Modelling Trust in Collaborative Environments

Nicola Mezzetti
Dept. of Computer Science
University of Bologna
email:nicola.mezzetti@CS.UniBO.IT

Abstract

Trust a crucial concept in order to address scalability in managing most of the security tasks (e.g., authentication, authorization, access control) that are needed to prevent unauthorized access to shared resources in the context of a wide area network.

In this paper we take in consideration the concept of trust by defining a general theoretical model to describe basic trust relationships in heterogeneus environments composed of several Administrative Domains (ADs), i.e. autonomous domains for both security and administrative issues. We propose a model that also allows to specify delegation by making use of a certain degree of redundancy; this model can help detecting failures or maliciousness of entities inside the system.

As a result, it is introduced the development of a TMS to be deployed into a middleware architecture. By making use of the principles treated in this document, that TMS would provide a common semantic for the representation of trust relationships in virtual organizations (VO); this would allow each AD the VO embodies to partecipy in resource sharing in a scalable and secure way, preserving its local security infrastructure.

1 Introduction

In a global networking context, distributed environments for group collaboration are no longer located within corporation boundaries; many are the applications in which several, possibly mutually distrustful, corporations need to share services and resources to perform a common task or to jointly provide a particular service. Informally, we say that these corporations create a Virtual Organization (VO)[1]. A VO can be thought as a dynamic environment consisting of a set of *"entities"* (e.g., users, resources, ADs) that do not necessarily trust each other.

Managing trust in collaborative environments which can be deployed in a wide area network is a real challange; a variety of *Trust Management Systems* (TMSs) [1, 2, 3, 4, 6, 5, 7, 8, 9] have been proposed; unfortunately, each one implements its own semantic of trust. We believe that, to achieve interoperability among different trust management architectures, it is necessary to introduce a common semantic of trust that every TMS should implement. The implementation of a common semantic would guarantee that trust relationships spanning

[1] In literature, both the terms Virtual Enterprises and Coalitions are used as synonyms of Virtual Organizations.

several AD, each one adopting its own local security solution, can take place with a high degree of security. In such environment, the mapping between the credential system can be performed without major difficulties.

In this paper we investigate trust and trust relationships. First, we give a definition of these concepts by describing the properties that trust relationships should satisfy. Next, we use these concepts to describe the trust relationships that model the behaviour of entities in collaborative environments; in order to allow a forlam description of VOs, we define what we mean by *trust zone*, i.e. an abstraction to model real world basic security domains.

In Section 2, we discuss the meaning of trust and present a formal model to express generic trust and trust relationships. Section 3 introduces the concept of trust zones and shows their use in modelling coalitions. Section 4 describes an idea for a middleware layer TMS that can be used for managing access control to shared resources in virtual enterprises; this discussion is meant to be the basis for the future work. In Section 5 we draw our conclusions.

2 On the Nature of Trust

When speaking of trust in distributed systems, the entities of interest are mostly individuals, resources and processes. From the point of view of trust, there is no difference between these entities so we can capture all of them under the abstraction of *principal.* A principal is an entity that can be involved in a trust relationship, thus it is an entity that is able to authenticate itself to other entities.

Before describing the trust relationships characterizing collaborative environments, we must understand what the word *trust* means and how trust relationships are conceived in distributed systems.

2.1 A Definition for Trust

The Oxford dictionary gives the following definition for the term "trust":

> *Trust*: "Firm belief in the reliability or truth or strength of an entity".

This definition gives a good idea about the issue behind trust. It is worthnothing that IT researchers prefer the term "competence" rather than "stength" when referring to computer science applications.

The concept of trust applies whithin a context; when we say "Alice trusts Bob" we mean that exists a context in which Alice believes in the reliability and competence of Bob.

In information technologytrust can be defined as follows [1]:

> *Trust*: "The firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context".

In the above definition it is assumed that dependability covers reliability and timeliness. To understand the relationships between the various institutions in a community, it is also important to define *distrust.*

> *Distrust*: "The lack of firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context"[1].

