Trust-$\mathcal{X}$: An XML Framework for Trust Negotiations

Elisa Bertino[1], Elena Ferrari[2], and Anna Cinzia Squicciarini[1]

[1] Dipartimento di Informatica e Comunicazione
Università degli Studi di Milano
Via Comelico, 39/41
20135 Milano, Italy
Fax +39-0250316253
{bertino,squicciarini}@dico.unimi.it
[2] Dipartimento di Scienze Chimiche, Fisiche e Matematiche
Università dell'Insubria Como
Via Valleggio, 11
22100 Como, Italy
Fax +39-0312386119
elena.ferrari@uninsubria.it

Abstract. In this paper we present Trust-$\mathcal{X}$, a comprehensive XML-based [9] framework for trust negotiations. The framework we propose takes into account all aspects related to negotiations, from the specification of the profiles and policies of the involved parties to the determination of the strategy to succeed in the negotiation. In the paper we present the system architecture, and describe the phases according to which negotiations can take place.

1 Introduction

The extensive use of the web for exchanging information and requiring or offering services requires to deeply redesign the way access control is usually performed. In a conventional system, the identity of subjects is known in advance and can be used for performing access control. This simple paradigm is not suitable for an environment like the web, where the involved parties need to establish mutual trust on first contact, even if they are total strangers. A promising approach is represented by *trust negotiation* [7], according to which mutual trust is established through an exchange of digital credentials. Disclosure of credentials, in turn, must be protected through the use of policies that specify which credentials must be received before the requested credential can be disclosed.

A number of approaches to trust negotiation have been recently proposed [1], [5], [8], [6], [4]. However, all these proposals mainly focus on one of the aspects of trust negotiation, such as for instance policy and credential specification [1], or the selection of the negotiation strategy, but none of them provide a comprehensive solution to trust negotiation, able to take into account all the phases of the negotiation process. For this reason, we propose Trust-$\mathcal{X}$. Trust-$\mathcal{X}$ provides an

A. Lioy and D. Mazzocchi (Eds.): CMS 2003, LNCS 2828, pp. 146–157, 2003.

XML-based language, named $\mathcal{X}$-TNL, for specifying Trust-$\mathcal{X}$ certificates. Trust-$\mathcal{X}$ certificates convey information about the profile of the parties involved in the negotiation. The formalism we propose allows the specification of both *credentials* and *declarations*, where a credential is a set of properties of a party certified by a Certification Authority, whereas declarations contain information that may help the negotiation process (such as for instance specific preferences of one of the party) but do not need to be certified. All the certificates associated with a party are collected into its $\mathcal{X}$-$Profile$. In addition, to better structure credentials and declarations into an $\mathcal{X}$-$Profile$, each $\mathcal{X}$-$Profile$ is organized into *Data sets*. Each data set collects a class of credentials and declarations referring to a particular aspect of the life of their owner, and can be used to facilitate certificate exchange. We provide the definition of *disclosure policies*, encoded using XML. With the notion of disclosure policy we mean requirement needs for the release of a resource expressed by rules. Such rules regulate the disclosure of a resource by imposing conditions on the certificates the requesting party should possess and can be organized into groups of policies, ordered by a sensitivity level, to better protect flow of sensitive information that policies contain. A resource can be either a service, a credential, or any kind of data that need to be protected. In this paper we focus on the approach used in Trust-$\mathcal{X}$ for policy disclosures during negotiation and we present the negotiation system architecture. The paper is structured as follows. In Section 2 we present an overview of the Trust Negotiation language we have developed. We refer the readers to [2] for details on the $\mathcal{X}$-TNL language.
In Section 3 we introduce Trust-$\mathcal{X}$ negotiation, whereas Section 4 discusses some additional features of the Trust-$\mathcal{X}$ system. Finally, Section 5 concludes the paper.

2 Overview of $\mathcal{X}$-TNL, Trust Negotiation Language

In this section, we summarize the key elements of $\mathcal{X}$-TNL, the XML [9] language we have developed for specifying Trust-$\mathcal{X}$ certificates and policies. Then, we present $\mathcal{X}$-TNL *disclosure policies*, that is, policies regulating the disclosure of resources by imposing conditions on the certificates the requesting party should possess. A detailed presentation of $\mathcal{X}$-TNL can be found in [2].

2.1 $\mathcal{X}$-TNL Certificates

Constructs of $\mathcal{X}$-TNL include the notion of *certificate*, which are the means to convey information about the profile of the parties involved in the negotiation. A certificate can be either a *credential* or a *declaration*.

