Enhancing access control for mobile devices with an agnostic trust negotiation decision engine*

Daniel Díaz-Sánchez, Andrés Marín, Florina Almenárez

Telematic Engineering Department, Carlos III University of Madrid
Avda. Universidad, 30, 28911 Leganés (Madrid), Spain
{dds, amarin, florina}@it.uc3m.es

Abstract. Dynamic open environments demand trust negotiation systems for unknown entities willing to communicate. A security context has to be negotiated gradually in a fair peer to peer basis depending on the security level demanded by the application. Trust negotiation engines are driven by decision engines that lack of flexibility: depend on the implementation, policies languages or credentials types to be used. In this paper we present an agnostic engine able to combine all that information despite its origin or language allowing to select policies or requirements, credentials and resources to disclose, according to user preferences and context using iterative weighted Multidimensional Scaling to assist a mobile device during a trust negotiation.

Key words: trust negotiation, access control, flexible

1 Introduction

"Access Control" requires to determinate if an entity is entitled to use a service or not. Moreover, it should decide other parameters as which quality of service should be granted to an entity. Determining wether a user or entity can access or not to a resource, can be simplified in finding an answer to the question: "Can entity E perform action A on resource R?".

The answer can be found in different ways but in general comprises authentication, authorization and policy enforcement. Moreover, the requirements for access control depend on the context of usage, application and the sensitivity of the resources to be accessed and the credentials to be disclosed.

Different Access Control systems have been proposed: Mandatory Access Control (MAC),Access Control Lists (ACL), Role Based Access Control (RBAC) and some recent XML based efforts like eXtensible Access Control Markup Language [1]. Besides, different credentials are used: key centric bindings as KeyNote described in RFC 2704, SPKI in RFC 2693; and unique name binding credentials as Public Key Infrastructure (PKI) in ITU Recommendation X.509 or RFC

* This work has been partially supported by the Spanish Ministry of Science and Education throught the ITACA project (TSI2006-13409-C02-01).

Please use the following format when citing this chapter:

Díaz-Sánchez, D., Marín, A., Almenárez, F., 2007, in IFIP International Federation for Information Processing, Volume 245, Personal Wireless Communications, eds. Simak, B., Bestak, R., Kozowska, E., (Boston: Springer), pp. 304-315.

3280, Privilege Management Infrastructure (PMI) in RFC 3281 or XML based like Security Assertion Markup Language (SAML) [2].

Modern distributed authorization models employ the Role Based Access Control (RBAC) that uses roles instead of identities. Roles can be expressed with PKCs [3], ACs ITU Recommendation X.509 that suffers from name-binding. SAML and other schemas [4] can be used also but suffer from limitations of key-centric approaches, or do not provide desirable definitions as separation of duties or authorization management as explained in [5].

This work presents a solution for assisting users to select policies or requirements, credentials and resources to disclose, according to their preferences and context, during a P2P trust negotiation. The work described here uses a human-mimicking decision engine able to simplify problems and to graphically present those problems to the user in a way that allows he/she to understand what is occurring despite his/her technical training.

1.1 Trust negotiation

Section 1 describes some authentication and authorization mechanisms that assume that every involved entity is known and trusted to the system in advance, for instance, a Certification Authority, the entity that vouches for an identity in PKI, should be trusted or the role/group should be accepted by both parties.

Promising efforts on trust negotiation as [6][7] allow **strangers to negotiate trust**, disclosing credentials, and even properties-based credentials (based on properties of the user rather than identities and capabilities). Thus, a stranger can be authenticated and authorized. Trust negotiation systems are based on the fact that any resource is protected by a policy that sets which credentials should be disclosed to obtain access to it. In [8] some requirements that a trust negotiation system must satisfy are described. Policies play an important role in trust negotiation, [6] recognizes that policies should be disclosed gradually according the level of trust reached since contain sensible information. Besides, rogue peers might build policies that force other entities to disclose more credentials and information than the necessary, so the credential disclosure should be done gradually in a peer to peer basis: providing only the necessary credentials to the peer holding the resource and asking the resource holder for credentials if more information than the necessary is required. Policy and credential disclosure should be driven by a decision engine to ensure the fairness of the process.