In the relationship "Alice trusts Bob", Alice is named *trustor* and Bob is called the *trustee.* Here, trust is the key element that makes it possible for the trustee to obtain from the trustor information or some privileges when accessing a service, a resource or a set of them.

Now, we can formalize trust by giving three properties that are relevant for analyzing a real system with the intention to understand the conditions under which communication can take place in a distributed environment and trust relationships can be established.

2.2 The Formal Model

From now on, a trust relation will be formalized as a ternary relation $\mathcal{T}(\alpha, \beta, \phi)$ where α and β are the two principals and ϕ is a context. By this formalism, the assertion "Alice trusts Bob in the context of authentication" can be expressed by the notation $(Alice, Bob, authentication) \in \mathcal{T}$.

We define *"authentication"* to be the simplest context in which a principal can trust another one; this assumption relies on the fact that trust in authentication can be built on common security protocols that are widely used and for which a proof of correctness and completeness is given. All security protocols rely on modern cryptography that is founded on widely known efficiently unsolved problems (e.g., the decision Diffie-Hellmann problem).

Going on, we talk about trust in particular environments named *trust systems*; informally, a trust system can be defined as an environment in which trust relationships can be established between that environment's principals.

Let P be a set of principals that belong to a trust system. Be $\mathcal{T}$ a trust relation that associates a 3-tuple representing two principals, belonging to the set P, and the context in which trust between them applies.

$$\mathcal{T} \subseteq P \times P \times \Phi$$

A trust relationship $\mathcal{T}$ satisfies the *reflexive* property, also called *implicit trust*; this can be described as follows:

> "Alice trusts herself".

The reflexive property requires that the entity involved is competent in the context in which the relation is defined; moreover, it captures the trust a principal poses on the hardware and security protocols used for implementing the system in which this principal belongs to. The reflexivity of the trust relation can be formalized as follows:

Definition 1 (reflexivity) *Let α be any principal. If a context ϕ exists such that $(\alpha, \alpha, \phi) \in \mathcal{T}$ then $\mathcal{T}$ is said to be reflexive and the implicit trust for principal α includes context ϕ.*

Moreover, a trust relationship might satisfy the *symmetrical* property, which can be defined as follows:

> "If Alice trusts Bob, Bob trusts Alice".

Note that symmetrical property states that for a trust relationship from Alice to Bob there is an inverse trust relationship from Bob to Alice, without any constraint about the contexts in which these two relations apply; for example, a client trusts a server for service provision, while that server trusts the authenticated client for the stated identity.

Common trust theories see symmetrical property tied to the same context and the same *trust degree* (see section 2.3). One such a definition is too strong for the model we are considering; for example, an web-buyer can believe a famous online-bookseller to be trustworthy in the context of book provision and this one can believe the client to be a legitimate visa card owner. In order to allow this kind of relationships, we are assuming a weaker definition of this property that sets apart constraints about the trust degree; we name this property *weak symmetry*, which we shall brifly refer to by the term symmetry.

Our formal definition of symmetry reads as follows:

Definition 2 (weak symmetry) *Let α, β be any two principals, $\alpha \neq \beta$. If there is a context ϕ_1 such as $(\alpha, \beta, \phi_1) \in \mathcal{T}$, then $(\beta, \alpha, \phi_1) \in \mathcal{T}$. If this assertion is true, then $\mathcal{T}$ is said to be symmetrical in the context ϕ_1.*

In the real world, any two entities carry out a communication successfully only if there is some kind of trust between them both, otherwise any communication between each other has no effect in that the exchanged information is ignored. From the definition of symmetrical property, it is possible to define when a communication can take place between any two principals; the following proposition defines the *communicability condition.*

