Modelling Trust Relationships in Collaborative Environments*

Nicola Mezzetti†

1 Introduction

In a global networking context, distributed environments for group collaboration are no longer closed into corporation boundaries; many are the cases in which several mutually distrustful institutions look for techniques to share services and resources in order to carry out a common task or to jointly provide a particular service. Informally, we say that these institutions make up a Virtual Organization (VO). A VO can be thought as a dynamical environment consisting of a set *"entities"* (e.g., users, resources, organizations) without any assumption about trust relationships between them.

Managing trust in collaborative environments which can be deployed in a global network is a real challange; a variety of *Trust Management Systems* (TMSs) [1, 2, 3, 4, 6, 8] have been proposed, most of them lack either flexibility or expressibility when applied to this context; in this paper we shall give a definition of what trust and trust relationships are, in order to describe the trust relationships that model the behaviour of entities in collaborative environments; moreover, we shall define the *trust zone*, i.e. the key abstraction to model real world collaborative environments.

2 On the Nature of Trust

In distributed systems, the entities of interest when speaking of trust are mostly individuals, resources and processes. From the trust viewpoint, there is no difference between these kinds of entities so we shall refer to a *principal* as an entity that can be involved in a trust relationship.

Before going on describing the trust relationships characterizing collaborative environments, we must understand what the word *trust* means and how trust relationships are conceived in distributed systems. The Oxford dictionary gives the following definition for the term "trust":

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

The former definition is quite good, even if IT researchers prefer the term "competence" reither than "stength" when referring to computer science applications. Moreover, the trust concept applies whithin a context; there is seldom full trust between two principals so saying "Alice trusts Bob" we mean that exists a context in which Alice believes in the reliability and competence of Bob.

Hence, the following definition explains what is meant by the term trust in information technology [1].

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

Given the above trust definition, we should see what is meant by *distrust* to understand the relationships between the various institutions in a community.

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

Whenever we say "Alice trusts Bob", then Alice is named *trustor* and Bob is the *trustee*. Here, trust is the key element by which it is possible that the trustee achieves some kind of information or some privilege upon a certain resource (or resource set) by the trustor.

Now, we can try to formalize the trust concept by giving three properties that could be useful when

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

†email: nicola.mezzetti@virgilio.it

[1] Here we assume that dependability covers reliability and timeliness

analyzing a real system to understand the underlying trust relationships.

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

We define "identity" to be the simplest context in which a principal can trust another one; this assumption is built on the fact that identity trust can be built on authentication, that is the base of every secure transaction in distributed environments.

Let P be the set of principals belonging to the trust system, then we shall represent the entire trust system as the couple $(P, \mathcal{T})$; $\mathcal{T}$ can be defined as

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

Basically, a trust relationship $\mathcal{T}$ respects the *reflexive* property; it can be described as follows:

reflexive[2]: Alice trusts herself[3];

Reflexive property can be formalized as follows:

Definition 1 (reflexivity) *Let α be any of 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 respect the *symmetrical* property:

symmetrical: If Alice trusts Bob, then there is an inverse trust relationship between Bob and Alice.

Note that symmetrical property states that for a trust relationship binding Alice to Bob there is an inverse trust relationship between Bob and 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*[4]; we are assuming a weaker definition of this property in order to define the right model of collaborative environments.

One such a property can be formalized as follows:

Definition 2 (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 there must be a context ϕ_2 such that $(\beta, \alpha, \phi_2) \in \mathcal{T}$. If this assertion is true, then $\mathcal{T}$ is said to be symmetrical.*

In the real world, any two entities carry out successfully a communication only if there is some kind of trust binding each other, otherwise any communication between each other has no effect. By 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 *Let $\alpha, \beta \in P$ be any two principals belonging to the trust system $(P, \mathcal{T})$, $\alpha \neq \beta$, then a bidirectional communication between them can take place if and only if $\mathcal{T}_{\{\alpha,\beta\}}$ is symmetrical, where $\mathcal{T}_{\{\alpha,\beta\}}$ is the restriction of $\mathcal{T}$ over the set of principals $\{\alpha, \beta\}$.*

Optionally, trust relationships may also have the transitive property, whose typical application is delegation. In the literature this property is quite discussed at lenght; as we will see, it 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, we can choose among the creation of group identities or delegation.

The first solution is not scalable in geographical distributed systems because a membership revocation to a principal would imply a password change for every principal in the group.

The latter solution implies the adoption of trusted certifiers (either local or global) with the task of having competence in a particular context; each of those, upon verification, can release certificates to principals in order to allow them to give other principals a proof of trustworthiness in the particular context in which the certifier operates. For example, certificates Authorities are global trusted certifiers, namely Trusted Third Parties, that are trusted in the context of releasing

[2] this property is also called *implicit trust.*

[3] This property requires that the entity involved is competent in the context in which the relation is defined.

[4] We shall define *trust degrees* in 2.1.

identity certificates or binding a public key to an identity certificate.

An informal definition for transitive property is the following:

transitive: If Alice trusts Bob and Bob trusts Cecilia, then there is a trust relationship between Alice and Cecilia.

In order 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.*

Althought some trust management systems [6] use this semantic for implementing transitivity, we find that such a definition is too weak and allows the definition of meaningless or illogical[5] trust relations. Moreover, this semantic could be dangerous if implemented in a collaborative environment in that it would allow a malicious party to use the absence of constraint to gain advantages over shared resources.