Systems supporting different credentials together with new trust negotiation systems are on the road to success since even strangers can be authenticated and authorized. This is the cornerstone for a real peer to peer secure interaction.

1.2 Multidimensional Scaling

Multidimensional Scaling [9], MDS, is a set of techniques widely used in behavioral, psychological and econometric sciences to analyze similarities of entities. From a pairwise dissimilarities matrix, usually m-dimensional Euclidean distances [10], MDS can be used to represent the data relations faithfully providing

a geometrical representation of these relations. MDS is used to reduce the dimensionality of a problem to a small value. MDS techniques have been used for several problems with good results: to determine the distance among elements of sensor networks [10], to classify music, browse it and generate playlists [11] or to derive an *interaction distance* measure for network selection [12].

MDS can consider not only Euclidean distances but also any other evaluation of dissimilarities: qualitative or quantitative. The dissimilarities from attributes of data can be weighted (weighted MDS), thus, assigning a different weight to each attribute allows to obtain more particular results depending on the problem. So, a complex m-dimensional problem can be simplified preserving the essential information using MDS.

In classical scaling the proximities are treated as distances, however, any (di)similarity can be derived from data attributes in order to obtain a metric: in case of ordinal data, another procedure has to be followed than the use of singular value decomposition since we want to recover the order of the proximities and not the proximities or a linear transformation of the proximities. A solution to this problem was given by Shepard [13] and refined by Kruskal [14]. These solution iteratively minimize a fit measure called *Stress* by an iterative algorithm, which is suitable for processing.

We have used an algorithm called ALSCAL [15], which uses alternate least-squares, combined with weighted (di)similarities, can combine both metric and nonmetric analysis and can also deal with spare proximity matrixes so it is suitable in the absence of some data.

1.3 Article organization

Trough this section we have described the related work, section 2 introduces the *Agnostic trust negotiation decision engine* starting with a set of definitions in section 2.1 and the architecture in section 2.2. Moreover, sections 2.3 and 2.4 describe how to classify, extract and combine security and context information and also how to derive (di)similarities. Section 3 shows results for an example and finally we summarize the goals achieved in 4.

2 Agnostic trust negotiation decision engine

Along this section we will introduce algorithms to combine data that can be used to assist the user during a trust negotiation. We will use an example to make the algorithms easier to understand.

2.1 Definitions

A negotiation process involves information disclosure between peers according to a strategy that warranties that the process is fair for all parts. The strategy avoids rogue peers asking for unneeded credentials to reveal sensible information.

To help the reader to understand the article, a set of definitions are provided here:

Policies are pieces of data issued by a resource manager, a domain administrator or a provider. A **resource** can be protected by more than one policy. Those policies should be combined to obtain requirements. A **requirement** represents the information to be disclosed in order to satisfy part of a policy or combined set of policies. So from a policy or set of policies a requirement or set of requirements can be derived. A **policy item** is a formal definition for a requirement that can be used by other peers to find out which credential should be disclosed in order to satisfy a requirement. A **credential** is a piece of information to be disclosed to satisfy a requirement. A **resource** is any information, service or mechanism which its disclosure implies a risk. Credentials are also considered resources and should be protected by policies (so some requirements should be satisfied to be accessed).

2.2 Architecture

To take access control decisions the information available at a given instant of time and the context, should be taken into account. The context defines for instance, the connection speed of the available network interfaces, the location, the level of battery... Thus, given a context, a resource is described by its own properties or attributes and the constraints that the context imposes.

Consider a mobile device governed by a set of policies. The policies are written by the user, the domain administrator of the user's company and the UMTS provider. Those policies are controlled by different access control engines (ACEs). Every ACE processes the policy or set of policies it understands and extracts **requirements** and **policy items**. Then they register requirements and policy items to the decision engine. Figure 1 shows the architecture.

2.3 Merging policies, context and constraints

In this section we show how different requirements, extracted from different policies in different languages, can be combined using a single decision engine. Policies might be written in any language and might be general enough to cover all the possible types of users over a domain (mobile network operator or company policies). (ACEs) process the policies and extract the requirements. We require the Access Control Engines (ACEs) (possibly with the aid of the user) to be able to extract the part of each policy which applies to the device during setup (first time used).

