Graphical Representation of Authorization Policies for Weighted Credentials

Isaac Agudo, Javier Lopez, and Jose A. Montenegro

Computer Science Department, E.T.S. Ingenieria Informatica
University of Malaga, Spain
{isaac, jlm, monte@lcc.uma.es}

Abstract. This paper elaborates on a solution to represent authorization and delegation in a graphical way, allowing users to better interpret delegation relationships. We make use of Weighted Trust Graph (WTG) as an instrument to represent delegation and authorization, extending it to cope with more complicated concepts, and providing a graphical representation of the level of confidence that exists between two entities regarding a resource or attribute. We represent the level of confidence for each pair of entities as a point in an axis diagram, as a set of points, or as a set of triangular regions depending on the accuracy we need. Then, we use the same diagram to represent the set of acceptable confidence level, that we call authorization policy set. In this way, a single diagram can be used to decide about authorization, thus providing a powerful tool for systems in which interaction of users is needed.

1 Introduction

Logic programming offers a nice mechanism to represent authorization and delegation statements and decisions (see [5, 6, 2] for a list of examples) Statements are represented as predicates and decisions are based on formulae verification. There are many logical solutions for formulae verification and it is not difficult to implement them. However, one disadvantage of logical programming is that it is not easy to understand and has an obscure transcription. Moreover, the syntaxis of the different solutions are not homogeneous and, as a consequence, the learning process of the syntax can be quite hard. When trying to use logic directly, one has to deal with too many details that might be avoided if one makes use of a more user-oriented solution. In some sense, logic could be useful in a second stage, that is, not in the specification phase but in the analysis or decision-making phase.

On the other hand, there are solutions that, though less powerful, are more expressive and easier to understand. One of them is to use graphs, and in particular, *directed graphs*. Proposals that make use of directed graphs to model authorization and delegation statements use to map each predicate to a directed arc in a graph. Arcs go from the issuer of the authorization or delegation statement to the subject who is granted privileges. Thus, the graph includes as many different arcs as different authorization/delegation statements are considered.

The result is a tree where the root usually is the owner of the resource we are reasoning about. That tree helps to understand the relations among entities in the system in a graphical way.

Varadharajan et al. have proposed two solutions to represent authorization and delegation using directed graphs. In [3] they presented a basic approach that support graphical representation of positive authorization, negative authorization and delegation. This solution follows the *predecessor-take-precedence* policy to resolve conflicts between positive and negative authorization. In [4] they presented a more elaborated proposal that makes use of integer numbers to assign a certainty level to each credential.

Weighted Trust Graphs (WTG), presented in [1] aims to generalize this proposal, defining it in a more flexible way. In fact, that proposal is supported as a particular case of WTG. WTG follows the predecessor-take-precedence policy with some refinements and a security level policy. Also, it proposes a new conflict resolution method, *strict-predecessor-take-precedence*. It means that the owner of the resource establishes a hierarchy of subjects and any of the further delegations made for these subjects has to preserve this hierarchy. For instance, if A gets from S the higher priority in the hierarchy, all his statements take preference over the others ones.

The aim of this paper is to present a graphical representation of both authorization policies and the trust on delegation and authorization credentials. It also elaborates on the idea of acceptance levels for authorizations, defining new indexes that take into account only a given percentage of credentials (but not all of them). We give some guidelines for the definition of authorization policies and provide, as mentioned, a graphical representation.

The rest of the paper has the following structure. In Section 2 we present a revision of WTG, with some interesting changes with respect to the original work. Section 3 offers new definitions that make WTG more suitable for certain environments, providing alternatives to the original definitions. Section 4 focuses on the concept of authorization policies and we provide an original graphical representation of them. Section 5 ends with some conclusions.

2 Weighted Trust Graph

In this section we examine WTG, overviewing its main ideas, what will help to understand better the concepts that will be explained in the following sections.

