Delegating Privileges over Finite Resources: A Quota Based Delegation Approach*

Isaac Agudo, Carmen Fernandez-Gago, and Javier Lopez

Department of Computer Science, University of Malaga, Malaga 29071 Spain
Email:{isaac,mcgago,jlm}@lcc.uma.es

Abstract. When delegation in real world scenarios is considered, the delegator (the entity that posses the privileges) usually passes the privileges on to the delegatee (the entity that receives the privileges) in such a way that the former looses these privileges while the delegation is effective. If we think of a physical key that opens a door, the privilege being delegated by the owner of the key is opening the door. Once the owner of the key delegates this privilege to another entity, by handing over the key, he is not able to open the door any longer. This is due to the fact that the key is not copied and handed over but handed over to the delegatee.

When delegation takes place in the electronic world, the delegator usually retains also the privileges. Thus, both users have them simultaneously. This situation, which in most cases is not a problem, may be undesirable when dealing with certain kind of resources.

In particular, if we think of finite resources, those in which the number of users accessing simultaneously is finite, we can not allow that a user delegating his access privilege is also granted access when the delegation if effective.

In this paper we propose an approach where each user is delegated an access quota for a resource. If further delegating of the delegated quota occurs, this is subtracted from his quota. That is, when delegating, part of the quota remains with the delegator and another part goes to the delegatee. This allows a more fairly access to the resource. Moreover, we show that this approach can also be applied to any kind of resources by defining appropriate authorization policies.

1 Introduction

When delegation in real world scenarios is considered, the delegator (the entity owning the privileges) usually passes the privileges on to the delegatee (the entity that receives the privileges) in such a way that the former looses these privileges while the delegation is effective. If we think of a contact-less ID card used for

* This work has been partially funded by the European Commission through the research project SPIKE (FP7-ICT-2007-1-217098), and the Spanish Ministry of Science and Education through the research project ARES (CONSOLIDER CSD2007-00004)

opening a door, the privilege being delegated by the owner of the card is opening the door. Contact-less ID cards are meant to be tamper resistant and hence non feasible to be copied. Then, once the owner of the card delegates this privilege to another entity, by handing over the key, he is not able to open the door any longer. This is due to the fact that the key is not copied and handed over but just handed over to the delegatee.

When delegation takes place in the electronic world, the delegator usually retains the privileges. Thus, both users hold the privilege simultaneously . This situation, which in most cases is not a problem, may be undesirable when dealing with certain kind of resources. Current solutions for privilege management with support for delegation (see PolicyMaker [3], KeyNote [2], SPKI [4], PMI [7]) do not address this situation.

In particular, if we think of finite resources, those in which the number of users accessing simultaneously is finite, we can not allow that a user delegating his access privilege is also granted access when the delegation is effective.

In our approach each user is delegated an access quota for a resource and when further delegation occurs the delegated quota is subtracted from his given quota. Thus, when delegating, part of the quota remains with the delegator another part goes to the delegatee.

In this paper we propose a model useful for delegating rights or authorization in order to use a specific resource that can be split in several parts. Granting a part of a resource can be seen as granting a percentage of it. Thus, when issuing credentials together with the resource we should specify the percentage, or quota, of it that is being delegated. Our model uses Markov's chains, widely used as a statistics model [8, 13].

This quota percentage can be compared with a trust value in the sense that the higher it is, the more power the holder of the credential has. In fact, the approach presented in this work is similar, and related, to some existing approaches for trust management. One of these methods is PageRank [12] that represents a way of ranking the best search results based on a page's reputation. Flow models such as Advogato's reputation system [9] or Appleseed [14, 15] use of transitivity. In these type of systems the reputation of a participant increases as a function of incoming flow and decreases as a function of ongoing flow.

The paper is organized as follows. In Section 3 we describe two scenarios where our model could be applied. The model is presented in Section 4 and its complexity in Section 4.2. Section 6 outlines the future work and concludes the paper.

2 Related Work

As we mentioned in the introduction, current delegation schemes used in authorization systems do not fully take into consideration the scenario where the delegated privilege can not or must no be shared between the delegator and the delegatee. Some works have defined the privilege transfer scenario, but nothing has been mentioned about the quota based approach.