We will use an example in which the ACEs extract four pieces (policy items) of the policies (P_n), and four requirements Rq_n. The policy items can be sent to the other part so the other peer can find out what credentials should disclose to satisfy a requirement. Policy items should be also protected. Disclosing a policy implies some risk, for instance, a policy item asking for a credential issued from a bank that asserts user's account balance might be disclosed only if other requirements has been previously satisfied. In the example, resources are named

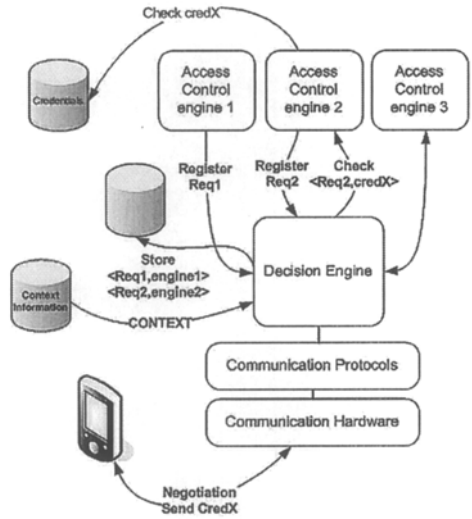

Fig. 1. Access control structure

C_n with n ranging from A to E for credentials and R_n with n ranging from A to H for other resources (not credentials). The table 1 illustrates the example.

Besides security attributes, properties of resources are also considered. In the example (table 1), **P/W** distinguishes among resources that can be accessed for personal (0) or work (1) usage. **RT** classifies resources as web-services (0), ftp (1), file sharing (2), and agenda resources (3). **Loc** specifies the resource location: 1 at mobile device and 0 outside mobile device (for situations where the device acts as a proxy). **CE** express battery consumption (0 for no consumption or connected to power line).

2.4 Computing dissimilarities

As can be seen in Table 1 we define a new item in that table to represent the negotiation (**Neg**). The attributes of **Neg** vary during a negotiation, thus, deriving dissimilarities including the negotiation makes possible to determine which resources are near the **Neg** point. After that calculation, the closer a resource is to the negotiation, i.e. its disclosure implies less risks.

(Di)Similarities between pairs of elements (rows in the table) can be derived as follows:

$$\delta_{i,j,\alpha} = \frac{|u_{i,\alpha} - u_{j,\alpha}|}{max(u_\alpha) - min(u_\alpha)}, \text{ for quantitative data} \quad (1)$$

$$\delta_{i,j,\alpha} = \frac{|rank(u_{i,\alpha}) - rank(u_{j,\alpha})|}{max(rank(u_\alpha)) - 1}, \text{ for ordinal data} \quad (2)$$

$$\delta_{i,j,\alpha} = \begin{cases} 0 : u_{i,\alpha} = u_{j,\alpha} \\ 1 : otherwise \end{cases}, \text{ for membership data} \quad (3)$$

	Reqs.	Loc.	P/W	RT	EC
Neg	Variable	1	V	V	V
P_1	none	1	U	U	U
P_2	Rq_1	1	1	U	U
P_3	Rq_2	1	1	U	U
P_4	$Rq_2 \& Rq_3$	1	0	U	U
C_A	Rq_1	1	0	U	U
C_B	$Rq_1 \mid Rq_2$	1	U	U	U
C_C	Rq_2	1	1	U	U
C_D	Rq_1	1	0	U	U
C_E	$Rq_1 \mid Rq_2 \mid Rq_4$	1	0	U	U
R_A	$Rq_1 \& Rq_2$	1	1	0	0
R_B	Rq_1	1	0	3	0
R_C	$Rq_1 \mid Rq_4$	1	1	3	0
R_D	$Rq_1 \& Rq_3 \& Rq_4$	1	1	0	0
R_E	$Rq_1 \& Rq_3 \& Rq_4$	1	0	2	0
R_F	$(Rq_1 \mid Rq_2) \& (Rq_3 \mid Rq_4)$	1	1	2	0
R_G	$Rq_1 \mid Rq_3$	1	1	2	0
R_H	Rq_1 & $Rq_2 \& Rq_3$	1	0	0	0

Table 1. Attribute values in a possible decision scenario. U:Unspecified, V:Variable

where $u_{i,\alpha}$ is the α^{th} attribute value of element i (policy item, credential or resource). We consider different types of data: quantitative (for trust relations [16] and distances [10]); ordinal (for QoS classes, and service differentiation); membership (to distinguish credential types).