Proposition 1 (communicability condition) *Let $\alpha, \beta \in P$ be any two principals belonging to the trust system $(P, \mathcal{T})$, $\alpha \neq \beta$, then a communication between them can take place if and only if $\mathcal{T}_{\{\alpha,\beta\}}$ is symmetrical in the context of authentication, where $\mathcal{T}_{\{\alpha,\beta\}}$ is the restriction of $\mathcal{T}$ over the set of principals $\{\alpha, \beta\}$.*

Optionally, trust relationships may also satisfy the transitive property. A typical application of this property is delegation; in the literature this property is quite discussed at length. As we will see, this property is necessary in collaboration oriented environments. In fact, in order to avoid the mapping of the complete identity set across every security domain (or sub-domain) of the environment, it is possible to address wide area access control by either the creation of principal's groups or delegation.

The creation of principal's groups is not scalable in geographical distributed systems because the group membership revocation to a principal would imply both a notification and the password change for every principal belonging to that group.

Delegation implies the adoption of trusted certifiers (either local or global) whose competence and authority are trusted (i.e. valid) only within the boudaries of a particular context; each of those, upon verification, can issue certificates to principals to be used as a proof of trustworthiness to other principals in the particular context in which the certifier operates. For example, Certification Authorities are global trusted certifiers, namely Trusted Third Parties, that are trusted in the Internet context to issue identity certificates or binding a public key to an identity certificate.

An informal definition for transitive property is the following:

> "If Alice trusts Bob and Bob trusts Cecilia, then Alice trusts Cecilia".

To be transitive, a trust relationship $\mathcal{T}$ must allow the involved entities to communicate with each other, therefore communicability condition must be included in the formalization of the property. Formally, this can be expressed as follows:

Definition 3 (transitivity) *Let ϕ_1, ϕ_2 be two contexts, not necessarily different from each other, and α, β, γ any three different principals such that $(\alpha, \beta, \phi_1) \in \mathcal{T}$ and $(\beta, \gamma, \phi_2) \in \mathcal{T}$. If the communicability condition holds for both $\{\alpha, \beta\}$ and $\{\beta, \gamma\}$ then a context ϕ_3 exists such that $(\alpha, \gamma, \phi_3) \in \mathcal{T}$. In this case $\mathcal{T}$ is said to be transitive.*

We find that such a definition is too weak and provides security administrators with too much freedom, allowing them to define also unjustified or sensless trust policies. Moreover, this semantic could be dangerous if implemented in a collaborative environment in that, in case of a collusion in the trust management system, it would allow a malicious party take advantage of the absence of constraints in order to gain access permissions over shared resources.

In real life, delegation is allowed only if there is knowledge about the delegator's competence and jurisdictional power over the delegate behaviour and ability. In the informal transitivity definition, it is not clear how the competence and jurisdictional power of Bob, the delegator, are expressed; it would be up to Alice, the trustor, to define the context in which trust would take place and the privileges granted to the trustee, Cecilia.

In order to express this idea in the transitivity definition, we must introduce what we call a *jurisdiction* predicate: the assertion "Bob has jurisdiction over the context ϕ" means that the principal Bob being either a local or global trusted delegator in the context ϕ, is able to verify the competence of a principal in order to delegate him/her work in this context and that Bob has revocation power over released delegation certificates. Therefore, the definition 3 changes as follows:

Definition 4 (transitivity (2)) *Let ϕ_1, ϕ_2 be two contexts, $\phi_1 =$ "jurisdiction over ϕ_2", and α, β, γ any three different principals such that $(\alpha, \beta, \phi_1) \in \mathcal{T}$ and $(\beta, \gamma, \phi_2) \in \mathcal{T}$. If the communicability condition holds for both $\{\alpha, \beta\}$ and $\{\beta, \gamma\}$ then $(\alpha, \gamma, \phi_2) \in \mathcal{T}$. In this case $\mathcal{T}$ is said to be transitive in the context of ϕ_1.*

In this section, we have discussed the main properties needed in order to describe the interactions between principals within a generic environment, before going on studying the properties of the environment, we shall talk about *trust degrees.*

2.3 Managing Trust Degrees

So far, we have seen a trust relationship $\mathcal{T}$ as a relation: given a couple of entities and a context, $\mathcal{T}$ indicates the boolean value expressing the existence of a trust relation without giving any other information about the "strength" of this trust binding.