PolicyMaker [3] is a general and powerful solution that allows the use of any programming language to encode the nature of the authority being granted as well as the entities to which it is being granted. KeyNote [2] is a derivation of PolicyMaker, and has been supported by IETF.

Blaze, Feigenbaum and Lacy introduced in [3] the notion of Trust Management. In that original work they proposed the PolicyMaker scheme as a solution for trust management purposes. It addresses the authorization problem directly, without considering two different phases (one for authentication and another for access control).

Keynote [2] uses a specific assertion language that is flexible enough to handle the security policies of different applications. Assertions delegate the authorization to perform operations to other principals. KeyNote considers two types of assertions called policies and credentials.

In both approaches, once users obtain privileges they can delegate it to any other user while also being able to use it at the same time. This is why none of these solutions can be used in our scenarios.

SKPI [4] was proposed by the IETF working group. The SPKI certificate contains at least an Issuer and a Subject, and it can contain validity conditions, authorization and delegation information. The delegation information is used to specify the maximal length of a delegation path. When set to 0 delegation is not allowed. This approach does not deal neither with finite resources nor with privileges that can not be shared.

Privilege Management Infrastructure (PMI) is defined in X.509 ITU-T Recommendation [7] as the framework for the extended use of attribute certificates. The Recommendation establishes four PMI models, one of them is the Delegation model. Initially, the Source of Authority (SOA) assigns or delegates the privilege to Attribute Authorities (AA). These can delegate the privileges to other AAs or to end entities (EE). AAs and EEs can use their delegated privileges and present them to the Privilege Verifier (PV) that verifies the certification path to determine the validity of the privileges. The mechanism used to contain the delegation statements is the attribute certificate. The extensions field is used by the authorities to include the delegation policy.

Even though PMIs do not directly deal neither with finite resources nor with privileges that can not be shared, we can take advantage of the extension mechanism in order to be able to manage them.

3 Applicability Issues or Scenarios

The type of scenarios where the quota model scheme can be applied are those where an entity has all the access to a resource and it could hand over shares of the access to this resource to some other entities. In this section we will outline two cases.

3.1 Residential Network Scenario

Let us assume a residential environment where the residents could use limited resources such as file space in a shared hard disk or the internet connection bandwidth. Those resources are shared among users in the residential environment according to some parameters such as how much they contribute to the residential network (e.g. money or hardware) or another less objective parameters such as friendship or trust relationships.

How are new users introduced to the residential network? How are resources assigned to them? All these questions could be answered according to the initial network configuration.

We assume a simple scenario where there are only two users in the residential network and a new user wants to access its services, in particular, the shared space and the bandwidth. In case both initial users had the same relevance in the network, both of them will own a half of the space and a half of the bandwidth. Then, one of them, or both, will hand over some of his space and bandwidth to the new user.

The easiest way for the new user to use the resources is to make an arrangement with one of the initial users in order to share his part of the network. This arrangement may involve some payment from the new user. In case the initial user shares half of his resources with the new user, the new configuration will include 3 users with a share of 50%, 25% and 25%. This process can be repeated in order to include new users in the system.

The situation is even more interesting when a new user knows two current users and obtains from them a part of their shares. In this case the share of the new user will be the sum of the parts handed to him. Thus, a new user can accumulate more quota by dealing with existing users.

The structure of the network could then be encoded as a weighted trust graph. This makes easier to define a central authorization module that takes as input the graph of the network and controls access to the resources in the network.

3.2 Grid Organization

Another scenario where the quota delegation policy is applicable is the following. Let us assume a grid composed of different organizations sharing multimedia resources. Each organization has a participating quota that determines the influence in the authorization process in order to use the resources.

When making authorization decisions this hierarchy, and the participation indexes of quota, have to be taken into account. There are several ways of doing this.

Each entity may issue different certificates, and those will be weighted according to their participation index.

Each entity participating in the grid has a share of quota. This share of quota could be handed over to external users. Usually, each organization would keep some of this quota for its use and will ‘sell’ the surplus of resources. This process could take place in a cascade effect. That is, if an external organization has bought a participation in a particular resource, this organization could also sell a portion to another external organization.

As consecutive sales advance, the quota that the new organizations obtain decreases. That means that more external organizations to the grid will have less influence on managing rights to access the resources of the grid.