Table 1 shows logical operators that are used to combine requirements. Sequences of requisites combined with logic operators (as *and,or*) cannot be compared using equations 3,4 and 5. For that reason we define the equation below to compare expressions:

$$\delta_{i,j,Rq_\alpha} = \begin{cases} 0: \ if \ f_{i,Rq_n} \neq f(Rq_\alpha) \\ 0: \ if \ f_{j,Rq_n} \neq f(Rq_\alpha) \\ eval(f_{i,Rq_n}) \ \& \ eval(f_{j,Rq_n}) : otherwise \end{cases} \tag{4}$$

where f_{i,Rq_n} is the logical function that combines the requirements satisfied during the negotiation for element i. $f_{i,Rq_n} \neq f(Rq_\alpha)$ means that the logical function that combines the requirements for element i does not depend on requisite Rq_α. $eval(f_{i,Rq_n})$ returns the result of evaluating the logical function using as parameters the requirements satisfied during the negotiation.

Once the (di)similarities are calculated, they are weighted (according to user preferences or context) in order to obtain an unique weighted (di)similarities matrix (weighted MDS, see 1.2). These weighted (di)similarities are defined for a set of n objects with q attributes as follows:

$$\delta_{i,j} = \left(\frac{\sum_{\alpha=1}^{q} w_{i,j,\alpha} w_\alpha \delta_{i,j,\alpha}^{\lambda}}{\sum_{\alpha=1}^{q} w_{i,j,\alpha} w_\alpha} \right)^{\frac{1}{\lambda}} \tag{5}$$

where $w_{i,j,\alpha}$ takes value 0 if objects i and j can not be compared on the α^{th} attribute and 1 otherwise, w_α is the weight given to attribute α and $\delta_{i,j,\alpha}$ is the (di)similarity between objects i and j on the α^{th} attribute.

The first element represents the negotiation in a time t and will be used to measure the *distance* from the negotiation to the resources: the nearer a resource is from the negotiation, the less risk the user experience. Even more, it is not possible to assign values to every attribute of every element during a negotiation. For example, the *type of service* attribute of a credential is **undefined**. MDS is suitable here due to the fact that is able to work in absence of some data.

Another key element in our model is the weights vector. The following equation shows the weights calculations. Table 2 gives the weights for the example at different instants of time:

$$w_\alpha = \begin{cases} \alpha \in Rq_n : \begin{cases} 0 : if\ \alpha \in \textbf{Neg} \\ \frac{k}{ua} : if\ \alpha \notin \textbf{Neg} \end{cases} \\ \alpha \notin Rq_n : \begin{cases} 0 : if\ \alpha \notin \textbf{Neg} \\ \frac{k}{ua} : if\ \alpha \in \textbf{Neg} \end{cases} \end{cases} \tag{6}$$

w_α is weight for α^{th} attribute. A weight value allows to express how important is a given attribute compared to the others. During the first round we use the same value for the unspecified attributes. The reader should note that a requirement cannot be unspecified: if not fulfilled a requirement is equal to 0 (false) and it is considered unspecified for weight calculation.

We maintain constant the sum of weights, so as long as the other peer provides credentials (fulfilling requirements) and the negotiation item becomes more specified, we uniformly spread the value among the attributes: if the attribute represents a requirement and it is already unspecified, we cannot grant access to any resource protected by that requirement, so we give to that attribute a weight of $\frac{k}{ua}$. k is a constant and ua the number of unspecified attributes.

Moreover, if the attribute represents a requirement but it has been fulfilled, the weight given to that attribute is 0. Otherwise, for attributes that represents properties of resources, we give them the value of 0 if unspecified and $\frac{k}{ua}$ if specified.

At this point we had the necessary data to run the MDS algorithm. We solved for two dimension and set $\lambda = 2$ to handle attributes as euclidean distances.