Credential. A credential is a set of properties of a party certified by a Certification Authority. $\mathcal{X}$-TNL simplifies the task of credential specification because it provides a set of templates called *credential types*, for the specification of credentials with similar structure. In $\mathcal{X}$-TNL, a credential type is modeled as a

```
<Library_Badge credID='12ab', SENS= 'NORMAL' >
<Issuer HREF='http://www.DigitaLibrary.com'
  Title=DigitaLibrary_Repository/>
  <name>
    <Fname> Olivia </Fname>
    <lname > White </lname>
  </name>
  <address>   Grange Wood 69 Dublin </address>
  <badge_number code=34ABN/>
  <e_mail> O.White@yahoo.com </e_mail>
  <position> Student   </position>
</Library_Badge>
```

Fig. 1. Example of Trust-$\mathcal{X}$ credential

DTD and a credential as a valid document with respect to the corresponding credential type. Each credential is digitally signed by the issuer Credential Authority, according to the standard defined by W3C for XML Signatures [9]. A credential is an instance of a credential type, and specifies the list of property values characterizing a given subject. A Trust-$\mathcal{X}$ credential is thus a valid XML document conforming to the DTD modeling the corresponding credential type. Figure 1 shows an example of credential, containing the basic information about a library badge issued by the digital library `Library_Badge`.

Declaration. Declarations are sets of data without any certification, therefore they are stated by their owner. Declarations can be considered as structured objects like credentials, collecting personal information about the owner. This kind of certificates provide thus auxiliary information that can help the negotiation process. For instance, a declaration named `book_preferences` describes the literary preferences of a given subject. In $\mathcal{X}$-TNL, we simply define a declaration as a valid XML document. Like credentials, also declarations are structured into declaration types, that are DTDs to which the corresponding declarations conform. Figure 2 shows the Trust-$\mathcal{X}$ representation of the `book_preferences` declaration. The declaration describes the genre of books Olivia prefers to read and lists some her favourite authors. This declaration can be used to communicate Olivia's personal preferences during negotiation with an online library.

2.2 Data Sets and $\mathcal{X}$-Profiles

All certificates associated with a party are collected into its $\mathcal{X}$-$Profile$. To better structure credentials and declarations into an $\mathcal{X}$-Profile, each $\mathcal{X}$-Profile is organized into *Data sets*. Each data set collects a class of credentials and declarations referring to a particular aspect of the life of their owner. For instance, `Demographic_Data, Education, Working Experience` are examples of possi-

```
<book_preferences >
  <name>
    <Fname> Olivia </Fname>
    <lname > White </lname>
  </name>
  <book_genre> horror   </book_genre>
    < author>
      <name> Stephen King </name>
      <name> John Smith </name>
    </author>
      <interests >
      Cinema tab     </interests >
</book_preferences>
```

Fig. 2. Example of an $\mathcal{X}$-TNL declaration

ble data sets.[1] For example, Alice's certificates concerning working experiences can be collected in the `Working Experience` data set. In this group of digital documents we can find Alice's work license number, a digital copy of her last job contract and some uncertified information about her precedent job experiences. Organizing certificates into data sets facilitates their retrieval during negotiation. Indeed, all the certificates collected in the same data set are logically related. Data sets can then be used to refer to a set of homogeneous declarations or credentials as a whole, and this can facilitate their evaluation and exchange during negotiation.

2.3 Disclosure Policies

Trust-$\mathcal{X}$ disclosure policies are specified by each party involved in a negotiation, and state the conditions under which a resource can be released during a negotiation. Conditions are expressed as constraints against the certificates possessed by the involved parties and on the certificate attributes. Each party adopts its own policies to regulate release of local information and access to services. Similar to certificates, disclosure policies are encoded using XML[2].

Trust-$\mathcal{X}$ policies are thus defined for protecting any kind of sensitive resources. Additionally, a resource can be characterized by a set of attributes, specifying relevant characteristics of the resource that can be used when specifying disclosure policies. To make easier resource management and protection, resources can be further classified into *simple* and *composite* resources. Trust-$\mathcal{X}$ language includes specific formalism to define resources relationship.[2] Intuitively, a simple resource is an atomic service or information whereas a composite resource R can be thought as the composition of several resources $R_1, R_2, ..R_k$. Resources $R_1, R_2, ..R_k$ in turn, can be either simple or composite and therefore

[1] Like for credentials we assume that data set names are unique and are registered through some central organization.

[2] We omit the complete specification for lack of space.

```
<!DOCTYPE policyBase[
<!ELEMENT policyBase(policySpec)+>
<!ELEMENT policySpec (properties, resource, type)>
<!ELEMENT properties (DELIV |certificate+) >
<!ELEMENT resource EMPTY>
<!ELEMENT type EMPTY >
<!ELEMENT certificate (certCond*) >
<!ELEMENT DELIV EMPTY >
<!ELEMENT certCond (#PCDATA) >
<!ATTLIST certificate targetCertType CDATA #REQUIRED>
<!ATTLIST DELIV value CDATA #FIXED 'DELIV' >
<!ATTLIST resource target CDATA #REQUIRED >
<!ATTLIST type value (CERT|SERVICE) 'SERVICE' >
] >
[]
```

Fig. 3. Trust-$\mathcal{X}$ policy base template

may be released singularly or all together as R. Each resource R_i $i \in [1, k]$ can have its own disclosure policies. Resource R itself can also have its own disclosure policies. When R is requested, the related disclosure policy is obtained by processing all policies for R, if any, and of $R_1 R_2, .., R_k$. An example of composite resource may be a theatre package, including a theatre ticket for a performance, the corresponding script, and a dinner at the restaurant of the theatre. The resource may be likely also characterized by a set of attributes giving information related to the performance, that is, the name of the show and the exact location and time where it is performed.

