the ontologies mapping. In the initial layer lexical and syntactic similarities are used to create a set of mappings. In subsequent layers, several iterations are made for refining the existing mappings using structural properties to create new mappings. After a sufficient amount of iterations, the final mapping is selected by

Security Comm. Networks 2015; **00**:1–13
DOI: 10.1002/sec
Prepared using secauth.cls

combining multiple computed mappings. The framework ASMOV [11] is based on the sense inventory WordNet as well as other specific domain ontologies for performing its mapping mechanism using a set of similarity measures. Afterward, a semantic verification is performed to remove correspondences that lead to inferences which cannot be verified or satisfied given the information present in the ontologies. The system YAM++ [12] using machine learning technique to combine different similarity metrics based on information retrieval techniques. Furthermore, similarity propagation technique are employed in order to discover and refine mappings. The inconsistency is also checked by YAM++ to enhance the mapping quality produced.

Recently several approaches are dealing by creating concepts profiles to perform the mapping. The PRIOR and CroMatcher frameworks [14] [15] are using the TF-IDF weight for building the concepts profiles whereas the Falcon-AO framework [16] applies a virtual document model with a parameterized weighting scheme.

However, these mapping approaches are not enough in the case of interoperability security policies. All the techniques presented above need to be updated in order to be used in our case. From our viewpoint, the virtual document technique is slightly interesting since it permits to check the concepts semantic ambiguity by comparing their virtual documents. In this paper we propose an adapted version of this technique to be applied in our system.

3. BACKGROUND

3.1. Basic Principles

Formally, an ontology `O` is a symbolic system consisting of:

- A set of concepts: a concept may represent a material object, a notion or idea in a real world domain. The concepts are also called classes of ontology. They are the basic objects manipulated by ontologies. They correspond to the abstractions of the relevant area of the domain, selected according to the objectives we give ourselves and the envisaged application ontology.
- A set of relations: a relation is an oriented function start from a set of classes called `Domain` to finish with an set of classes called `Range`. We note that the sets `Domain` and `Range` could hold one or several classes defined in the ontology. A relation can be symmetric, transitive, functional, and have a reverse cardinality.
- A hierarchy: concepts and relations are hierarchically related by a subsumption relation $\subseteq$ (a partial ordering), where c1 $\subseteq$ c2 (r1 $\subseteq$ r2) means that c1 is a subconcept of c2 (r1 is a subrelation of r2).
- A set of instances : an instance is the thing represented by a concept and they are related trough the `is a` relation. Instances also represent singular elements or individuals conveying knowledge about the domain. For example `ABC` is an instance of the concept `Triangle`.

3.2. The OrBAC Ontology

OrBAC was created in order to specify, model and implement security policies that have to control the access of the shared resources. However, different decisions can be badly chosen if the shared information are not well understood. Indeed, each partner has its local policy that depends on its resources, its roles, its activities, etc. Moreover, in some cases two organizations may define the same name for two different concepts.

A common language for all participants may be a solution of this problem. However, this approach is not suitable for all types of interoperability. This is due to the fact that the organization has its own security policy, expression language and specific information. Moreover, it can interact with different types of organization.

Therefore, the use of a common language does not entirely solve the problem. Thus, the use of ontology was necessary where each participants retains its own language and inform others about his knowledge. In Figure 1, the blue components presents the main classes of the OrBAC ontology. These classes may be divided into five parts. The first one permits to define an 'organization' that is a central element in OrBAC. Intuitively, an organization is any entity that is responsible for managing a security policy. For this purpose, OrBAC defines two abstraction levels: concrete and organizational. The first one forms the second ontology class 'concrete entity' that contains the concrete and implementation related concepts of subject, action and object. The third class defines the 'relevant entity'. It contains the roles that subjects, actions or objects are assigned in the organization. The role of a subject is simply called a role. On the other hand, the role of an action is called an activity whereas the role of an object is called a view. The fourth class focused on the 'security rules'. A security rule is a relation between organization roles, views, activities and contexts. It is defined as a role having permission, prohibition or obligation to perform an activity on a view within an organization. Finally, the fifth class, 'the context', is used to express different types of extra conditions or constraints that control activation of rules expressed in the access control policy. More details about this ontology may be found in [17, 29]

4. TRUST ONTOLOGY FOR ACCESS CONTROL

Several researchers are using ontologies to define the security policies. Moreover, trust ontologies has also been described before. However, as the best of our knowledge,

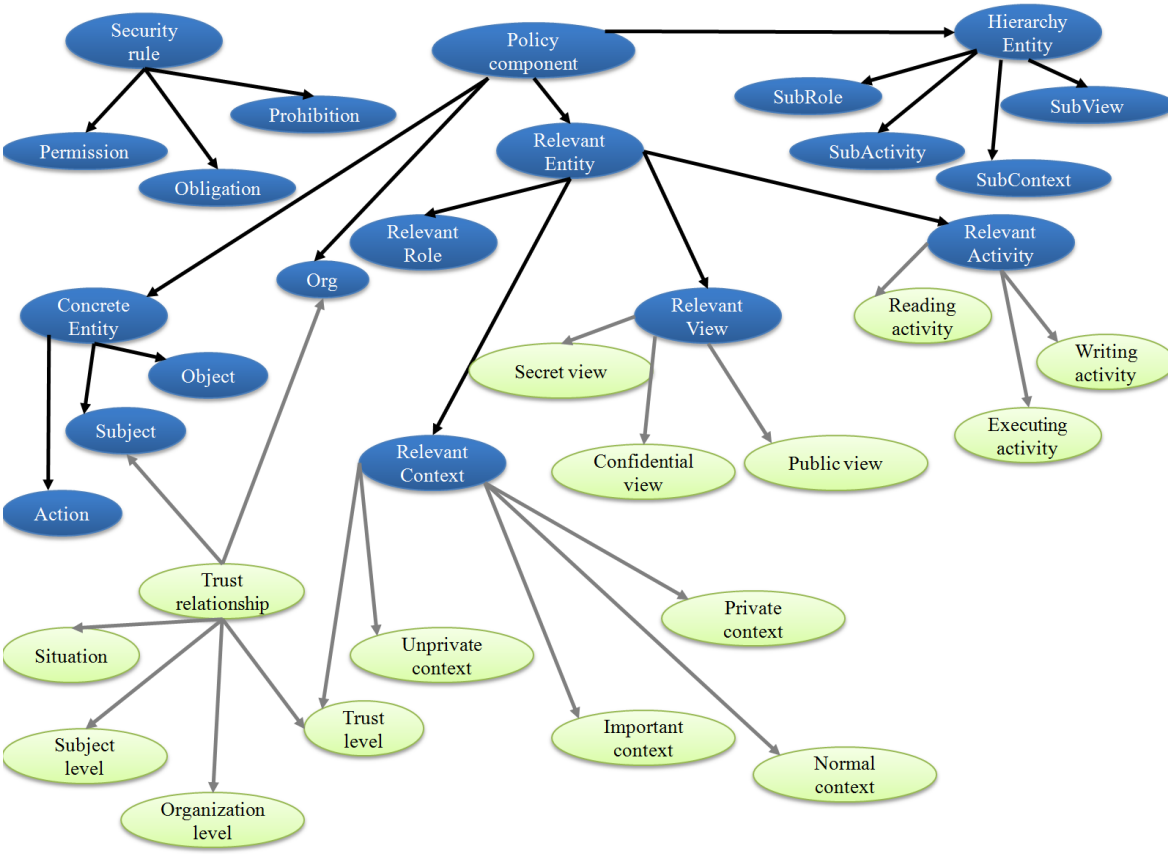


Figure 1. Trust ontology based on access control concepts

we could not find a trust ontology based on access control concepts. In our approach, we will not start from scratch. We will reuse the OrBAC ontology and we enrich it with new trust concepts and relations between them. The enrichment technique will be also proposed.