There should also be a policy of access for each resource established by the grid and a common agreement where the minimum quota needed for accessing such a resource is reflected. Depending on the nature of the resource, this quota will be higher or lower. For instance, if the resource is a cluster, a request for a minute slot of use requires a lower quota than for using it for longer. Also, if there are two or more users competing for the same resource the quota can be used in order to prioritize the access to the resource.

4 A Quota Approach for Delegation

In Section 3 we have presented a scenario where the entities participating in a grid delegate rights management to another organizations, from inside and outside the grid. In this section we will introduce a mathematical model that formalizes the situation of the scenarios described above. We will call this model the *Quota Delegation Model*.

Since organizations in lower levels have been delegated less quota as we descend one level in the chain, we are interested in the exact quota that any organization retains. The quota is a real number in the interval $[0, 1]$, representing a percentage of the total quota where 1 corresponds to 100%. A user expresses the quota delegated to other entities relatively to the actual quota they obtain. The actual quota that a user delegates to another one is computed by multiplying the relative quota (encoded in the delegation quota credential) and the actual quota of the delegator. The absolute quota of a user is computed by summing up all the absolute quotas delegated by all the users of the system. In this section we will provide with an iterative mechanism for computing the absolute quota of each user.

In order to compute the quota of a certain organization we will use the product of all the quota of the links in the chain from the root node to the node we want to compute, minus the quota that this entity delegates. In fact, for each entity we can distinguish the delegated quota, which is the quota that reaches this entity trough delegation paths, and the actual quota of this entity that is computed by subtracting the quota delegated to other entities from the delegated quota of this entity.

If there are several chains for delegating quota to organizations, the final quota will be the addition of all the quotas obtained from all the different chains.

Loops are not allowed in our quota delegation model and have to be solved outside of it. For instance, if entity X has delegated some of its quota to entity Y and later, Y wants to delegate some of its quota to X, then there are two possibilities:

- If the absolute quota that Y wants to delegate to X is greater than the absolute quota X delegates to Y, then the quota delegation credential from X to Y should be removed from the system and a new quota delegation credential from Y to X should be added, where the delegated quota corresponds to the difference between the absolute quota that Y wants to delegate to X and the absolute quota delegated from X to Y expressed relatively to the quota of Y.
- If the absolute quota that Y wants to delegate to X is lower than the absolute quota X delegates to Y, then the current quota delegation credential from X to Y should be removed from the system and a new quota delegation credential from X to Y should be added. Then the delegated quota corresponds to the difference between the absolute quota delegated from X to Y and the absolute quota that Y wants to delegate to X expressed relatively to the quota of X.

In both cases the resulting quota delegation graph has no loops.

We have initially used attribute certificate credentials [7, 5] to implement our Quota Delegation Model. The privilege delegated is encoded as an attribute and the value of the attribute is set to the quota. All the credentials have to be passed to a central server which checks that the quota assignments that each user does for each privilege is fair, i.e. the quotas of each credential issued by the same user and regarding the same privilege sum less than 1, which represents 100% of the quota. This central server stores all the delegated quotas in a matrix form.

If we establish an equivalence between nodes or organizations and states, and between the quota that an entity X hands over to Z and the statistical concept of transition from one X to Z, our particular problem could be modelled as a discrete Markov's chain where the number of phases or stages corresponds to the length of the chain that we consider.

4.1 Computational Model

The initial organization or entity that first holds all the quota is called the *initiator* and it will be the initial state of the Markov's chain. The values of the quotas could be placed in a matrix such as the addition of the elements in each row is lower or equal than 1 ('almost' a stochastic matrix). The reason is that the addition of all quota could never be greater than 1 but it can be less, i.e., there could be states without any assigned quota. Therefore, in order to make this matrix a stochastic matrix we should add an additional state, namely, the state of the non-assigned quota. This new state will mean a new column and row for the matrix of the states where the values in the columns are calculated in such a way that the addition of the values in the row is 1. If we call this matrix A, the associated stochastic matrix A^* is as follows:

$$\left(\begin{array}{c|c} & 1-\sum_{j=1}^{n} a_{1,j} \\ A & \vdots \\ & 1-\sum_{j=1}^{n} a_{n,j} \\ \hline 0 \cdots 0 & 1 \end{array}\right)$$

The state of 'non-assigned' quota is a non-transition state as once it has been reached no other state can be reached afterwards.