In WTG, credentials are represented using arcs in a graph, so both terms are used likewise. A credential is a 4-tuple: $(Issuer, Subject, Type, Right)$, where the first component is the issuer of the authorization or delegation statement; the second one is whom the statement refers to; the third is the type of the statement; and, finally, the fourth is the right together with the resource we are reasoning about.

In fact, $Right$ can be represented as a 2-tuple consisting of the resource and the kind of access, thus $Right = (Resource, Access)$. It must be noted that $Type$ can be expressed as a 3-tuple composed of the following parameters:

- *Weight*, which represents the level of trust in this authorization. It is a number in the interval $[0, 1]$. A credential with weight zero is equivalent to a null credential.
- *Delegation*, which represents whether it is a delegation statement or not.
- *Sign*, which represents the sign of the statement (either negative or positive). Negative credentials may override positive ones and the other way around, depending on the weight.

According to that presentation, WTG defines four types of credentials: positive delegation, positive authorization, negative delegation and negative authorization, that are graphically represented in figure 1.

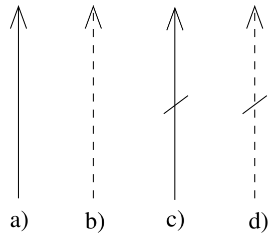


Fig. 1. Representation of credentials

Suppose we are reasoning about an attribute a and we have two principals involved: Alice, the grantor or issuer of the statement and also the owner of the administrative right over attribute a, and Bob, the subject of the statement.

The simplest type of credential corresponds to the concept of Authorization. In this situation Alice owns attribute a and she can issue statements regarding attribute a and attribute $\neg a$. It is important to note that, even if a and $\neg a$ are different attributes, they are very related so that the authority about a should imply the authority about $\neg a$. For instance, let's think about the case of the attributes $Female$ and $Male$. They are complementary attributes, which means that $Male := \neg Female$. In this case, positive authorization regarding $Male$ is equivalent to a negative authorization regarding $Female = \neg Male$.

As a result, although we can easily include both attributes a and $\neg a$ in an authorization chart using positive and negative authorizations, we can not do it in a delegation chart because we would require six different arcs. This leads us to the situation of having a more simple graphical representation with only the four types of statements depicted in Figure 1, but with a meaning different to the original WTG:

a) *Positive delegation.* A positive delegation about both a and $\neg a$.
b) *Positive authorization.* A positive authorization about a or a negative Authorization about $\neg a$.
c) *Negative delegation.* A negative delegation about both a and $\neg a$.

d) *Negative authorization.* A negative authorization about a or a positive Authorization about $\neg a$.

When making decisions regarding authorization to perform certain operations over a resource, it is necessary to consider all the chains or paths of credentials from the owner of the resource to the specific subject. WTG defines paths of credentials as sequences of consecutive credentials, distinguishing between *delegation paths* (those in which there are only delegation credentials), and *authorization paths* (delegation paths followed by a single authorization credential). Only credentials regarding the same Right can be chained, otherwise the resulting path makes no sense.

WTG defines *metrics* over paths. Those metrics help us to measure the relative authorization power of different paths. It only uses monotone metrics (for motivation, see [1]). Some examples of metrics are:

- $|C|_{\cdot} = |m_1||m_2| \cdots |m_n|$
- $|C|_{min} = min(|m_1|, |m_2|, \ldots, |m_n|)$
- $|C|_{+} = |m_1| + |m_2| + \ldots + |m_n|$
- $|C|_{max} = max(|m_1|, |m_2|, \ldots, |m_n|)$

where each m_i represents an arc/credential in the path C and $|m_i|$ refers to the weight of the arc/credential.

Depending on the domain of the possible values for Weight the metrics previously defined are increasing or decreasing functions. In particular, the metric $(|C|_{+}, \mathbb{N})$ is that one used by Ruan et al. in [4]. Note that Ruan define the weight 0 as the higher certainty level, so a higher value for $|C|_{+}$ represents a "weaker" path.