In real life, delegation is allowed only if there is knowledge about the delegator's competence and jurisdiction over the delegate behaviour and ability so that, in the informal transitivity definition, it would be up to Alice, the trustor, to define the context of trust and the privileges granted to Cecilia, the trustee.

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

[5] From the "good sense" point of view.

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 this section, we have given the main properties that allow 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.*

2.1 Managing Trust Degrees

Up to now, we have seen a trust relationship $\mathcal{T}$ as a boolean value: given a couple of entities and a context, $\mathcal{T}$ returns a value about the existence of the 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 0 means absence trust and 1 means full trust.

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

This model also allows to correctly formalize the intuition behind the 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 $\delta_{A,C} \leq \delta_{B,C}$, with equality if and only if Alice fully trusts Bob as a delegator.

This intuition can be formalized by assuming, in the case of delegation:

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

This relation verifies the *delegation conditions*, that are

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

The trust degree could be taken into account from trustor by performing for a more pedantic or verification of the trustee credential or by asking the trustee for a more frequent authentication as more the trust value is near the absence of trust [12].

In the rest of this paper, we shall make use of reflexive, symmetrical and transitive property in order to describe the collaborative environment and the trust relationships that may hold inside it.

3 Modelling Trust in Collaborative Environments

In section 2 we have given a definition and a semantic of what we mean by trust in information technology. We have pointed out three main properties to help describing real world trust relationships. We also have provided a condition for communicability among any two principals belonging to the environment in which the trust relation is defined. In this section we will make use of these properties to describe the base abstraction of collaborative environments, namely *trust zone*. From now on, we will make use only of reflexive and symmetrical properties; we shall introduce transitive property later in this section.

Let $(P, \mathcal{T})$ be the environment, with P the set of principals and $\mathcal{T}$ the trust relationship over P's principals; the following definition explains what a trust zone is.

Definition 5 (trust zone) *Let X 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 $\mathcal{T}_X$ is both reflexive and symmetric and P does not contain a set of principals Y such that $(Y, \mathcal{T}_Y)$ is a trust zone and $X \subset Y$.*

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; moreover, they point out the main properties of trust zones:

Proposition 2 *Let $(X, \mathcal{T}_X)$ be a trust zone, then for any two principals the communicability condition holds.*

Proposition 3 *Let $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$ be two trust zones over $(P, \mathcal{T})$, with $X \neq Y$, then one and only one of the follow relations are valid:*

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

The next corollary directly follows from propositions 2 and 3.

Corollary 1 *Let $(X, \mathcal{T}_X)$ and $(Y, \mathcal{T}_Y)$ be two trust zones over $(P, \mathcal{T})$, with $X \neq Y$. 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, then $X \cap Y \neq \emptyset$ and communication condition holds for any two entities α, β, with $\alpha \in X$ and $\beta \in Y$.*

The interaction theorem says that for $\mathcal{T}_{X \cup Y}$ to be transitive, $X \cap Y$ must include at least one principal, say it 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'.

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

4 A Practical Example

The authorization infrastructure for the DataGrid architecture introduced in [13, 14] subsumes the

principles introduced in this paper. In Grid, mutually distrustful physical organizations which can be deployed in a global network address security and trust issues by instantiating a local trusted party for each security domain to act as a delegator and certifier.

Inside each organization boundaries, trust is addressed by a local policy that defines the rights over shared resources of both external[6] and internal users; each physical organization is thus a security domain and can be defined as a trust zone $(O, \mathcal{T}_O)$ where O is the set of principals belonging to that organization and $\mathcal{T}_O$ the formalized expression of the local policy.

The same can be said about the set of the local certifiers of the organizations; they make up a trust zone to transitively allow trust relationship to cross organizationals boundaries, allowing thus collaborative resource sharing. Actually, local certifiers belonging to different physical organizations are mutually distrustful; in this context, trust relationships are established by legal contracts and enforced by cryptography mechanisms such as digital signatures, timestamps and receipts. The properties[7] granted by these building blocks allow the development of fault detection mechanisms and thus can make these principals trust each other by means of the security protocols they use.

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 avoiding the global knowledge of entities belonging to the system.

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] J. Barkley, K. Beznosov, J. Uppal, "Supporting Relationships in Access Control Using Role Based Access Control", *Fourth ACM Workshop on Role-Based Access Control*,1999.

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

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

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

[9] J. Bacon, K. Moody, J. Bates, R. Hayton, C. Ma, A. McNeil, O. Seidel, M. Spiteri, "Generic Support for Distributed Applications", *IEEE Computer*, March 2000.

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

[11] W. Yao, K. Moody, J. Bacon, "A Model of OASIS Role-Based Access Control and its Support for Active Security", *Proceedings of Sixth ACM Symposium on Access Control Models and Technologies, SACMAT 2001*, May 2001.

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

[6] Globus Toolkit's GSI provides services and APIs to map identities of remote domains into a local one.

[7] Respectively non-repudiation, originality and memory.

[13] 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.

[14] N. Mezzetti, F. Panzieri, "The Data Grid: Security and Privacy Issues", *Submitted to EDCC4 Call for Fast Abstract*, August 2002.