This matrix can be expressed in its canonical form as

$$\left(\begin{array}{c|c} A & R \\ \hline 0 & I \end{array}\right)$$

where A is the matrix of quota delegation that contains all the transition states; R is the matrix that contains the non-assigned quota by entities and I is the identity matrix of length 1. As there are no loops in the quota delegation graph, the matrix A is upper triangular or at least we can make it upper triangular by reordering the nodes using a topological sort algorithm over the quota delegation graph. Therefore, the matrix of the chain, N, is calculated as follows:

$$N = (I - A)^{-1}$$

Next we will show that the element n_{ij} of N is the share of quota that entity i hands over to entity j.

An element $a_{ij}^{(k)}$ of matrix A^k represents the percentage of quota that node i hands over node j indirectly, if we only consider chains of length k. Also, $A^n = 0$ if n is greater or equal than the size of the matrix, as this is an upper triangular matrix. Therefore, we can define matrix $\widehat{A}$ as

$$\widehat{A} = \sum_{i=1}^{\infty} A^i = \sum_{i=1}^{n-1} A^i$$

The elements of $\widehat{A}$ can be calculated in the following way:

$$\hat{a}_{ij} = \sum_{s=1}^{n-1} a_{ij}^{(s)}$$

$$\widehat{A} = (\hat{a}_{ij})$$

The elements $\hat{a}_{ij}$ of $\widehat{A}$ are the addition of all the quota that node i hands over to node j for chains of any length.

Next we will show that $N = I + \widehat{A}$, i.e., $I + \widehat{A}$ is the inverse of $I - A$. In order to do this, we will show that the matrix is invertible from the right-hand side (it is analogous from the left-hand side).

$$
\begin{aligned}
(I - A)(I + \widehat{A}) &= (I - A)(\sum_{s=0}^{n-1} A^s) \\
&= \sum_{s=0}^{n-1} A^s - \sum_{s=1}^{n} A^s \\
&= I + \sum_{s=1}^{n-1} A^s - \sum_{s=1}^{n-1} A^s - A^n \\
&= I
\end{aligned}
$$

This gives us two ways of obtaining the percentage of quota handed over by entity i to entity j. Either by calculating the element ij of matrix N or by calculating the element (i, j) of matrix $\widehat{A}$.

In both cases, we can use the first row of those columns to form the vector of quota delegation from the initiator.

Definition 1 (quota delegation vector). *The quota delegation vector is the vector v consisting of all the quota delegation values from the initiator to the rest of the entities. The first element is set to* 1*.*

In order to obtain the actual quota that remains in each entity, we have to subtract the quota it delegates from the quota it has been delegated. This can be easily done by multiplying the corresponding element of the quota delegation vector, v_i, by the element r_i of the column vector R.

By multiplying the column vector R by v, element by element, we obtain the quota distribution over the entities. The sum of all the elements of this vector is 1.

4.2 Efficiency Analysis

The quota delegation model presents a feature that, in some cases, could be an advantage and, in some others, a disadvantage depending on the nature of the system. This feature is that if all the available quota has been assigned and we would be interested in assigning quota to a new entity, this should be taken away from the previously assigned quota.

Taking quota away could affect the entities which already had them assigned and therefore, the quota should be re-distributed again. This re-distribution will affect more to entities closer to the entity that it is the root of the quota. Thus, if we establish an order where the entity origin of the quota is of order 1 and the other entities' order follows from the order how the quota is assigned, as higher the order is, less impact will have the new distribution on this entity.

Taking into account all the above considerations we can make some remarks concerning the complexity of the calculation of matrix $\widehat{A}$. First, the consecutive powers of matrix A have mainly zeros as their elements. Thus, for example, while

A is an upper triangular matrix, the elements of the diagonal a^2_{ii+1} of the matrix $A^2 = (a^2_{i\,j})$ are all 0. Also, if the size of the matrix A is n then A^{n-1} has only one element which is not 0. This element is a^n_{1n}. Thus, in order to calculate the elements of the diagonal $\hat{a}_{i,i+k}$ of matrix $\widehat{A}$, we only need elements of the powers of A which are less or equal than k.

Next, we will see how many elemental operations we need in order to calculate those elements.