In order to define a trust ontology we need the following:

- To design a global vocabulary about trust to ensure a comprehensible way to share feedbacks.
- To define the goal concept (its classes and properties) that is the core of the mapping process.

The new trust elements, are depicted in Figure 1 (see green boxes), its main ontology classes are: **trustRelationship** (presented in Section 4.1), **Goal** (presented in Section 4.2), and **TrustLevel** (presented in Section 4.3).

4.1. Trust Relationship

In order to define the trust relationship, we need to specify the following five concepts: **Truster, Trustee, TimeStamp, Goal,** and **TrustLevel**.

The **truster** in our system is an O-grantor. It is the organization that will provide the service or offer the access. Each O-grantor have to define its trust policy and to evaluate periodically the trust values of the different participants. Let us note that any organization may be a truster.

The **trustee** can be the user or the organization that applies for a service. The trust relationship aims to evaluate its trust level.

In our framework, the trust level of an entity is evaluated each period with respect to the **TimeStamp**. The concept of time permits to illustrate the dynamism of our framework. For this reason, each relationship will be characterized by its evaluated time.

Finally, the **Goal** and **TrustLevel** are defined as two ontological classes.

4.2. Goal

A **Goal** is a set of parameters that permits to define the restriction area of a trust relationship. In our trust model, this concept is defined as a tuple (a, v, ctx) where a is an activity (a set of actions), v is a view (a set of objects) and ctx is a context (a specific condition that activates or deactivates a rule).

We may have several trust values regarding the same trustee since they depend on the goals. For example, a

Security Comm. Networks 2015; **00**:1–13
DOI: 10.1002/sec
Prepared using secauth.cls

laboratory TSPlab may trust the company MOS to manage a French project and distrust it for consulting an European project.

The OrBAC ontology gives a presentation of these three classes views, activities and contexts [17]. In this work, we extend this definition for trust proposal; Figure 2 shows our goal taxonomy. Following the new trust classes related to the tuple (a, v, ctx) presented in Figure 2 are introduced.

a For any activity entity, we have three new subclasses that are related to the type of the activities. Indeed, any activity in an access control model [23] belongs at least to one type of these classical classes *ReadingActivity*, *WritingActivity* or *ExecutingActivity*. The table I presents an example for a mapping between some OrBAC activities and these new ontology classes of TMSP university.

	ReadingActivity	WritingActivity	ExecutingActivity
declare			✓
manage	✓	✓	
add_note		✓	
submit			✓
consult	✓		
notify			✓

Table I. Mapping between OrBAC activities and the new classes.

v The new subclasses of the view concept are the *PublicView*, *ConfidentialView* and *SecretView*. This conception is inspired from the information classification of the European Commission and the OCCAR organizations: **(1)**Resources that contain information that may be used with any extern employees of another partner will be in *PublicView*. Any unclassified resource belongs to this type. The modification, the consultation of these data does not influence any financial, operational or personal problem.

(2) A resource that contains a sensitive data belong to the *ConfidentialView*. This information has an impact to the service level and performance of the enterprise. They may cause some financial loss, penalty, loss of confidence, etc.

(3) Finally, *SecretView* contains very serious personal and enterprise data. The malicious use of this document may cause a major economic impact, a fire of an employee, an interruption of relationship with other enterprise. They are available to some users that belonging to a particular mission (project, task, etc).

ctx For the context, we have four new classes that are related to the *privacy* and *importance* of the context. On one hand, a private context is defined for application proposal that means it depends on the local parameters as auditing results designed by the administrator, provisional context, logic context [17]. Any context that does not belong to the private class will be an element of the unprivate class. Mapping between two private contexts is high difficult.

We say that a context is unprivate or general, when it can be understandable or used in several organizations as temporal and geographical contexts.

On the other hand, we have that the importance property is determined by the administrator in order to highlight the value of a context in a goal. We deal only with two classes important and normal context. The first one is defined for the context that must be considered in order to share trust knowledge between the different organizations.

With our framework, a context is defined by default as a normal one that means it will not be used during the reputation process. In this case, the different goals modify_files_at_night, modify_files_today and modify_files are similar. Otherwise, the administrator has to change this type.

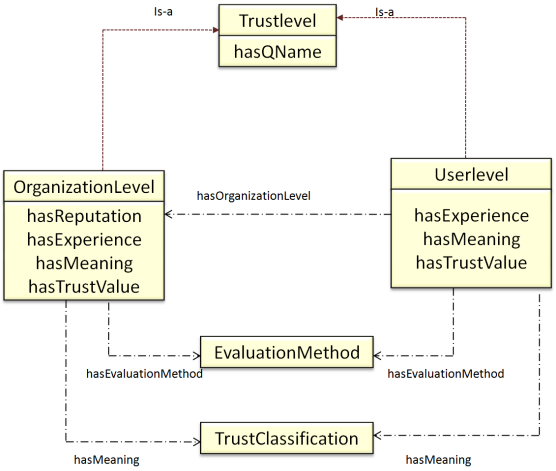


Figure 3. TrustLevel components

4.3. Trustlevel

Figure 3 illustrates the components of the **TrustLevel** class. This class is defined in order to present a trust level of a user or an organization. This element has two subclasses, *OrganizationLevel* and *UserLevel*. Both of them are related to the two classes *EvaluationMethod* and *TrustClassification*. The first one specifies the method used to evaluate the trust value that can be based on combining different parameters as it is defined in [23] or based on one parameter as the work in [22].