In order to express this *trust degree*, we extend the trust relationship to return a continuous value between 0 and 1, with the assumption that the value 0 indicates distrust and 1 stands for full trust.

$$\mathcal{T} : P \times P \times \Phi \longrightarrow [0,1]\,.$$

As we said above, the introduction of trust values improves the expressivness of the trust function. By having a trust function instead of a trust relationship, we can embody in this function not only information about the existence of trust between any two principals, but we can specify by a value the strength of this trust relationship.

We also defined the trust function result as belonging to the closed set $[0,1]$ saying that the value 1 indicates "full trust" and 0 indicates "absence of trust". We can think the trust function to be a likelyhood function that returns the likelyhood a principal associates with a dependable behaviour of a second principal in a specified context.

By giving this definition, we also have to deal with the meaning of another value in the set of the possible results of the trust function, that is the value 0.5. For example, be $\mathcal{T}(Alice, Bob, \phi) = 0.5$, then the likelyhood Alice associates with a dependable behaviour of Bob in context ϕ is the same Alice associates with a malicious behaviour of Bob in the same context, she has no more information to decide wether to interact with Bob or not. In this case, the uncertainity about the expected behaviour is maximized and it should not take place any interaction between Alice and Bob that could expose her to troubles.

This model also allows to precisely formalize the intuition behind trust degree in delegation: in fact, if Alice trusts Bob in having jurisdiction over context ϕ with trust degree $\delta_{A,B}$ and Bob trusts Cecilia in context ϕ with trust degree $\delta_{B,C}$, then Alice would trust Cecilia in the same context with degree $\delta_{A,C} \leq \delta_{B,C}$. Here, the equality holds if and only if Alice fully trusts Bob as a delegator. Moreover, $\delta_{A,C}$ should also be upper bonded by $\delta_{A,B}$, in that it is not possible that a trustor puts more trust on the delegate than the trust he/she places on the delegator.

Therefore, by formalizing the above mentioned concepts, a transitive trust relationship must verify the *delegation conditions*, that are

1. $\delta_{A,C} = \delta_{B,C} \iff \delta_{A,B} = 1$
2. $\delta_{A,C} = 0 \iff \delta_{A,B} = 0 \vee \delta_{B,C} = 0$
3. $\delta_{A,C} \leq \delta_{B,C}$
4. $\delta_{A,C} \leq \delta_{A,B}$

The mathematical operator that satisfies these conditions is the multiplication, and the transitive trust can be expressed in terms of the direct trust relationships as follows:

$$\delta_{A,C} = \delta_{A,B} \cdot \delta_{B,C}$$

Trust degree could be taken into account by the trustor by either performing by being more or less pedantic about the verification of the trustee credential or by asking the trustee for a more or less frequent authentication according to the value the trust relationship assumes [10].

In the remaining of this paper, we shall make use of reflexive, symmetrical and transitive properties to describe the collaborative environment and the trust relationships that may be required inside it.

3 Trust in Collaborative Environments

In Section 2, we discussed the definition and semantic of trust in computer systems. We discussed three main properties that help describing real world trust relationships within practical implementations. We also discussed a condition for communicability among any two principals belonging to the environment in which the trust relation is defined, i.e. trust system. In this section, we will use these properties to describe basic abstraction of collaborative environments, namely *trust zones*. The propositions that we are going to introduce assume only the reflexive and symmetrical properties; the transitive property shall be introduced later in this section.

Informally, a trust zone can be defined as a trust system in which any two principals can effectively interact, i.e. exchange information, without requiring the intervenction of a third party acting in the role of a mediator (i.e. there can be direct exchange of information).