However, WTG takes the opposite approach by using $(|C|_{\cdot}, [0, 1])$ as the metric. In this case, a lower weight represents a "weaker" path ("weak" means that if the path C' is weaker than C, then it should be overridden by C).

The definition of metrics is the key for conflict resolution and allows to measure the priority of each authorization, or at least to compare them. Although there are a variety of orders that can be defined using metrics, others can not. One example is the lexicographic or dictionary order, denoted by $<_l$. In this case, the lexicographic order refers to the weight of the arcs in a path. When comparing two paths using the lexicographic order, we start by the closer arc from the owner of the resource and compare them until we find two arcs with different weights. At this stage, the path with the lower weight in this arc is overridden by the other path.

Given a metric, $|\cdot|$, over paths WTG defines $\mathcal{H}_{AB}$ and $\mathcal{L}_{AB}$ as the maximum and minimum weight, respectively, over all the authorization paths from A to B regarding the same Right. In other words, $\mathcal{H}_{AB}$ is the weight of the "better" path and $\mathcal{L}_{AB}$ is the weight of the "worst" path. That will help us to define the authorization policies. Note that all the indexes are meaningless without the definition of an associated policy. Authorization policies will be defined in section 4.

2.1 Re-visiting the Mean Index concept

WTG original work defined the average weight (Mean index), $\mathcal{M}_{AB}$, for softening the differences between $\mathcal{H}_{AB}$ and $\mathcal{L}_{AB}$. Although this index is very useful, it was relegated to a second level in that work.

Now, we further elaborate on this concept, providing an algorithmic definition. $\mathcal{M}_{AB}$ will be considered as a graph exploration using a *branch and bound* alike algorithm in which we incrementally calculate $\mathcal{M}_{AX}$ for each node X that is in a path from A to B.

We initializate $\mathcal{M}_{AX} = 0$ for all $X \neq A$ and $\mathcal{M}_{AA} = 1$, and associate those values to the corresponding nodes. In order to calculate $\mathcal{M}_{AB}$, it is necessary to inspect in the first step the principals connected from A with a single arc (*branch* phase), and add the weight of the corresponding arc to the weight of the node.

Then, the negatives nodes are marked as "non useful" because they can not further delegate, thus can not be part of any delegation path (*bound* phase). The process is repeated until B is reached. The result is that all non-useful nodes are marked. When reasoning about two principals A and B, the non useful nodes are omitted and an effective graph containing only the useful nodes is obtained. The resulting graph is easier to inspect, both visually and arithmetically. Each time a new arc (credential) is added to the system, we have to update the values calculated previously. As a consequence, valid nodes may be turned into negative ones (see Figure 2 for example). A positive delegation arc may imply a positive authorization arc, i.e. nodes which receive a delegation arc may issue an authorization arc pointing to themselves. Then, a negative authorization arc is in conflict with a positive delegation.

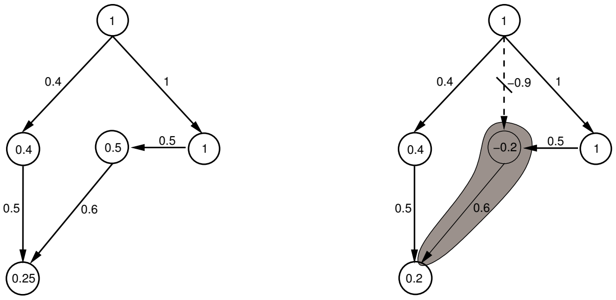


Fig. 2. Update of Trust Graph

3 Improving authorization presentation through graphics

In this section we present important improvements to WTG that are of great value to provide a general solution. That is, these improvements provide alternatives to organizations where traditional authorization systems do not fit well.

We also propose here a graphical representation of the authorization strength, defined as the level of trust that the issuer of the statement gives to the subject. This graphical representation is specially useful for those organizations that combine human and computer decisions.