The second one is based on Golbeck classification of trust [34]. In this classification, there are 9 types:

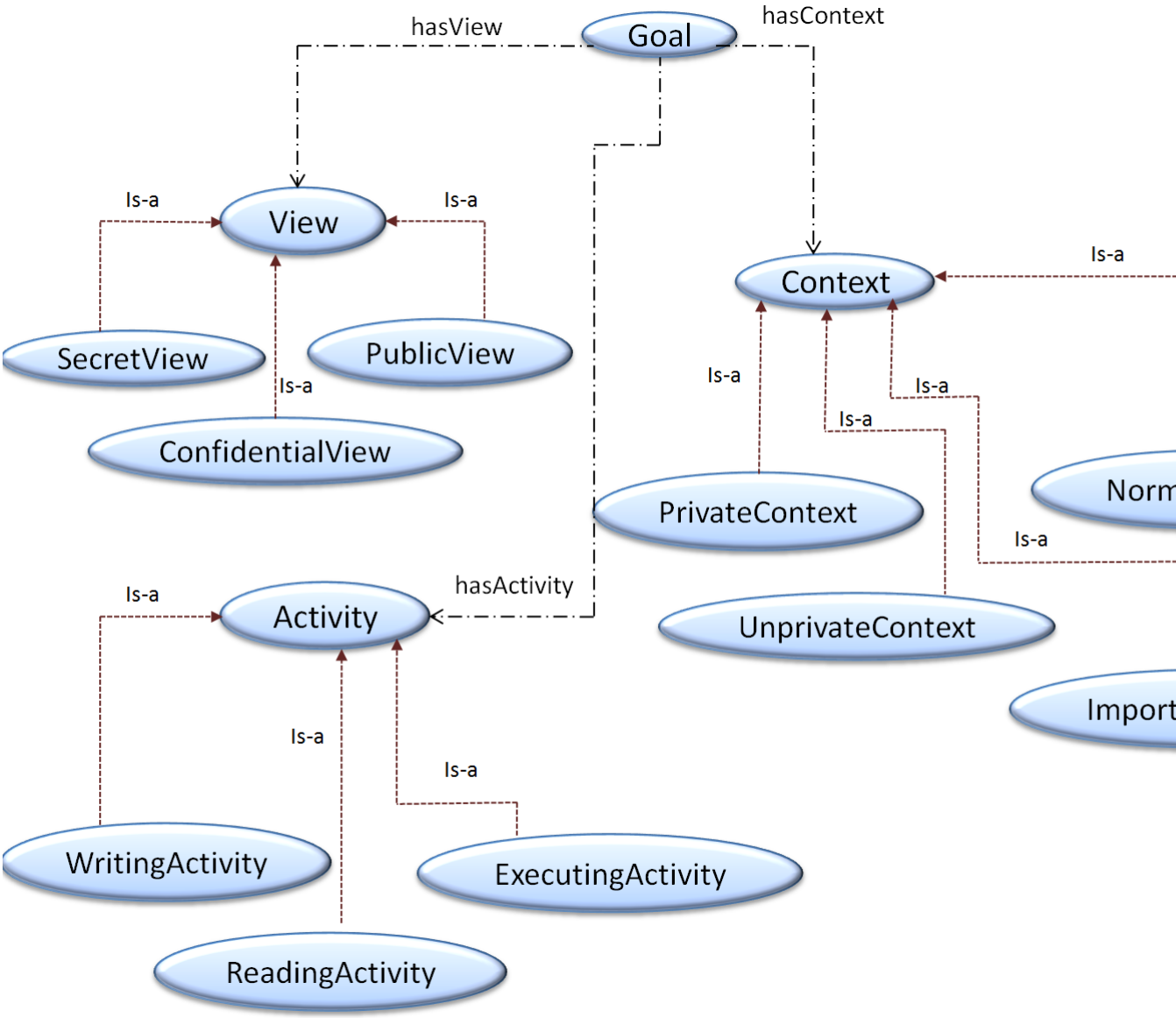


Figure 2. Goal Ontology

absolute distrust, high slightly trust, moderate distrust, slightly distrust, neutral, slightly trust, moderate trust, high trust and absolutely trust. This classification is used to have more comprehensible trust evaluation. Since, the same trust value may be considered different between two organizations. Therefore, with our solution the trust belief captures the trust value with its meaning (class).

4.4. Enrichment Technique

Throughout this section, we focus on the enrichment algorithm that permits to extend the OrBAC ontology to be used in our reputation mechanism. For an organization denoted `OrgA` with its ontology $\mathtt{O}_{\mathtt{OrgA}}$, we create a TrustOrBAC ontology denoted $\mathtt{TOO}_{\mathtt{OrgA}}$. Indeed, we use the new ontology structure shown in Figure 1, our problem is how to integrate automatically the used instances of $\mathtt{O}_{\mathtt{OrgA}}$ into the new structure. Therefore, our methodology aims to do this automatically in order to help the administrator. Our solution is based on intermediate semantic verification that we have denoted *InstSem*.

4.4.1. The InstSem Method

This instantiation method is performed merely for the trust concepts related to the OrBAC concepts *Relevant Activity* and *Relevant View*. To do this, we proceed by an intermediate semantic verification in order to decide, for a given instance, which is the relevant trust concept

Security Comm. Networks 2015; **00**:1–13
DOI: 10.1002/sec
Prepared using secauth.cls

Algorithm 1 InstSem(TrustOrBAC ontology (TOO))

Input: An instance I in the OrBAC ontology of the organization Org.
Output: I is instantiated into TOO.
InsHyp(Execute) ← GetInstance(WordNetURI, Execute).
InsHyp(Read) ← GetInstance(WordNetURI, Read).
InsHyp(Write) ← GetInstance(WordNetURI, Write).
if I in InsHyp(Execute). **then**
  Instantiate(TOO, I, Executing Activity).
else if I in InsHyp(Read) **then**
  Instantiate(TOO, I, Reading Activity)
else if I in InsHyp(Write) **then**
  Instantiate(TOO, I, Writing Activity).
else
  $sense_I$ ← GetSense(WordNetURI, I).
  MaxRela ← 0.
  TrustConceptName ← "".
  for each set in (InsHyp(Execute), InsHyp(Read), InsHyp(Write)). **do**
    MaxRelaSet ← 0
    if set = InsHyp(Execute) **then**
      Name ← Executing Activity.
    else if set = InsHyp(Read). **then**
      Name ← Reading Activity.
    else
      Name ← Writing Activity.
    end if
    for each I_j in set **do**
      $sense_{I_j}$ ← GetSense(WordNetURI, I_j).
      relat ← ComputeRelatedness($sense_{I_j}$, $sense_I$).
      if relat > MaxRelaSet. **then**
        MaxRelaSet ← relat.
      end if
    end for
    if MaxRelaSet > MaxRela. **then**
      MaxRela ← MaxRelaSet.
      TrustConceptName ← Name
    end if
  end for
end if
Instantiate(TOO, I, TrustConceptName).

that it will be linked to. For example, we suppose that the instance *send* is related to the concept *Relevant activity* in an OrBAC ontology of an organization, the *InstSem* role is to decide whether this instance will be linked to the trust concept *Reading Activity*, *Writing Activity*, or *Executing Activity* in the TrustOrBAC ontology `TOO`. To do so, three main steps are proceeded for a given instance `I`:

1. **Retrieving the instances related to the activities Read, Write, and Execute in a dictionary**. We have used the sense inventory WordNet [36] in order to retrieve this instances. Thus, we have implemented a query that asks WordNet for Instance Hyponyms related to each one of theses activities denoted InsHyp(act) where act is the activity Read, Write, or Execute.
2. **Checking whether the instance `I` belongs to an InstHyp set of an activity**. Whether `I` belongs to one of the InsHyp set of these activities then it will be linked to the corresponding trust concept. For instance, if we find that `I` belongs to InsHyp(Execute) then it will be linked to the trust concept *Executing Activity*. Otherwise, the mechanism skips to the next step.
3. **Computing the Relatedness value between the `I` sense and each instance sense in InsHyp sets**. For each InsHyp we compute the Relatedness value between each one of its instances senses and the `I` sense then we store the maximum value. Finally, `I` is linked to the trust concept that provides the maximum value of relatedness among the three obtained maximum value of each set. Let s_{I_j} is the sense of the an instance I_j, that belongs to InsHyp set of one of the three given activities, retrieved from WordNet and s_I is the sense of the instance I. The relatedness between s_{I_j} and s_I is computed as follows:

$$Relatedness(s_{I_j}, s_I) = \frac{NumberOfOverlaps}{\frac{|Gloss(s_{I_j})|+|Gloss(s_I)|}{2}} \quad (1)$$

We note that the Gloss of a sense is its description in natural language that is also provided by WordNet.

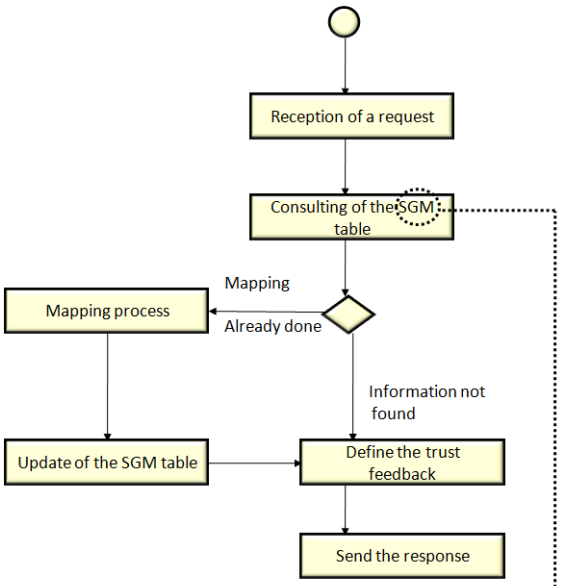
Algorithm 1 implements the *InstSem* mechanism. The procedure *Instantiate* aims to link the instance `I` to the target trust concept. *ComputeRelatedness* is the function that computes the relatedness value between two senses. The procedures *GetInstance* and *GetSense* implement two queries asking Wordnet respectively for the set of instances of a given activity and the sense of a given instance.

5. REPUTATION PROCESS

Example:

Firstly, we introduce the reputation process by a simple example illustrated in Figure 4. In this example, different cloud providers are collaborating together to respond to the user requests. In this scenario, we may have an organization `OrgB` that needs to know the trust beliefs regarding a cloud provider `OrgC` related to a specific goal (`saving a patient file`) before requesting it. Therefore, it sends a request to a list of its friends (among them, we find the organization `OrgA`) using our framework. Each company that receives this request may accept or refuse to share its trust beliefs for a security issue. If it will send its beliefs, it has first to understand the trust goal (`saving a patient file`). This process is a non trivial task due to the difference of vocabularies and protocols that are used into each organization. Therefore, our new algorithm will be processed in order to extract the list of objectives that can be equivalent to this goal. Three main steps are characterizing this solution.

Initially, for each concept in the recommender ontology, the algorithm constructs a virtual document based on any related information in this ontology. Furthermore, this document will be used in order to predict the concept meaning. Moreover, a similarity metric is performed based on this meaning and its trust characteristics. This metric permits to decide about the equivalence between this concept and the designed goal. Next, the trust feedback (value and meaning) related to the equivalent concept will be sent to `OrgB`. * Finally, the requester (`OrgB`) will receive different feedbacks from all its friends that permits to take a decision about this cloud provider.



Example of a Server Goal Mapping (SGM) table for OrgA

Organization	External goal	Local goal	Similarity value	Meaning
OrgB	Goal1_OrgB	Goal2_OrgA	0.89	Absolutely trust
OrgB	Goal3_OrgB	Goal1_OrgA	0.5	Slightly trust
OrgC	Goal2_OrgC	Goal4_OrgA	0.75	Slightly trust

Figure 5. Different tasks of the trust web service

Process description:

The extension of the OrBAC ontology with the required trust elements is compulsory to participate in the reputation process. Indeed, any organization that aims to share or receive trust beliefs from other participants has to update its ontology with the enrichment algorithm defined in Section 4.4. This task will be done only once (before the collaboration). In the following, we will detail the reputation process between the different participants.

* How to evaluate the trust is not detailed in this paper. Interested reader may have more information in [23].

Different steps have to be realized during this process:

1. **Before the collaboration:**
   (a) We install a trust web service in each organization. This web service will be responsible of providing the list of equivalent goals to an external one. This service will use two inputs, the TrustOrBAC ontology and a table called Server Goal Mapping (SGM). In this table, we save the equivalent goals and equivalence rate with the external goals. Initially, this table is empty, and it will be filled during the communication.
2. **During the collaboration (see Figure 5):**
   (a) Initially the web service is waiting of any request from a truster.
   (b) When a reception of a request is provided, the recommender will extract the requested goal and will consult its SGM table.
   (c) If the mapping is already done (that means the recommender has received a previous request related to the same goal), we will jump to the definition of the trust feedback task.
   (d) Otherwise, the recommender has to do the mapping algorithm that will check the equivalence between the requested goal and all the local goals of the recommender then it will update the SGM table.

5.1. Mapping Algorithm

Ontology mapping is a complex process that helps in reducing the semantic gap between different overlapping representations of the same domain and facilitates the information exchange among them [1].

Definition 1
Formally, we define the mapping between two ontologies O_1 and O_2 as 5-tuples of the form $(id, e_1, e_2, \eta, \rho)$ [2], with id is a unique identifier of the relationship, e_1 and e_2 are two entities which respectively belong to O_1 and O_2, η a confidence measure in $[0...1]$, and ρ a relation between e_1 and e_2 typically equivalence or disjointness. □

In Section2, we have shown some ontology mapping techniques have been proposed in the literature. After a careful study of these techniques, we conclude that the mapping method based on virtual document technique for computing ontologies entities similarity is the most suitable for our application. However, this technique, as it is, cannot be used in our case for security reasons: First, for providing a high security level, it is forbidden the exchange of the whole ontologies between two organizations. Indeed, the use of the classical mapping solution will give the possibility to an attacker to predict the content of some data (such as the organization structure of the ontology, and the instance names). The organizations

Security Comm. Networks 2015; **00**:1–13
DOI: 10.1002/sec
Prepared using secauth.cls

are merely allowed to share needed information about their ontologies entities not their all information, moreover even information that can be exchanged are restricted to well-defined conditions. Second, the mapping technique should take into consideration the new trust concepts that have been defined. Third, in our case, we do not need to do the mapping between all the components of an ontology, we have just to propose a mapping technique between instances since each organization uses the same TrustOrbac ontology structure. Then, the organization populates these ontology concepts with instances according to its own needs. Hence, the TrustOrbac ontology is different from an organization to another in terms of instances. Therefore, the mapping is performed only between instances. Therefore, we proposed an adaptation of this mapping method for being applied in our case. Figure 6 presents the adapted mapping method.