Let d_k be the diagonal k for $k \in \{1, \ldots, n-1\}$ of any matrix $A = (A_{ij})$. This diagonal has $n - k$ elements which are

$$d_k(A) = \{a_{j\,j+k}\}_{j=1}^{n-k}$$

The number of operations needed in order to calculate $d_m(A^p)$ is $(m - p + 1)(n-m)$ multiplications and $(m-p)(n-m)$ additions (the detailed explanation is beyond the scope of this paper).

Therefore, the number of multiplications needed in order to calculate $\widehat{A}$ and the number of additions needed in order to calculate the powers of A are respectively

$$\frac{(n+1)n(n-1)(n-2)}{24} \text{ and } \frac{n(n-1)(n-2)(n-3)}{24} \tag{1}$$

In order to calculate $d_m(\widehat{A})$ we have to add the non-null diagonals $d_m(A^p)$, i.e., $m - 1$ diagonals. Since these diagonals have $n - m$ elements, the number of additions for adding the consecutive powers of A is

$$\frac{n(n-1)(n-2)}{6} \tag{2}$$

From those results we can calculate the total number of additions that are required for calculating $\widehat{A}$ and the total number of operations. Those two numbers are respectively

$$\frac{n(n^3 - 2n^2 - n + 2)}{24} \text{ and } \frac{n(n^3 - 2n^2 - n + 2)}{12}$$

As a remark we can say that this algorithm for calculating the distribution of quota for n entities is of the order $\mathcal{O}(n^4)$.

If we used the matrix N instead, we can determine the complexity of the method by analyzing the complexity of the calculation of the inverse of $I - A$.

This matrix is invertible and its inverse is $I+\widehat{A}$. It is also upper triangular. In this case, we could solve as n systems of simultaneous equations $(I - A)x^i = b_i$, where b_i are the consecutive columns of matrix I in order to calculate the inverse. This inverse will be the matrix, in columns, $(I - A)^{-1} = (x^{(1)}\| \ldots \|x^{(n)})$. We can deduce that the inverse matrix is also upper triangular by observing the sub-system of the $n - i$ equations of each system,

$$\begin{pmatrix} 0 \; 1 \cdots & -a_{i+1\,n} \\ \vdots \; \vdots & \vdots \\ 0 \; 0 \cdots & 1 \end{pmatrix} \begin{pmatrix} x_{i+1}^{(i)} \\ \vdots \\ x_n^{(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

therefore, $x_j^{(i)} = 0$ for $j > i$. Thus, resolving the remaining i equations of the system they can be simplified as follows

$$\begin{pmatrix} 1 & -a_{12} & \cdots & -a_{1i} \\ 0 & 1 & \cdots & -a_{i+1\,i} \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} x_1^{(i)} \\ x_2^{(i)} \\ \vdots \\ x_i^{(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

This triangular system of i equations can be resolved by performing i^2 operations by using the substitution method. Thus, the final number of operations needed for calculating the inverse of $I - A$ will be

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$$

This means that by using the matrix N we can reduce the complexity to $\mathcal{O}(n^3)$.

5 Quota Based Delegation and Authorization Policies

Our quota based delegation model is mainly focused on facilitating delegation of access to resources that can be measured and consequently divided among users according to the specified quota percentage. Examples of such a kind of resources are the Internet connection bandwidth (see Section 3.2), file storage space, CPU load in cluster environments, etc. In those cases there is not need for a specific authorization policy to be used against the certificates, as they encode both things. The rights are already included in the credential, therefore we do not need to contrast it with an authorization policy. However, not all the resources are easily split and furthermore, sometimes it is undesirable to include the resource in the credential. In those cases, we use a role or group membership attribute and split this attribute among the users in the system. The grid scenario is a clear example of this situation. In those cases, we do need an authorization policy, therefore actual privileges or rights can be derived from the quota membership to a particular attribute. In Figure 1 we illustrate how those two scenarios are characterized according to where more effort is needed, either in the definition of credentials or in the definition of the authorization policies.

Even though we have made a distinction here between these two examples, there might be cases where it is not that easy to make it. For the definition of

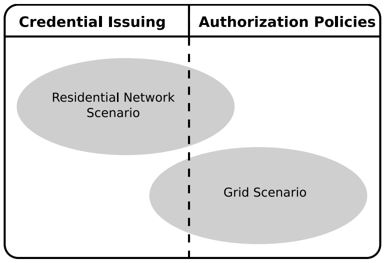


Fig. 1. Characterization of the two scenarios

the authorization policy we take as an input the quota or the percentage of the attribute that the user holds, and the higher it is, the more privileges it will be delegated.