2.5 Computing risk limits

We use MDS to reduce the complexity of the problem to a visible number of dimensions, so we are able to graphically present to the user the decision space. Thus, the user is aware of the risk that involves a given interaction. Typically, users do not spend many time checking, for instance, website's certificates or other credentials, furthermore, when a user is prompt to accept or not a given credential, usually he/she accepts without wondering about the risks. A graphical

t	Rq_1	Rq_2	Rq_3	Rq_4	**Loc**	**P/W**	**RT**	**EC**
$t = 0$	0	0	0	0	1	1	Un.	0
$w[\,]$	1.16	1.16	1.16	1.16	1.16	1.16	0	0
$t = 1$	1	0	0	0	1	1	Un.	0
$w[\,]$	0	1.4	1.4	1.4	1.4	1.4	0	0
$t = 2$	1	1	0	0	1	1	Un.	0
$w[\,]$	0	0	1.75	1.75	1.75	1.75	0	0
$t = 3$	1	1	1	0s	1	1	Un.	0
$w[\,]$	0	0	0	2.33	2.33	2.33	0	0

Table 2. weight calculation during the negotiation. Requirements not fulfilled have 0 value. $K(t = 0) = 7$.

presentation of the problem can be useful to be aware of the services that are been exposed to outside: the user can see which resources are similar, in terms of requirements, so he/she becomes aware of the resources affected by a decision.

Once a two dimensional presentation of resources has been obtained, we consider necessary to define a limit for the accepted risk: the risk can be displayed as a circle whose center is the **Neg** point and the radius depends on the context. Resources inside that circle can be disclosed since the risk is inside the accepted limits. But, how can we define that limit? The most restrictive approach should defend that resources can be disclosed only if their distance to the **Neg** point is 0.0. However, being some of the attributes unspecified, an exact match is likely difficult to achieve. We consider that the risk should vary depending on the context in the following fashion: the less defined a negotiation is, the more attributes are undefined and the biggest are the dissimilarities, so the bigger are the distances between the **Neg** and the resources, thus the more risk can be assumed since the less points will be inside the circle. To derive a value for the risk circle radius we propose the following equation:

$$radius = \begin{cases} 0 \; if \; ua < uaMin \\ (\frac{ua}{attrNum}) * (\frac{maxDist}{attrNum}) \; otherwise \end{cases} \tag{7}$$

where $maxDist$ is the maximum distance, $attrNum$ the number of attributes, ua the number of unspecified attributes and $uaMin$ the minimum number of attributes that should be specified to derive a radius different from 0.

3 Proof of concept

In this section we present the results of a negotiation. The resources to be analyzed have been already shown in Table 1. The negotiation advances in the following fashion: peer A, the one that holds the resources, discloses policy items, that express requirements, and peer B discloses credentials to fulfill those requirements. The negotiation evolves as displayed in table 2: includes both the requirements fulfilled at a given instant of time and the evolution of dissimilarities weights.

At $t = 0$ peer B tries to access a resource held by peer A. Peer A's decision engines has registered several requirements that affect that resource: the resource is disclosable only to company's employees, so it turns the attribute **P/W** to 1 and *Loc* to 1 (located at the mobile device). Since battery is full charged and there is no restriction about the type of resource **RT** and **EC** remains undefined. No requirement has been already fulfilled, so the values for Rq_n are 0. The mobile device compute the weights (see table 2) for $t = 0$. Then the mobile device compute dissimilarities and simplify the problem to a two dimensional problem. Figure 2 shows the decision space and table 3 shows the distances.

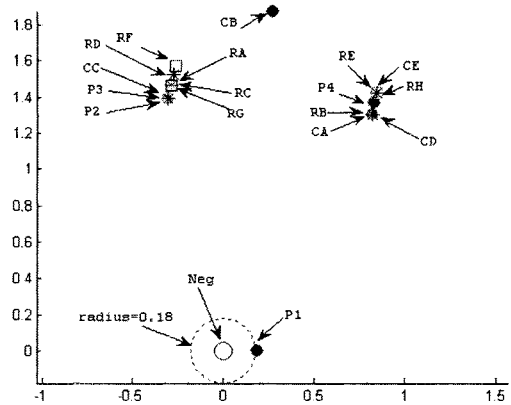

Fig. 2. Negotiation space at $t = 0$

As can be seen in figure 2, P_1 is the only resource that fits into the risk circle, so it can be disclosed. The rest of the resources form two groups, the first with the center located approximately at $[0.4, 1.5]$ is composed by the resources that can be disclosed for work purposes. The other group is composed by the resources that can be disclosed only for personal purposes.