Let $(P, \mathcal{T})$ be a trust system, where P is the set of principals and $\mathcal{T}$ is the trust relationship between the principalsof the set P. Formally, a trust zone can be defines as follows:

Definition 5 (trust zone) *Let X be a set of principals and $\mathcal{T}_X \subseteq \mathcal{T}$ the restriction of $\mathcal{T}$ to the set X. $(X, \mathcal{T}_X)$ is termed trust zone over $(P, \mathcal{T})$ if for any two principals in X the communicability condition holds and P does not contain a set of principals $Y \supset X$ such that $(Y, \mathcal{T}_Y)$ with the same requirements.*

So, let $(X, \mathcal{T}_X)$ be a trust zone over $(P, \mathcal{T})$ then the set X is the biggest subset $\bar{X}$ of P containing X in which $\mathcal{T}_{\bar{X}}$, that is the restriction of $\mathcal{T}$ over $\bar{X}$, is symmetrical and reflexive.

From the definition of trust zone, it is possible to proof the following propositions which describe the main properties of trust zones:

Proposition 2 *Let $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$ be two trust zones over $(P, \mathcal{T})$, where $X \neq Y$. It can be proved that one and only one of the following two relations is valid:*

1. $X \cap Y = \emptyset$
2. $X \cap Y \neq \emptyset \wedge ((X \not\subseteq Y) \wedge (Y \not\subseteq X))$

This proposition follows from the definition of trust zone: it states that any two trust zones in the same environment can either be disjoint or have a common subset of entities without one being the subset of the other.

The next corollary directly follows from Proposition 2; it states that any two principals belonging to different trust zones can effectively interact with each other if each of them actually belongs to both the trust zones. Therefore, any two principals can effectively communicate with some information exchange if and only if the them belong to the same trust zone.

Corollary 1 *Let $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$ be two trust zones over $(P, \mathcal{T})$, where $X \neq Y$. Let $\alpha \in X$ and $\beta \in Y$ be two principals then the communicability condition among α and β holds if and only if:*

$$(\{\alpha, \beta\} \subset X \cap Y) \wedge (\alpha \neq \beta)$$

In the environment $(P, \mathcal{T})$ is then possible to define one or more trust zones; as we showed before, communication cannot take place among two principals belonging to different trust zones, in fact the corollary 1 states that communication may succeed only if the two entities belong to the same trust zone.

By introducing the transitive property, we can allow the principals belonging to two different trust zones to communicate, making use of the principals that are in the intersection of the respective sets of principals. The following theorem explain how communication can take place among different trust zones.

Theorem 1 (Interaction) *Let $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$ be two trust zones over $(P, \mathcal{T})$, with $X \neq Y$. If $\mathcal{T}_{X \cup Y} = \mathcal{T}_X \cup \mathcal{T}_Y$ is transitive in the context of authentication, then $X \cap Y \neq \emptyset$ and communicability condition holds for any two entities α, β, with $\alpha \in X$ and $\beta \in Y$.*

The interaction theorem states that if $\mathcal{T}_{X \cup Y}$ is transitive, then $X \cap Y$ includes at least one principal, say θ, that transitively allows the establishment of a trust relationship by acting as a trusted delegator. Of course, there could be a couple of delegators, θ and θ', such that α trusts β via θ and β trusts α via θ'. Informally, this means that in order to allow two ADs to communicate with each other (to simplify the notation, we make each one of them correspond to a trust zone), we must distinguish between two cases:

1. if the ADs are joint, then at least a principal belonging to the intersection must act as trusted delegator in order to satisfy the transitive property in authentication context and to grant trusted communication (see figure 1);

2. if the ADs are disjoint, then there is the need for another trust zone to act as a *bridge domain* (BD) between the trust zones relative to the first ADs; by chaining authentication transitiveness it is possible to achieve trusted communication between principals belonging to different disjoint ADs (see figure 2).

   The BD should embody a trusted delegator for each AD; moreover, it should be a trust zone in which every principal trusts any other principal in the context of having jurisdiction over authentication.