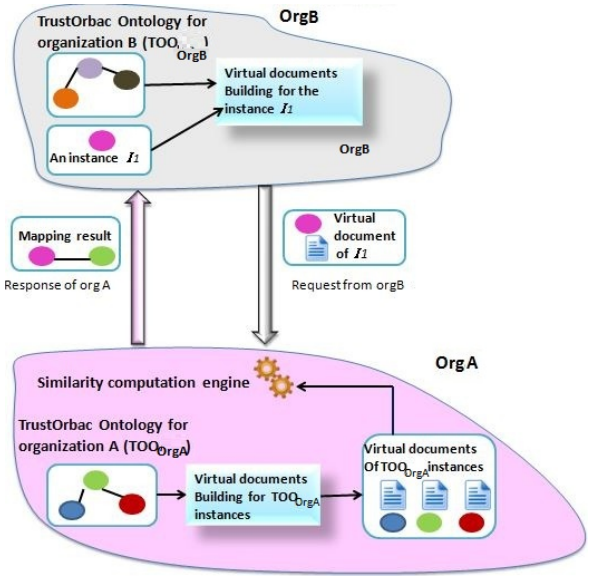


Figure 6. The adapted mapping method overview

In the case depicted in Figure 6, we have two organizations `OrgA` and `OrgB`. The latter wants to ask the former about its trust belief regarding a user bob related to the goal I_B. As we have defined the concept goal, it is composed of three instances of `an activity a1, a view v1, a context c1`. Therefore, `OrgB` will send a special document called virtual document about each one of these three instances to `OrgA`. Here, we recall that the different instances defined in TOO_{OrgB} belong to the shared information. Otherwise the `OrgB` is not allowed even to ask neither organization about them.

Now, when `OrgA` received the request, it collects for each possible shared instance of its local activities, views and contexts its corresponding virtual document. Thereafter, `OrgA` computes the similarity between each virtual document with the virtual document of the three instances. The instance I_A which provided the higher values of similarity will be used and the trust levels (value and its meaning) related to the goal I_A will be send to `OrgB` as a recommendation.

Next, we explain each step in this adapted ontology mapping method: *Virtual document building*, *Similarity computation*, and *Mapping construction*.

5.1.1. Virtual Document Building

The virtual document model serves as *background* gathering method. Indeed, a virtual document for an ontology entity holds the *background* of this entity from the ontology. Usually, the entity *background* holds the label of the target entity, its neighbor entities labels in the ontology e.g entities that are linked with this target entity, and the comment associated to the designed entity . According to our application restrictions, we define some differently an instance *background*.

Definition 2
`An instance background`. Let `Org` be an organization, TOO_{Org} be the TrustOrbac Ontology of the organization `Org`, `I` be an instance in TOO_{Org}, `E` be the set of entities that are linked to `I` in TOO_{Org}. `E` = $\{e_1, e2, ..., e_n\}$ where $\forall$ $e_i \in$ `E`, if e_i is a concept or an instance so $\exists$ a relation $e_j \in$ `E` that links `I` and e_i. Let `SH(E)` be the set that contains entities in `E` which have the attribute shared. Let `C(I)` is the *background* of `I`. `C(I)` includes the set of all trust concepts, that are related to `I`, denoted `TC(I)`, the set of words in the comment of `I` denoted `Com(I)`, and `SH(E)` after removing the trust concepts in order to avoid redundancy. More formally, this can be defined as follows: `C(I) = TC(I) ∪ Com(I) ∪ (SH(E)\TC(I))` □

To enhance the performance of our solution, it is recommended to create the virtual documents before the collaboration. With our solution, we provide to the administrator the possibility to choose between the building of this documents during the collaboration or before.

5.1.2. Similarity Computation

At this step, the virtual documents for instances are built, we are able now to compute the similarity between two virtual documents. To do this, two steps are performed:

1. **Transforming virtual documents into a vector-space model** [6]. Once created, it is necessary to process the virtual documents to facilitate the computation of documents similarities. This is performed by transforming the documents into two vector-space model. A given document is represented in this space as two vectors one for the trust parameters and another for the rest of information. Each vector contains the frequency weight of each term belonging to this target document. The frequency weight of a term in document is computed by the `tf-idf` measure [7]

as follow:

$$tf - idf(t, d_i, D) = \frac{n_{t,d_i}}{\sum_k n_{t,d_k}} \times log \frac{|D|}{|\{d \in D : t \in d\}|} \quad (2)$$

where D is the collection of all virtual documents, d_i is a given document into D, t is a term that belongs to the d_i, and n_{t,d_i} is the number of occurrence of t in d_i.

2. **Computing similarities between two vectors**. The `cosine` similarity measure [8] is established for computing the similarity between two vectors in the space. Let $\vec{v_i}$ and $\vec{v_j}$ be two vectors that represent respectively the two virtual documents d_i and d_j. Their cosine similarity is computed as follow:

$$cos(\vec{v_i}, \vec{v_j}) = \frac{\vec{v_i} \times \vec{v_j}}{\left\|\vec{v_i}\right\| * \left\|\vec{v_j}\right\|} \quad (3)$$

5.1.3. Mapping Construction

For a better insight, we return to Figure 6. At this phase, the virtual documents related to $\mathtt{I}_B$ was built by the `OrgB` and both of them were sent to the `OrgA`. This latter, built virtual documents of its ontology instances that have the attribute shared. Every virtual document building process is performed by following the method presented in subsection 5.1.1. Next, the `OrgA` computed the similarity between their instances virtual documents with the instance $\mathtt{I}_B$ virtual document according to the similarity computation mechanism detailed in subsection 5.1.2.

The final step is to choose the equivalent goal that will be the instance with the high similarity value and having the same trust vector as I_B. In order to fill the five elements in each mapping form, it remains to define the identity `id` and the relation ρ. We remember that we assign to η the similarity value obtained between $\mathtt{I}_B$ and a selected instance which are respectively the elements e_1 and e_2. The identity `id` of each mapping is selected by the recommender. Regarding the relation between the couple of instances whether it is equivalence or disjointness. We recall that two instances are considered as equivalent if they share the same trust concepts, otherwise they are disjoint. Therefore, the `OrgA` checks the virtual documents related to each selected goal and verifies whether it holds the same trust concepts that are belonging to the virtual document of $\mathtt{I}_B$.