The authorization problem can be also tackled by using negative statements, such as 'this user will never access this resource'. As each user is delegated a quota of the resource, an authorization credential can be given a weight associated to the quota of the issuer. Then, positive authorization credentials, i.e., those granting some privileges but not delegating them, can be counted as positive votes for authorization decisions and negative ones as negative votes. Those votes are proportional to the quota of the issuers of the credentials, in such a way that at the end, we can sum up all the positive votes and subtract from them all the negative ones. If the result is positive the authorization request will be granted. However, this is just a simple authorization policy. More complex solutions can be defined, such as requesting a lower bound for the delegated quota.

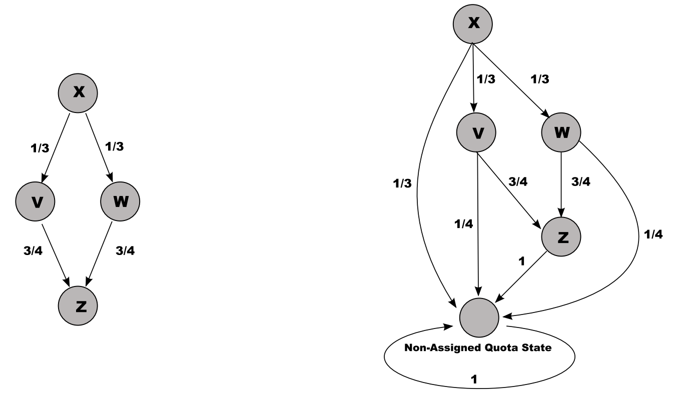
5.1 Example

Let us assume a grid of four organizations X, Z, V and W. Let us also assume that the X is the user who initially possesses 100% of the quota of a given resource, i.e. X is the owner of the resource and wants to contribute a share of it to the grid. X hands over a third of the quota to V and another third to W. V and W also delegate $3/4$ of their quota to Z. We are interested in calculating the exact quota that each entity in the grid retains after all the quota delegations are effective.

In order for the matrix A to be upper triangular, we can establish the following order of the nodes: $X \rightarrow 1$, $V \rightarrow 2$, $W \rightarrow 3$ and $Z \rightarrow 4$.

Figure 2 shows the distribution of the quota. The Figure on the left-hand side describes how the quota originally is distributed among peers and, on the one on the right-hand side the distribution of quota is represented as a States Transition Diagram (STD) of the associated Markov's chain which includes the 'non-assigned' quota states.

The matrix representing the quota assigned is as follows



(a) Original Assignment of quota

(b) STD of the associated Markov's chain

Fig. 2. From quota assignment graph to Markov's chain STD

$$A = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{3}{4} \\ 0 & 0 & 0 & \frac{3}{4} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

From this matrix we can obtain the stochastic matrix by including the non-assigned quota in the last row and column.

$$A^* = \left(\begin{array}{cccc|c} 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{3}{4} & \frac{1}{4} \\ 0 & 0 & 0 & \frac{3}{4} & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 \end{array}\right)$$

The fifth row and column correspond to the state of non-assigned quota.

In order to calculate the quota that X hands over Z we should calculate, in Markov's processes terminology, the probability of going from state X to Z. We will use matrix N for doing it.

$$N = (I - A)^{-1} = \begin{pmatrix} 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{2} \\ 0 & 1 & 0 & \frac{3}{4} \\ 0 & 0 & 1 & \frac{3}{4} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In order to obtain the actual quota belonging to each user, we have to subtract the quota that has already been delegated. We can do this by multiplying the first row of this matrix by the column of the non-assigned quota, element by element.

$$(1, \frac{1}{3}, \frac{1}{3}, \frac{1}{2}) \times (\frac{1}{3}, \frac{1}{4}, \frac{1}{4}, 1) = (\frac{1}{3}, \frac{1}{12}, \frac{1}{12}, \frac{1}{2})$$

Therefore, the shares of assigned quota of all the participants are the corresponding elements of this vector. Note that the sum of the elements of this vector is one, therefore the quota property holds.

If we implement an authorization policy such as the simple one defined in the previous section in such a way that Z and W decide that access to a third party has to be granted, it does not matter what X and Y state, as the votes of Z and W count more than 50%. Thus, the decision will be to grant access.