One of the problems that we faced in the initial WTG work is that the Mean Index was affected negatively by low paths. Hence, adding a new path with a very low index leaded to a very low Mean Index. The main reason is that the weights of the credentials were not always normalized.

There is a basic relation among the idexes defined in WTG,

$$-1 \leq \mathcal{L}_{AB} \leq \mathcal{M}_{AB} \leq \mathcal{H}_{AB} \leq 1$$

$\mathcal{M}_{AB}$ is a index that offers us an average information about the authorization; however, the computation involved is very hard. On the other hand, $\mathcal{H}_{AB}$ and $\mathcal{L}_{AB}$ are less helpful but the computation involved is very simple. Thus, we have to combine them to reach a balanced solution.

We now present some methods to represent graphically those indexes. Given two principals A and B, we can represent $\mathcal{L}_{AB}$ and $\mathcal{H}_{AB}$ in the box $[-1,1] \times [-1,1]$ using the point $(\mathcal{H}_{AB}, \mathcal{L}_{AB})$. Moreover, as $\mathcal{L}_{AB} \leq \mathcal{H}_{AB}$, the point should be in the triangle of vertexes $(-1,-1),(1,-1),(1,1)$, as shown in Figure 3.a.

Then, the Mean can be represented as the point $(\mathcal{M}_{AB}, \mathcal{M}_{AB})$ in the region $\mathcal{H}_{AB} \geq X$,$Y \geq \mathcal{L}_{AB}$. Once we know how to represent the Mean in the axis, we may think of making authorization decisions based only in this index. However, the Mean is a more complex index regarding calculation, and there are cases in which this means that is not a good representative of the set of path weights. We could do it only when the Mean is a "good" representative of the whole set of path weights.

From the theory of statistics, we know that it is necessary to take a look at the standard deviation, what informs us how tightly all the weights are clustered around the Mean. Then, a low deviation informs us that the Mean is a "good" mean, while a high deviation represents that there are too many contradictory statements, so in this case the mean can not be considered as a measure of the trust strength between A and B and we should define a different mechanism.

In order to avoid computing the deviation, we could represent every path weight with the point (w,w), where w is the weight of the path. In figure 3.a we represent the weights as grey filled points and the mean as a bigger dark point. The representation of these points allow us to know the density of the path weights around the mean and the indexes $\mathcal{L}_{AB}$ and $\mathcal{H}_{AB}$ together in the same axes diagram.

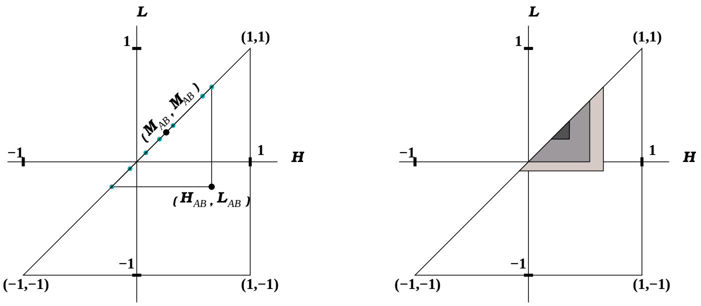
Although figure 3.a gives enough information regarding how good is the Mean Index, we can simplify this chart in the following way. We define a function that provides, for a given percentage x, the smallest interval centered in the Mean Index which contains at least the x percentage of weights.