Figure 3 shows the decision space at $t = 1$, just when the other peer discloses credentials to fulfill Rq_1. Under this condition, peer A can disclose P_2, R_G and R_C since those resources can be disclosed for work and requires Rq_1 to be satisfied. Resources C_E,C_A,C_D and R_B require also Rq_1 to be satisfied, but are separated enough from the **Neg** since they can be disclosed only for personal issues. Resources R_A,P_3 and C_C are grouped together since depends on Rq_2 to be fulfilled and should be disclosed only for work purposes. Despite resources R_H,R_E and P_4 can be disclosed only for personal issues as C_E,C_A,C_D and R_B, they form a different group, separated even more from **Neg**, since they have more complex security requirements. Moreover, resources R_D and R_F are located closer to the group of P_3 since they depend on Rq_2 also.

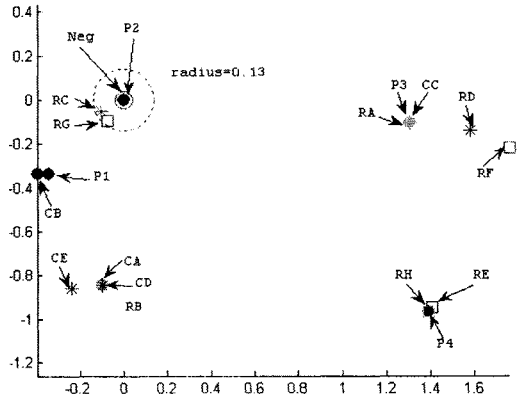

Fig. 3. Negotiation space at $t = 1$

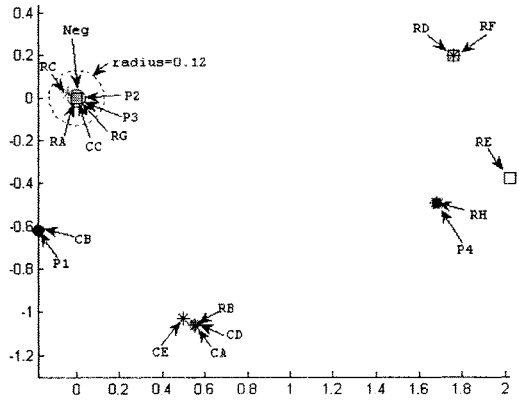

Fig. 4. Negotiation space at $t = 2$

Decision space at $t = 2$ is represented in figure 4. Rq_2 has been disclosed: the group of resources R_A, P_3 and C_C is now inside the circle, so can now be disclosed. The rest of resources remains far from the **Neg** element due to either their requirements or their personal nature.

In Figure 5 it can be seen that once peer B fulfill Rq_3, the resource R_F becomes available, and the rest remain far from the center. P_4 can not be disclosed since it is personal and, for that reason, many resources that depend on Rq_4 are not disclosed.

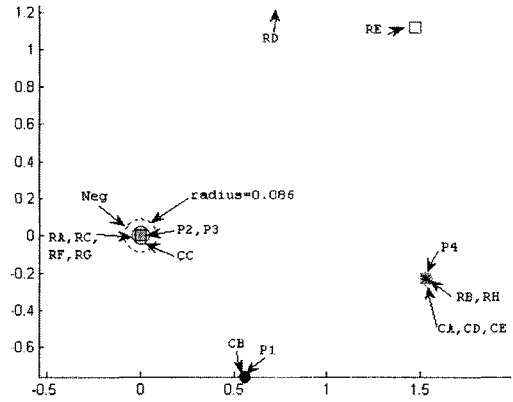

Fig. 5. Negotiation space at $t = 3$

t	**Neg**	P_1	P_2	P_3	P_4	C_A	C_B	C_C	C_D
0	0	0.18	1.42	1.42	1.60	1.53	1.89	1.42	1.53
1	0	0.48	0.00	1.30	1.69	0.84	0.51	1.30	0.84
2	0	0.64	0.00	0.00	1.75	1.19	0.64	0.00	1.19
3	0	0.94	0.00	0.00	1.54	1.54	0.94	0.00	1.54

t	C_E	R_A	R_B	R_C	R_D	R_E	R_F	R_G	R_H
0	1.65	1.49	1.53	1.49	1.54	1.65	1.59	1.49	1.65
1	0.89	1.30	0.84	0.11	1.58	1.69	1.77	0.11	1.69
2	1.14	0.00	1.19	0.04	1.76	2.05	1.76	0.00	1.75
3	1.54	0.00	1.54	0.00	1.44	1.84	0.00	0.00	1.54