6 Conclusions and Future Work

In this work we have presented a delegation mechanism for finite resources or environments where there could be a conflict of interest among the entities involved in the decision making process.

By specifying quotas in credentials we allow for fair delegation among the participants, as we can control that a resource is never overflowed by a massive access from participants. When using the quota delegation mechanism in conjunction with attributes, instead of with proper resources, fairness is not the main objective but solving disputes between participants about which other users are granted some privileges.

Currently we are exploring how this quota can be included in standard X.509 attribute certificates in a more coherent manner. We are using ideas from [10, 1]. For doing this, we should first implement a mechanism such that a user is not allowed to issue credentials for more than 100% of his quota.

The trivial solution consists of storing all the credentials in a central server that performs consistence checking over delegated quota. However, we believe that distributed solutions, or at least semi-distributed solutions, can be achieved.

We are also focusing our efforts on exploring the field of encrypted databases [11, 6] in order to try to implement a quota service database where each user stores all the quota delegation credentials in an encrypted manner. Thus, this quota service database, that may also be distributed, could answer consistence queries, i.e. the delegated quota does not exceed the 100% of the own quota, without revelling information neither about the delegated entities nor the actual quota being delegated to them, to the privilege verifier.

The privilege verifier would therefore be able to compute at least a lower bound for the delegated quota, based on a subset of the actual delegation paths. In order to determine a lower bound of the delegated quota to a user, it should be feasible to verify with the Quota Service Database that the specified quotas in each of the credentials paths are part of fair assignment of quota. That is, the summation of all the quotas of credentials of a given issuer for the same privilege should never exceed 100%.

Furthermore, the Quota Service Database would allow us to use both, push and pull mechanisms for authorization. By using the quota service database, the privilege verifier can check that the credentials a user has sent to in order to attest his quota are all the existing one.

References

1. Isaac Agudo, Javier Lopez, and José A. Montenegro. A representation model of trust relationships with delegation extensions. In Peter Herrmann, Valérie Issarny, and Simon Shiu, editors, *iTrust*, volume 3477 of *Lecture Notes in Computer Science*, pages 116–130. Springer, 2005.
2. M. Blaze, J. Feigenbaum, and A. D. Keromytis. KeyNote: Trust Management for Public-Key Infrastructures (position paper). *Lecture Notes in Computer Science*, 1550:59–63, 1999.
3. M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized Trust Management. In *IEEE Symposium on Security and Privacy*, 1996.
4. C. Ellison. *SPKI Certificate Theory, Request for Comments 2693*. IETF SPKI Working Group, September 1999.
5. S. Farrell and R. Housley. *An Internet Attribute Certificate Profile for Authorization*. IETF PKIX Working Group, April 2002. Request for Comments 3281.
6. Stuart Haber, William Horne, Tomas Sander, and Danfeng Yao. Privacy-preserving verification of aggregate queries on outsourced databases. Technical report, Trusted Systems Laboratory, HP Laboratories Palo Alto, 2007.
7. ITU-T Recommendation X.509. *ITU-T X.509, ISI/IEC 9594-8, Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks*, August 2005.
8. J. G. Kemeny. *Finite Markov Chains*. New York, 1976.
9. R. Leiven. *Attack Resistant Trust Metrics*. PhD thesis, University of California at Berkeley, 2003.
10. J. Montenegro and F.Moya. A practical approach of X.509 attribute certificate framework as support to obtain privilege delegation. In *1 st European PKI Workshop: Research and Applications*, pages 160–172. Lecture Notes in Computer Science (LNCS) 3093, Springer-Verlag, 2004.
11. Maithili Narasimha and Gene Tsudik. Authentication of outsourced databases using signature aggregation and chaining. In *DASFAA*, pages 420–436, 2006.
12. L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.
13. A. Papoullis. *Probability, Random Variables and Stochastic Processes*, chapter Brownian Movement and Markoff Processes, pages 515–553. New York, 1984.
14. C. N. Ziegler and G. Lausen. Spreading Activation Models for Trust Propagation. In *IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE'04)*, Taipei, March 2004.
15. C.-N. Ziegler and G. Lausen. Propagation Models for Trust and Distrust in Social Networks. *Information Systems Frontiers*, 7(4-5):337–358, December 2005.