Definition 1. *Let A, B be two entities and*

$$r_x := min\{y \in \mathbb{R} : [\mathcal{M}_{AB} - y, \mathcal{M}_{AB} + y] \textit{ contains the } x\% \textit{ of weights}\}$$

*Then, we define the **x-percentage interval** as* $[\mathcal{L}^x_{AB}, \mathcal{H}^x_{AB}]$*, where* $\mathcal{L}^x_{AB} := max\{\mathcal{L}_{AB}, \mathcal{M}_{AB} - r_x\}$ *and* $\mathcal{H}^x_{AB} := min\{\mathcal{H}_{AB}, \mathcal{M}_{AB} + r_x\}$

As a particular case, $\mathcal{H}^{100\%}_{AB} = \mathcal{H}_{AB}$ and $\mathcal{L}^{100\%}_{AB} = \mathcal{L}_{AB}$. We can now represent those intervals for the cases of x equal to 50%, 75% and 100%, respectively, using the triangle of vertexes $(\mathcal{L}^x_{AB}, \mathcal{L}^x_{AB})$, $(\mathcal{H}^x_{AB}, \mathcal{L}^x_{AB})$, $(\mathcal{H}^x_{AB}, \mathcal{H}^x_{AB})$. We fill them with different grey color scales, considering that the darker is the color, the lower is the percentage. We show a sample representation in figure 3.b. If the deviation is low, then the darker triangles will become small and, on the contrary, if the deviation is high then the darker triangles will become big.



(a) Authorization indexes (b) Percentage intervals

Fig. 3. Graphical representation

4 Associated Policies

The owner of the resource or attribute should be able to define different authorization requirements depending on the situation and on the resource or attribute. Note that there could be some critical resources that require a more restrictive authorization policy, but also other situations in which a non-critical object become critical and the associated security policy has to be changed. By separating the definition of the credential from the definition of the policies we obtain a more flexible authorization system. In this sense, we mean by *policies* the way of determining, according to the weights of each credential, if one entity is authorized by another one to perform an operation.

What we propose in the previous paragraph would allow us to change the policy according to the credentials defined, and the other way around. In this section we propose different authorization policies that can be used separately, or grouped.

One possible authorization policy would be to define an acceptance interval for the Mean Index. However, we start defining simpler policies which only depends on the simplest indexes $\mathcal{H}_{AB}$ and $\mathcal{L}_{AB}$ because they are more efficient in computation time and complexity.

The simplest policy we can think about is to grant the operation if there is any positive path between the owner of the resource and the subject. The major drawback of this policy is that positive and negative paths can be in conflict because we also use negative paths. The case *there is a positive path from A to B* translates in our formalism to $\mathcal{H}_{AB} > 0$. This is a must for authorization, but we need more restrictive conditions. A high restrictive approach would be to impose $\mathcal{L}_{AB} > 0$, that translates to *there are no negative paths from A to B.*

If we assign to each pair of entities A, B the bidimensional vector $(\mathcal{H}_{AB}, \mathcal{L}_{AB})$ as done in the previous section, we can represent a policy in an axes diagram as a subset of the triangle of vertexes $(-1,-1)$, $(1,-1)$ and $(1,1)$ that include all the acceptable tuples $(\mathcal{H}_{AB}, \mathcal{L}_{AB})$ for that particular operation to be granted.

Once explained the concept of authorization policy, we examine how a policy set P looks like. Suppose that we have a point inside P, i.e. let A, B be two entities such that $(\mathcal{H}_{AB}, \mathcal{L}_{AB}) \in P$. If A', B' are two entities with $\mathcal{H}_{A'B'} \geq \mathcal{H}_{AB}$ and $\mathcal{L}_{'A'B} \geq \mathcal{L}_{AB}$, this means that the lower path from A' to B' is greater than the lower path from A to B. It occurs similarly with the greater path. So we conclude that the paths from A' to B' are "better" than the ones from A to B. As a consequence, the point $(\mathcal{H}_{A'B'}, \mathcal{L}_{A'B'})$ should be in P too.

Definition 2. *We define a bound policy as a subset $\mathcal{P}$ of the triangle of vertexes $(-1,-1)$, $(1,-1)$ and $(1,1)$ with the following property: $(x,y) \in P$ implies that $(x', y') \in P$ for all $x' \geq x, y' \geq y$.*

We say that B is granted authorization from A if $(\mathcal{H}_{AB}, \mathcal{L}_{AB}) \in \mathcal{P}$.