   Since there are not previous trust relationships between the two ADs (the fact that they are disjoint is a proof of that), a basic trust relationship has to be built between the trusted delagators by making use of cryptography (symmetric and asymmetric) and timestamps. For example, in [11] basic trust relationships are built making use of a PKI, whereas in [12, 13] it is introduced the use of timestamps in order to reduce the risk of message replay attacks.

Moreover, if the interactions between principals enclose some set of shared services, with a proper access control policy to be enforced, then the trusted delegators, belonging either to the BD or to the domains intersection, should also

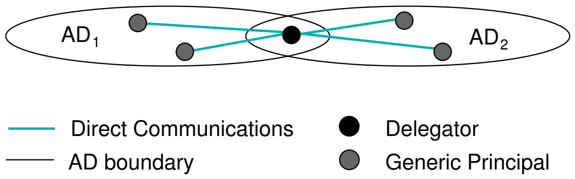

Figure 1: Transitive interactions between principals in different joint ADs.

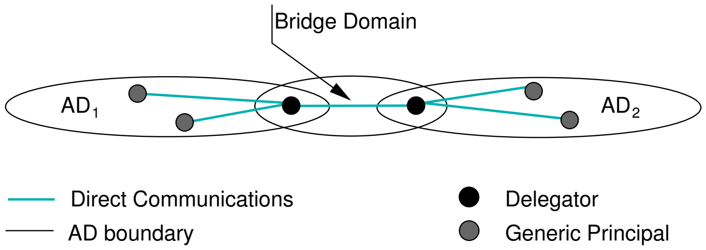

Figure 2: Transitive interactions between principals in different disjoint ADs.

have jurisdiction over the enforcement of the authorization policy that regulates those shared services.

An AD can also be modelled as a set of overlapping trust zones, where the communicability condition holds for every couple of principals; in figure 3 the example pertains an AD composed by two trust zones.

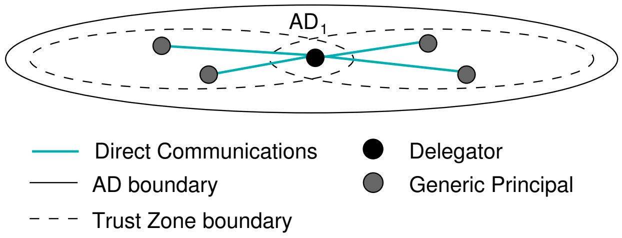

Figure 3: An AD composed by two overlapping trust zones.

The properties of the trust zones make them similar to real world security domains; trust zones are a good abstraction for a designer to model a collaborative environment by means of trust relationships and to implement it using a suitable trust management system.

4 Future Work

Going on, we shall consider only the TMS based on the approach given by in [4] that they grant scalability even in wide area contexts; current trust and access control technologies require that the resource owners take an active role in defining the access control policy and directly deal with the access control protocol.

Due to the heterogeneity which characterizes wide area distributed systems, addressing the creation coalitions by requiring the adoption of the same TMS could be infeasible. On the other hand, the recent middleware technologies (e.g., CORBA, J2EE, .NET, Grid) provides heterogeneous distributed systems with a common abstraction layer that allows complex applications to be developed in a way that is independent from the underlying architecture (e.g., physical machines, operating systems, security architectures).

In the context of resource sharing and access control in VOs, middleware technologies offer the chance of separating the management of the resource sharing policy from the management of the local policy of each AD that the VO embodies, provided that the local policies and the high level one do not contraddict each other. This is possible by i) implementing the VO resource sharing policy at middleware layer, taking advantage of the higher abstraction level, and ii) mapping the high level resource sharing policy into a local resource access policy for every AD that belongs to the VO.

According to their purpose, the middleware architectures implement ideal abstraction layer in which to deploy a TMS; this would provide a VO with a way to define and enforce a resource sharing policy that is independent from the underlying local security solution giving a unique semantic to the trust relationships spanning over different ADs.