6. EVALUATION

6.1. Test Environment

In this section, we show how to apply our solution in a real scenario that is taken from the French project HOSANNA. Our distributed system is composed of four cloud providers. These organizations are sharing their resources based on their interoperability security policies. These organizations are:

- A library, LIB, that manages access to the list of books (managements books, rights of access for students, researchers and teachers, etc.)
- A School Administration, SA, that generates all the data about the courses, working times, the internships and manages the school websites.
- A MOS organization, offers the possibility to use its network infrastructure, provide storage servers, etc. Its security policy is about the access control of its resources, project files, document description, etc.
- TSPlab is a laboratory working in different research projects. Its employees are researchers, PhD students and engineers. Different network infrastructure can be used as a service with this cloud provider. Moreover, some storage servers are also deployed.

6.2. Assumptions

Each organization is implementing its security policy following the OrBAC model. We recall here, that the OrBAC ontologies are automatically created during the specification of the security policy [17].

The evaluation of the trust levels are evaluated based on the solution in [23]. In this approach, authors are defining a framework that evaluates periodically the trust level for the local entities related to an objective. Our framework is using these trust values to construct the trust feedback.

6.3. Communication Process

The communication between the different participants follows two steps: registration and collaboration.

1. Registration step:

- Each organization has to decide to participate or not in the reputation process.
- Each ontology has to be updated with the trust elements. This update was realized based on our enrichment technique. Figure 7 shows the number of the different concepts of the four ontologies proposed in this example.
- The virtual documents of the different instances are prepared based on the methodology defined in Section 5.1.1.

2. Collaboration step:

In the following, we deal with an example of a reputation process. First, a request is sent by the organization LIB to their friends. In this case, the list of friends of LIB contains TSPlab and MOS. This request contains the trustee *Bob*, the goal that is *"save confidential document"*, the virtual documents of the action *"save"* and the view *"confidential document"*.

The reception of this request will trigger the reputation process for TSPlab and MOS. The first step for each organizations is to consult its SGM table in order to check if the equivalent instance of the requested goal was extracted before or not. If no information was found in the

table, the mapping process will start. Otherwise, the trust feedback will be sent directly to the organization LIB.

In our running example, we consider that the two organizations do not find this information in their SGM tables.

6.4. Experiments

In order to evaluate our system, we define two experimental scenarios. The first one is related to the mapping process and the second one aims to test the performance of the reputation mechanism.

1. Mapping based on trust elements Vs Global mapping:

We recall that the mapping process can be realized based on all the entities collected about an instance (called global mapping) or only based on the trust elements. In this scenario, we aim to evaluate the precision of our solution with once by considering only the trust concepts during the build of the virtual documents and one by taking into account all the concepts related to the target instance (Figure 8).

Next, Figure 9 shows the number of equivalent instances for eight goals of the organization LIB. This mapping is realized with the two organizations MOS and TSPlab and with the two mapping options.

Based on these two experiments, we may conclude that the precision of our global mapping technique provides a high level of precision. For this technique, we have obtained a precision more than 0.6 for all the twelve possibilities (Figure 8). However, the second type of mapping that uses only the trust concept can have very bad precision value: This result is logic since this mapping does not take into account the semantic of an instance. But, this second method is very interesting for some particular cases. Indeed, with the global mapping, the number of equivalent instances is always less than the trust element based mapping (see Figure 9). For example, for the goals g3 and g6 with TSPlab and g5 and g7 with MOS, no equivalent instances is given by the global methodology. Therefore the trust solution can be used when the administrator decides to ignore the semantic similarities in order to have some trust feedbacks.

2. The reputation process

Previously, a TrustOrBAC framework was proposed in [23] to integrate a trust evaluator into OrBAC. This solution does not deal with the reputation mechanism. Therefore, we tried in this experience to use our solution with this framework to have a model denoted TrustOrBAC$_{rep}$. Next, we provide some experiments to discuss the relevant properties of our approach with previous solution as: Xena [21] and TrustBAC [27]. The first one aims to establish trust based on the negotiation and the collect of attributes. The second one defines a trust as a combined parameters of experiences, reputation and knowledge. However, the definition of the reputation process was not implemented and detailed.

Figure 10 presents one diagram that illustrates the dynamic responses of the same request sent by the organization LIB in order to perform the goal g3 with four algorithms: Xena (Red curve), TrustBAC (blue curve), TrustOrBAC (Purple curve) and TrustOrBAC$_{rep}$ (green curve). We have repeated several time (during 12 periods) this scenario and we have compared the trust values given by our solution compared to the other three solutions. In this example, we have tried to simulate a malicious behavior of the user. We recall that the assessment of the behavior and the evaluation of the local trust level was realized based on the works in [23].

We conclude that the use of reputation process aims firstly to construct a trust belief during the first periods of collaboration. That means even if no previous work have been realized between the truster and the trustee, our proposition permits to evaluate the trust level of the trustee. When we compare the three curves (blue, green and purple), we remark that only our solution provides trust level during the first four periods thanks to the received feedbacks. Moreover, any organization, using our reputation mechanism, will have a global view about the different entities. For example, the organization will have the possibility to recognize an attacker and react against him faster than the other solutions. We remark in Figure 10 that the bad trust level is always given by our solution. This trust belief was built thanks to the received trust feedbacks.

7. CONCLUSION

We have proposed a new reputation mechanism under the organizational based access control model. We have defined the required trust elements for distributed systems. Then, we have presented an enrichment algorithm to update automatically an OrBAC ontology based on the requirements of our reputation solution. In addition, we designed a new mapping algorithm that permits to share the trust feedback between the organizations. Finally, the proposed methodology has been validated against different experiments compared to other solutions. The experimentation results proof the importance of our solution for the construction of a trust between the organizations, the share of the trust beliefs and the harmony between participants.

As future work, we are planning to (1) update our reputation mechanism to be used between different access control model other than OrBAC (i.e RBAC, ABAC, etc) [33], (2) to integrate our reputation solution into the editor policy tool MOTOrBAC and (3) to study our solution with other distributed systems.

REFERENCES

1. L. Otero-Cerdeira, F. J. Rodriguez-Martinez and A. Gomez-Rodriguez; Ontology matching: A literature review, in Expert Systems with Applications 42(2015):949–971.