The next authorization policy we include in our solution is based on Definition 2. In this case, we relax the condition of $(\mathcal{H}_{AB}, \mathcal{L}_{AB}) \in \mathcal{P}$ by allowing a few number of path weights to be out of P.

Definition 3. *Given a policy set P, we say B is granted access from A according to the x-Percent Policy for the set P, if the x percent of path weights are in P.*

In particular, if $(\mathcal{H}^x_{AB}, \mathcal{L}^x_{AB}) \in P$ then B is granted by A at x percent.

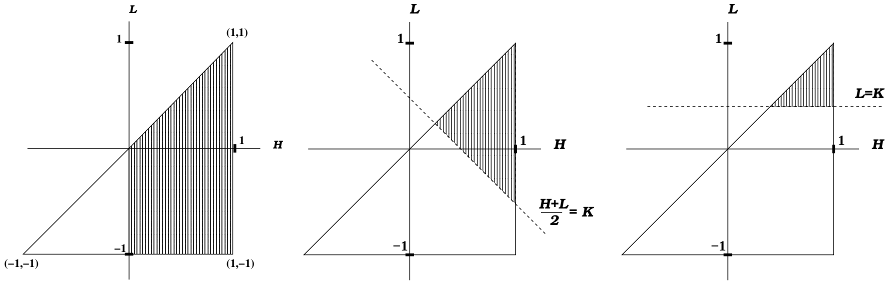
Based on Definition 2, we define two example policies in which we always force $\mathcal{H}_{AB} > 0$:

- **Absolute bound policy**: we choose a lower bound K for the lower path between two nodes. Only entities with $\mathcal{L}_{AB} > K$ will be authorized by this policy. We may define the policy set formally as

$$P := \{(\mathcal{H}, \mathcal{L}) : \mathcal{L} > K\}$$

- **Mean bound policy**: we choose a lower bound K for the mean of $\mathcal{H}_{AB}$ and $\mathcal{L}_{AB}$. In this policy, a positive path overrides a lower negative path. A particular case is when K is equal to zero. Formally,

$$P := \{(\mathcal{H}, \mathcal{L}) : \mathcal{H} + \mathcal{L} > 2K\}$$



(a) $\mathcal{H} > 0$ (b) Mean bound policy (c) Absolute bound policy

Fig. 4. Different types of policies

In figure 4, we represent some policies in an axis diagram.

In a Mean bound policy, when $K = 0$, it could happend that $\mathcal{H}_{AB} = -\mathcal{L}_{AB}$. Then, we use the lexicographic order to decide if we grant authorization or not by using the following procedure: we authorize entities if any of the paths with the highest weight is greater (using the dictionary order) than all the paths with the lowest weight. In other words, if there exits a path C with $\mathcal{H}_{AB} = |C|$ and $C >_L C'$ for all C' with $|C'| = \mathcal{L}_{AB}$

We use the lexicographic order to solve the case in which $\mathcal{H}_{AB} + \mathcal{L}_{AB} = 0$ but we can use the lexicographic order alone to decide about authorization. This is the reason why we define a **lexicographic policy** or hierarchical policy. We order all the paths from A to B according to the dictionary order and, if the maximal elements are all positives, then B is authorized. In case there are maximal negative paths, authorization is denied.

We define a third kind of authorization policy that we name **security level policy**. We define a real number K, as in the bound policy, but we use it to discard credentials with weight lower than K. Thus, we refine the delegation graph after computing the indexes, discarding all the credentials with a "low" weight. After this, we should apply some of the previous policies to decide about the authorization. We say that a credential path is *k-valid* if all the arcs in the path have weight greater or equal than k.