Our TMS shall be designed according to the principle of trust zones in order to make the resulting architecture be totally decentralized. That TMS will provide an authorization service entity that will take the place of the trust delegator in a trust zone in order to allow communicability and, possibly, delegation in confront of principals belonging to other trust zones. Moreover, trusted delegators remove the need for having a global knowledge about the principals that belong to the VO.

Access to shared resources and services in one such an environment will be possible thorugh a Service Access Protocol (SAP) that shall be based on public key cryptography and the use of timestamps; moreover, communication contents will be protected by making use of OpenSSLv.3. In addition, SAP properties be the cornerstone for realizing trust in the bridge domain, allowing principals belonging to different, either joint or disjoint, ADs to communicate in a secure way and to catty out service provisions.

Even if one such a TMS is not tied to a particular middleware technology, a prototye of this authorization infrastructure is going to be implemented using JAVA technologies and deployed into the Objectweb's JOnAS J2EE platform.

5 Conclusion

In this paper we described a formal model that could help representing relationships among principals in collaborative environments; by introducing delegation issues, this model shows that it is possible to establish trust relationships among entities belonging to different security domains without the need of global knowledge of entities belonging to the system.

We also introduced a new TMS to be deployed into the middleware layer in order to manage access control over shared resources in VOs; one such a solution shall allow the enforcement of a global resource sharing policy in a VO context without requiring changes in the underlying security infrastructures by the ADs the VO embodies.

Acknowledgement

This work has been partially supported by European FP5 RTD project TAPAS (IST-2001-34069).

I also thank Carlos Molina, research associate in computer science at University of Newcastle upon Tyne, for his advices and accurate revisions.

References

[1] T. Grandison, M. Sloman, "A Survey of Trust in Internet Applications", *IEEE Communications Surveys*, Fourth Quarter 2000.

[2] M. Blaze, J. Feigenbaum, J. Ioannidis, A. Keromytis, "The Role of Trust Management in Distributed System Security", *Secure Internet Programming.*

[3] M. Blaze, J. Feigenbaum, D. Keromytis, "KeyNote: Trust Management for Public-Key Infrastructures", *Security Protocols International Workshop*, 1998.

[4] D. Ferraiolo, R. Kuhn, "Role Base Access Control", *Proceedings of 15th Nationa Computer Science Conference*,1992.

[5] N. Li, J. Mitchell, W. Winsborough, "Design of a Role-based Trust-management Framework", *2002 IEEE Symposium on Security and Privacy*, 2002.

[6] M. Blaze, J. Feigenbaum, J. Lacy, "Decentralized Trust Management", *IEEE Conf. Security and Privacy*, 1996.

[7] R. Hayton, J. Bacon, K. Moody, "OASIS: Access Control in an Open, Distributed Environment", *Proceedings of IEEE Symposium on Security and Privacy*, May 1998.

[8] J. Bacon, K. Moody, W. Yao, "Access Control in the Use of Widely Distributed Services", *Middleware 2001, Lecture Notes in Computer Science*, 2001.

[9] E. Freudenthal, T. Persin, L. Port, E. Keenan, V. Karamcheti. "dRBAC: Distributed Role Based Access Control for Dynamic Coalitions Environment", *Proceedings. 22nd International Conference on Distributed Computing Systems* , 2002.

[10] D. Manchala. "E-Commerce Trust Metrics and Models", *IEEE Internet Computing*, March 2000.

[11] K. Fürst, T. Schmidt, G. Wippel. "Managing Access in Extended Enterprise Networks", *IEEE Internet Computing*, September-October 2002.

[12] N. Mezzetti, A Secure and Anonymous Authorization Service for Grid Architectures, “Laurea” Degree Thesis in Computer Science, Department of Computer Science, University of Bologna, June 2002.

[13] N. Mezzetti, F. Panzieri, “The Data Grid: Security and Privacy Issues”, *4th European Dependable Computing Conference*, August 2002.