2. A. Solimando, E. Jimenez-Ruiz, and G. Guerrini; Detecting and Correcting Conservativity Principle Violations in Ontology-to-Ontology Mappings; in Proceedings of ISWC, 2014.
3. I. F. Cruz, M. Palmonari, F. Caimi, and C. Stroe; Building linked ontologies with high precision using subclass mapping discovery; Artificial Intelligence Revue, 40(2013):127–145.
4. J. Gracia, V. Lopez, M. dAquin, M. Sabou, E. Motta, E. Mena, Solving semantic ambiguity to improve semantic web based ontology matching. 2nd International Workshop on Ontology Matching 2007, 11 Nov 2007, Busan, South Korea.
5. F. Isabel , A. P. Flavio, S. Cosmin, C. Ulas Keles, M. Angela, Using AgreementMaker to Align Ontologies for OAEI 2009: Overview, Results, and Outlook.
6. G. Salton, A. Wong, C. Yang, A vector space model for automatic indexing. Commun ACM 18(1):613–620, 1975.
7. K. Jones, A statistical interpretation of term specificity and its application in retrieval. J Doc 28(1):1121, 1972.
8. P. Tan, M. Steinbach, V. Kumar. Introduction to Data Mining, 1st edn Addison Wesley 2005.
9. I. Cruz, F. Antonelli, C. Stroe, Agreementmaker: efficient matching for large real-world, 2009.
10. M. Komarova, M. Riguidel, Adjustable trust model for access control. In Autonomic and Trusted Computing. Springer Berlin Heidelberg, 200.8.
11. Y. Jean-Mary, E. Shironoshita, M. Kabuka, Ontology matching with semantic verification. Web Semant 7(1):235–251, 2009.
12. D. Ngo, Z. Bellahsene, R.Coletta, Yam++ — A combination of graph matching and machine learning approach to ontology alignment task. J Web Semant, 2012.
13. F. Giunchiglia, M. Yatskevich, P. Avesani, P. Shvaiko P, A large dataset for the evaluation of ontology matching. Knowl Eng Rev J 24(1):137–157, 2009.
14. M. Gulic, B. Vrdoljak, Cromatcher-results for OAEI. 2013. 8th ISWC international workshop on ontology matching, pages 117–122, 2013.
15. M. Mao, Y. Peng, M. Spring, A profile propagation and information retrieval based ontology mapping approach. In: Proceedings of the third international conference on semantics. Knowledge and Grid, IEEE, pages 164–169, 2009.
16. W. Hu, Y. Qu, Falcon-ao: a practical ontology matching system. Web Semant Sci Serv Agents World Wide Web 6(3):237–239, 2008.
17. C. Coma, N. Cuppens-Boulahia, F. Cuppens, and A. R. Cavalli. Context Ontology for Secure Interoperability. In 3rd International Conference on Availability, Reliability and Security (ARES'08), 2008.
18. C. Choi, J. Choi, P. Kim, Ontology-based access control model for security policy reasoning in cloud computing. The Journal of Supercomputing, 67(3):711–722, 2014.
19. T. Y. Chen, Knowledge sharing in virtual enterprises via an ontology-based access control approach, Computers in Industry, Volume 59, Issue 5, pages 502–519, 2008.
20. F. G. Marmol, G. M. Perez, Security threats scenarios in trust and reputation models for distributed systems. computers and security, 28(7), pages 545–556, 2009.
21. D. A. Haidar, N. Cuppens-Boulahia, F. Cuppens, H. Debar, XeNA: an access negotiation framework using XACML. annals of telecommunications, 64(1-2), pages 155–169, 2009.
22. Y. Wang, K. J. Lin, D. S. Wong, V. Varadharajan, Trust management towards service-oriented applications. Service Oriented Computing and Applications, 3(2), pages 129–146, 2009.
23. K. Toumi, C. Andres, A. Cavalli. Trust-orbac: A trust access control model in multi-organization environments. In Information Systems Security, pages 89–103. Springer Berlin Heidelberg, 2012.
24. F. Cuppens, A. Miege, Administration model for OrBAC. In On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops, pages 754–768. Springer Berlin Heidelberg, 2003.
25. F. Cuppens, N. Cuppens-Boulahia, C. Coma, C, O2O: Virtual private organizations to manage security policy interoperability. In Information Systems Security, pages 101–115. Springer Berlin Heidelberg, 2006.
26. A. A. El Kalam, Y. Deswarte, A. Baina, M. Kaaniche, PolyOrBAC: a security framework for critical infrastructures. International Journal of Critical Infrastructure Protection, 2(4):154–169, 2009.
27. S. Chakraborty, I. Ray, TrustBAC: integrating trust relationships into the RBAC model for access control in open systems. In Proceedings of the eleventh ACM symposium on Access control models and technologies, pages 49–59, ACM. 2006
28. N. Dokoohaki, M. Matskin, Effective design of trust ontologies for improvement in the structure of socio-semantic trust networks. International Journal On Advances in Intelligent Systems, 1(1942-2679), pages 23–42, 2008.
29. C. Coma, Interoperability and security policy coherence for self-organizing networks. PhD thesis, 2009.
30. P. T. Chen, C. S. Laih. A challenge based trust establishment protocol for peertopeer networks. Security and Communication Networks, 4(1), 71-78,2011.
31. O. Khalid, S. U. Khan, S. A. Madani, K. Hayat,M. I. Khan, N., MinAllah, D. Chen. Comparative study of trust and reputation systems for wireless sensor networks. Security and Communication Networks, 6(6), 669-688,2013.
32. A.E. Arenas, B. Aziz, G. C. Silaghi. Reputation management in collaborative computing systems. Security and Communication Networks, 3(6), 546-564,2010.

33. K. Toumi, A. Cavalli, M. El Maarabani. Role based interoperability security policies in collaborative systems. In the international conference of the Collaboration Technologies and Systems (CTS 12), IEEE, 2012.
34. J. Golbeck, B. Parsia,J. Hendler, Trust networks on the semantic web, pages 238–249. Springer Berlin Heidelberg, 2003.
35. S. Toivonen, G. Denker, The Impact of Context on the Trustworthiness of Communication: An Ontological Approach. In ISWC Workshop on Trust, Security, and Reputation on the Semantic Web (Vol. 127), 2004.
36. R. Beckwith, C. Fellbaum, D. Gross, G. A. Miller, WordNet: A lexical database organized on psycholinguistic principles. Lexical acquisition: Exploiting on-line resources to build a lexicon, pages 211–231, 1991.
37. R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swartout. Enabling technology for knowledge sharing. AI Magazine 12(3), pages 36–56, 1991.