Suppose that we have three different security levels in the system represented in figure 5. For each security level we choose a different K. In this case, let $K_1 = 0.2$ be the lower security level, $K_2 = 0.3$ the second one and $K_3 = 0.5$ the highest level. In order to show how it works we use students authorizations:

- **Level 1** (0.2). In this security level, Student gets access to the resource, because Professor 2 issued a 0.2-valid statement.
- **Level 2** (0.3). In this security level, Student gets access to the resource, but Professor 2 authorization is not enough because the path goes across himself if not 0.3-valid. Student needs to use the statement issued by Professor 1.
- **Level 3** (0.5). In this security level, Student gets no access to the resource, because there is no any path higher than 0.5 that allows him to access the resource. The only one who can issue such a statement is the Dean.

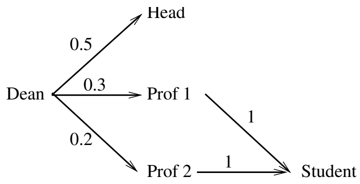


Fig. 5. Security level policy

4.1 Example of usability

Consider the following example in which we have five entities $\{A, B, C, D, E\}$ and seven credentials. We focus on the relations between A and E.

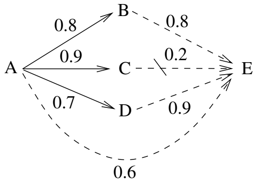


Fig. 6. Example of delegation/authorization graph

In this case, $\mathcal{H}_{AE} = 0.64$, $\mathcal{L}_{AE} = -0.18$ and $\mathcal{M}_{AE} = 0.4225 \sim 0.42$. Then we calculate the 75% interval for the Mean. We first calculate $r_{75\%}$

$$r_{75\%} := min\{y \in \mathbb{R} : [0.42 - y, 0.42 + y] \text{ contains the 75\% of weights}\}$$

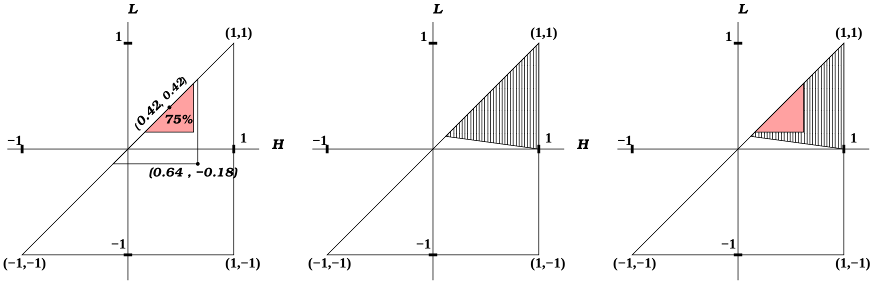
$r_{75\%} = 0.64 - 0.42 = 0.22$ so, $\mathcal{L}^{75\%}_{AE} := max\{-0.18, 0.20\} = 0.20$ and $\mathcal{H}^{75\%}_{AE} := min\{0.64, 0.64\} = 0.64$

The graphical representation of those indexes is in Figure 7. The grey triangle represent the 75% interval for the Mean. In this diagram we see that the area of the grey triangle is more or less half the area of the big triangle, what indicates that the weight of the paths are clustered around the mean. Hence, the Mean is a good representative for all the weights.

If we remove the negative path we get a very small triangle. Then, the smaller the triangle is, the better representative the Mean Index is. Looking at Figure 7.a we conclude that the Mean Index really represents the overall weights. A computer is also able to reach the same conclusion by inspecting $r_{75\%}$ (the smaller it is, the better representative the mean is).

The next step is to define different authorization policies for the previous example.

With this information we propose different authorization policies. In the example, if we propose a triangle P with their vertexes in the first quadrant (which means that we do not allow negative credentials for granting authorization) of the axes, the associated bound policy is that we will deny the authorization to



(a) Indexes and 75% interval

(b) Bound set

(c) 75% interval inside bound set

Fig. 7. Sample bound policy

E because $(0.64, -0.18)$ is not in the first quadrant. If we relax these conditions and opt for a Percent Policy of 75%, we may define several sets P in the first quadrant, what will lead us to grant the authorization (see Figure 7.b).