Table 3. Distances

4 Conclusions

Through this paper we have demonstrated how simple, easy to understand by the user and strong is our decision engine. It is simple since we use just a distance to find the resources that can be disclosed, we treat resources, credentials and policies in the same way (agnostic). Thus, if during the negotiation, B changes the role and starts asking for credentials to peer A, to protect itself against rogue peers, peer A can use the results of the engine to determinate whether a credential can be disclosed to B or not.

The decision engine is easy to understand by the user and minimize the risks since the user is now aware of the consequences: the user can see which group of resources is affected by a decision so he/she can consider a larger picture just having a look to the resource clustering. Our decision engine is strong: no resource is disclosed unless every requirement is fulfilled.

We have also demonstrated that there is no need to find a common model to represent every element involved in a negotiation: despite its language, origin, and attributes, we provide a mechanism able to work even with unspecified data.

We propose also some rules to adopt logic operators and to derive a risk limit that depends also on the context. Furthermore, access control engines that, instead of returning true or false when verifying a credential, returns a continuous value can be used also just computing those values as shown in equation 3.

References

1. OASIS: eXtensible Access Control Markup Language (XACML) (2003) http://www.oasis-open.org/apps/org/workgroup/xacml/.
2. Mishra, P.: Saml v2.0 oasis standard specification. Technical Report SAML v2.0, OASIS Security Services TC (2005)
3. Herzberg, A., Mass, Y., Michaeli, J., Ravid, Y., Naor, D.: Access control meets public key infrastructure, or: Assigning roles to strangers. In: SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy, Washington, DC, USA, IEEE Computer Society (2000) 2
4. Chadwick, D., Otenko, A.: The PERMIS X.509 role based privilege management infrastructure. Future Generation Computer Systems **19** (2003) 277–289
5. Bhatti, R., Bertino, E., Ghafoor, A.: An integrated approach to federated identity and privilege management in open systems. Commun. ACM **50** (2007) 81–87
6. Squicciarini, A.C.: Trust negotiation systems. In: EDBT Workshops. (2004) 90–99
7. Bertino, E., Ferrari, E., Squicciarini, A.: X -tnl: An xml-based language for trust negotiations. policy **00** (2003) 81
8. Bertino, E., Khan, L.R., Sandhu, R., Thuraisingham, B.: Secure knowledge management: confidentiality, trust, and privacy. Systems, Man and Cybernetics, Part A, IEEE Transactions on **36** (2006) 429–438
9. Borg, I., Groenen, P.: Modern multidimensional scaling, theory and applications. In: IEEE SECON 2004, New York, NY, USA, Springer-Verlag (1997)
10. Shang, Y., Ruml, W., Zhang, Y., Fromherz, M.P.J.: Localization from mere connectivity. In: MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, New York, NY, USA, ACM Press (2003) 201–212
11. Platt, J.C.: Fast embedding of sparse music similarity. In: Advances in Neural Information Processing Systems vol. 16. (2004)
12. Díaz, D., Marín, A., Alménarez, F., García-Rubio, C., Campo, C.: Context awareness in network selection for dynamic environments. In: 11th IFIP International Conference on Personal Wireless Communications "PWC'06", Springer (2006)
13. Shepard, R.N.: The analysis of proximities: multidimensional scaling with unknown distance function part i. In: Psychometrika 27. (1962)
14. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. In: Psychometrika 29. (1964)
15. Takane, Y., Young, F.W., de Leeuw, J.: Nonmetric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features. In: Psychometrika 42. (1977)
16. Almenárez, F., Díaz, D., Marín, A.: Secure Ad-hoc mBusiness: Enhancing WindowsCE security. In: 1st Conference on Trust Digital Business (TrustBus'04). (2004)