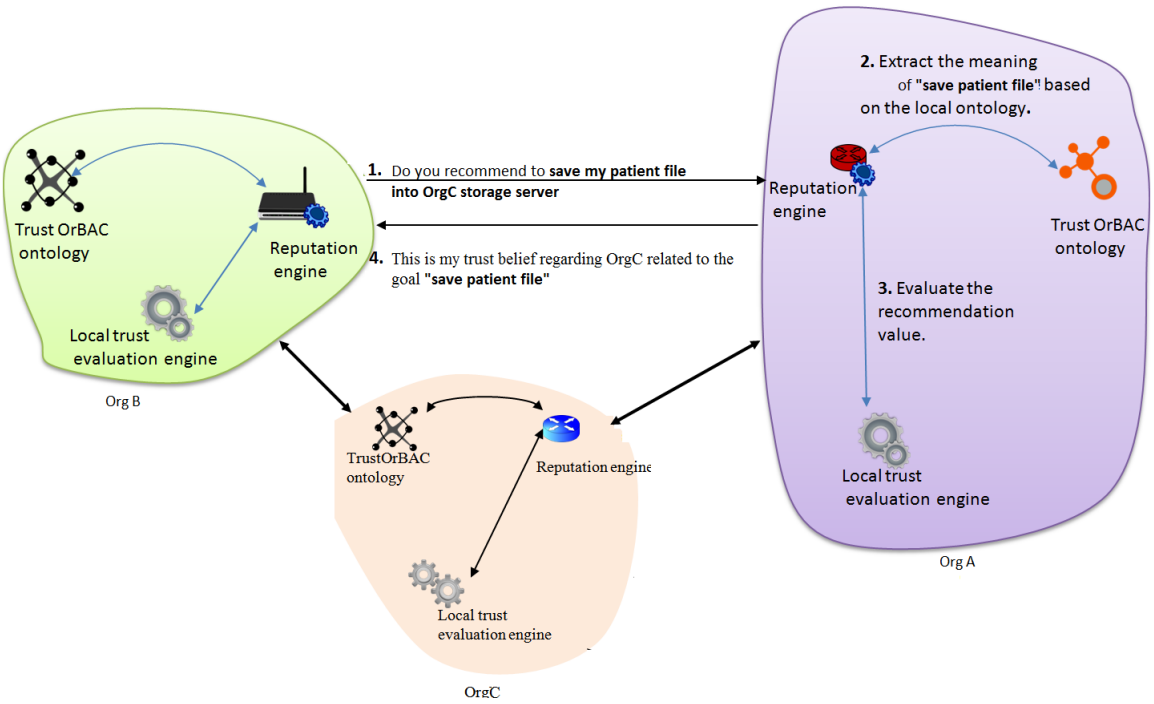


Figure 4. An example of a reputation request.

	Activities numbers			View numbres			Context numbers				Number of goals
TSPlab	7			8			5				280
SA	6			9			3				162
MOS	4			5			5				100
LIB	5			7			6				210
	E.A	R.A	W.A	P.V	C.V	S.V	I.C	N.C	P.C	U.C	
TSPlab	2	2	3	4	2	2	2	3	4	1	
SA	3	2	1	5	3	1	2	1	1	2	
MOS	2	1	1	1	1	3	4	1	2	3	
LIB	2	2	1	4	2	1	2	4	3	3	

E.A: Executing Activity | R.A: Reading Activity | W.A Writing Activity | P.V: Public View | C.V: Confidential View | S.V: Secret View | C.I: Important Context | N.C: Normal Context | P.C: Private Context | U.C: Unprivate Context

Figure 7. The trust ontology components of the different organizations

Security Comm. Networks 2015; **00**:1–13
DOI: 10.1002/sec
Prepared using secauth.cls

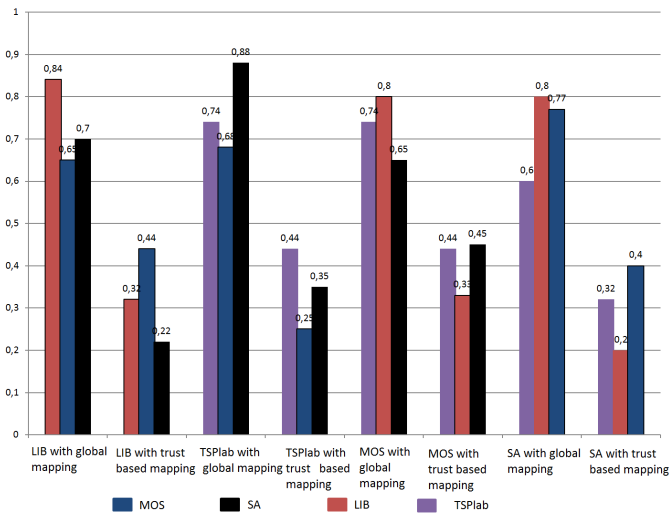
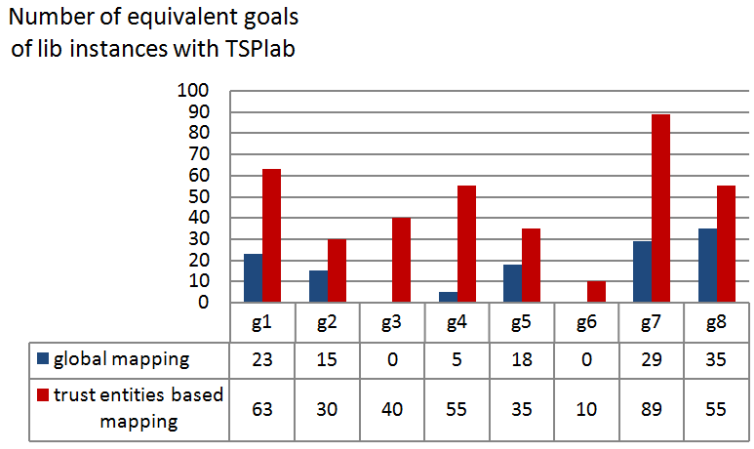
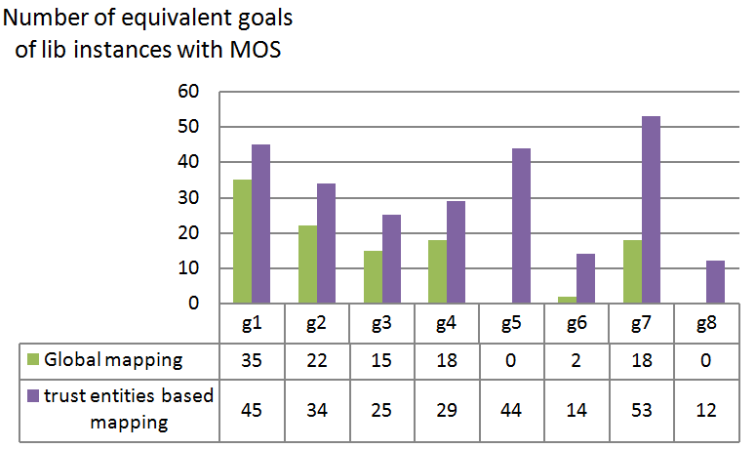


Figure 8. Precision value for our mapping solution



	g1	g2	g3	g4	g5	g6	g7	g8
global mapping	23	15	0	5	18	0	29	35
trust entities based mapping	63	30	40	55	35	10	89	55



	g1	g2	g3	g4	g5	g6	g7	g8
Global mapping	35	22	15	18	0	2	18	0
trust entities based mapping	45	34	25	29	44	14	53	12

Figure 9. number of equivalent goals with TSPlab and MOS organizations

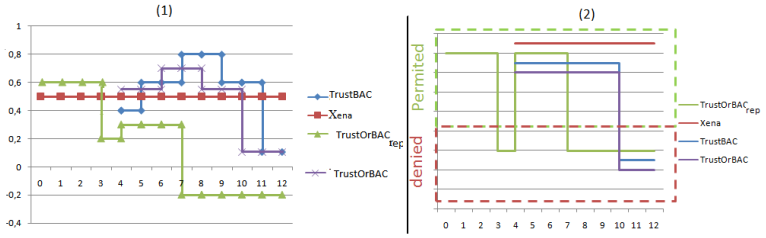


Figure 10. The use of the reputation mechanism with TrustOrBAC

Security Comm. Networks 2015; **00**:1–13
DOI: 10.1002/sec
Prepared using secauth.cls