Therefore, we choose if the resource is so critical as to avoid negative credential or if we allow them but in case the Mean is good enough. In case we decide to use the lexicographic policy or hierarchical policy we will use the negative path to decide about authorization because it has the higher weight over all the first arcs in all paths. Choosing the lexicographic policy will lead to a denial of authorization from A to E.

Let's suppose that we apply a security level policy to discard non-relevant credentials (those which has a "small" weight). If we choose a bound lower than 0.2 then all credentials remain in the system. But if we choose 0.5 as a security level bound, we will avoid the negative path (see Figure 8). After that we could apply any of the previous policies to decide about authorization.

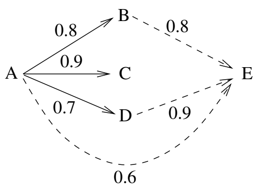


Fig. 8. Delegation graph after applying the security level policy

The application of the security level policy changes dramatically the situation. If we calculate the indexes for the new graph represented in Figure 8 we get $\mathcal{H}_{AE} = 0.64$, $\mathcal{L}_{AE} = 0.6$ and $\mathcal{M}_{AE} = 0.623\ldots \sim 0.62$. In this case, the Mean Index is very accurate (indeed $r_x \leq 0.02$ for all percentage intervals). We can compare the two situations according to r_x: in the original example, $r_{100\%} = 0.6$ and $r_{75\%} = 0.22$; now $r_{100\%} = 0.02$ and $r_{75\%} = 0.016\ldots$. Note that smaller intervals represent more clustered values.

5 Conclusions

We have presented in this work a major extension for WTG which comprises three main issues: definition of new indexes, graphical representation of indexes, and graphical representation of authorization policies.

Because our goal was to integrate this work into a user-oriented application, we provide a graphical representation of the indexes that is helpful for security administrators in defining authorization policies.

The graphical representation of indexes has opened the door to the definition of a graphical representation of policies. Having a graphical representation of authorization policies allows human decisions to be better included in the system. With logic-based frameworks, human interaction in the decision phase is impossible. However, in our framework, and because the representation of both policies and authorization (delegation) credentials is graphical, humans can interact based on this graphical information.

References

1. Isaac Agudo, Javier Lopez and Jose A. Montenegro. A representation model of trust relationships with delegation extension. In *3rd International Conference on Trust Management, iTrust 2005*, volume 3477 of *Lecture Notes in Computer Science*, pages 116 – 130. Springer, 2005.
2. Ninghui Li, Benjamin N. Grosof, and Joan Feigenbaum. Delegation logic: A logic-based approach to distributed authorization. *ACM Trans. Inf. Syst. Secur.*, 6(1):128–171, 2003.
3. Chun Ruan and Vijay Varadharajan. Resolving conflicts in authorization delegations. In *ACISP'02*, volume 2384 of *Lecture Notes in Computer Science*, pages 271 – 285. Springer, 2002.
4. Chun Ruan and Vijay Varadharajan. A weighted graph approach to authorization delegation and conflict resolution. In *ACISP'04*, volume 3108 of *Lecture Notes in Computer Science*. Springer, 2004.
5. Chun Ruan, Vijay Varadharajan, and Yan Zhang. Logic-based reasoning on delegatable authorizations. In *ISMIS '02: Proceedings of the 13th International Symposium on Foundations of Intelligent Systems*, volume 2366 of *Lecture Notes in Computer Science*, pages 185–193. Springer, 2002.
6. Chun Ruan, Vijay Varadharajan, and Yan Zhang. A logic model for temporal authorization delegation with negation. In *6th International Information Security Conference, ISC 2003*, volume 2851 of *Lecture Notes in Computer Science*, pages 310 – 324. Springer, 2003.