Different policies for the same sensitive resource denote alternative policies equally valid to obtain it. Each resource R can be disclosed if and only if one of the corresponding policies is satisfied. In addition, the disclosure policy language can be used to specify prerequisite information. Such policies denote conditions that must be satisfied for a resource request to be taken into consideration, and are therefore used at the beginning of the negotiation process, how explained in Section 3. Figure 3 shows the template of $\mathcal{X}$-TNL policy base. According with our XML compliant formalism, the template correspond to a DTD, whereas a policy base is the corresponding valid XML document.

3 Trust-$\mathcal{X}$ Negotiation

The reference scenario for Trust-$\mathcal{X}$ negotiations is a network composed of different entities that interact with each other by exchanging sensitive resources controlled by the entities themselves. The notion of resource comprises both sensitive information and services, whereas the notion of entity includes users,

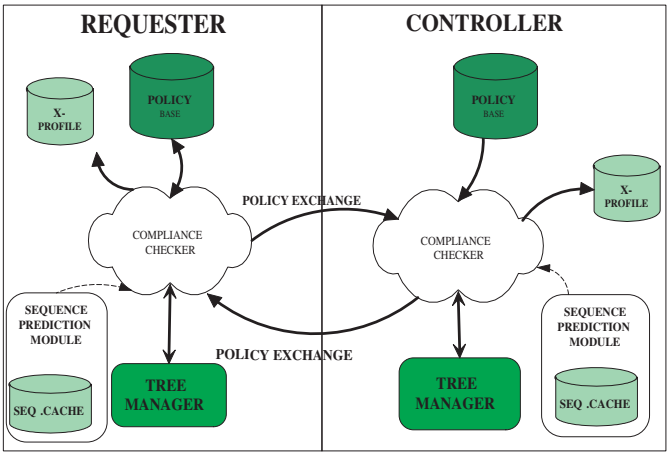

Fig. 4. Architecture for Trust$\mathcal{X}$ negotiation

processes, servers. Entities are characterized by a set of credentials, issued by CAs. Each credential describes attributes characterizing the owner, and they are used as a means to certify properties of the parties. A negotiation involves two entities (also named parties); each of them has a Trust-$\mathcal{X}$ profile of certificates, conforming to the syntax introduced in the previous section. During a negotiation mutual trust might be established between the controller and the requester: the requester has to show its certificates to obtain the resource, and the controller, whose honesty is not always assured, submits certificates to the counterpart in order to establish trust before receiving sensitive information. Release of information is regulated by disclosure policies, introduced in section 2.3, and by appropriate policies that govern access to protected resources by specifying credential combinations that must be submitted to obtain authorization. Disclosure policies are exchanged to inform the other party of the trust requirements that need to be satisfied to advance the state of the negotiation.

The main components of the Trust-$\mathcal{X}$ architecture are shown in Figure 4. Note that the architecture is peer-to-peer. The goals of the system components are essentially the following: supporting policy exchange, testing whether a policy might be satisfied, supporting certificate and trust ticket exchange, and caching of sequences. Each of these functions is executed by a specific module of the Trust-$\mathcal{X}$ system. Facets modules may be also added to make the negotiation easier and faster, but we omit them to focus on the most relevant components. The system is composed by a *Policy Base*, storing disclosure policies, the $\mathcal{X}$-*Profile* associated with the party, a *Tree Manager*, storing the state of the negotiation, and a *Compliance Checker*, to test policy satisfaction and determine request replies. In the following, we assume that both parties are Trust-$\mathcal{X}$ compliant but it is possible to enforce negotiations even between parties that do not adopt the same negotiation language, simply by adding a translation mech-

anism to guarantee semantic conversion of possibly heterogeneous certificates. In the following we illustrate in more details the negotiation phases.

3.1 Negotiation Phases

A Trust-$\mathcal{X}$ negotiation is structured according to the following phases: introductory phase, the policy evaluation phase, certificate exchange phase. The policy evaluation phase is the core of the negotiation process. Certificates and services are disclosed only after a complete counterpart policies evaluation, that is, only when the parties have found a trust sequence of certificate disclosures that makes it possible the release of the requested resource, according to the disclosure policies of both parties. Even disclosure of sensitive policies may be protected, by disclosing sensitive policies gradually according to the degree of trust established. The concluding step of a Trust-$\mathcal{X}$ successful negotiation is an analysis of the certificates and information exchanged and may result in caching the generated sequence, as illustrated in section 4.2. In what follows we illustrate each one of the above-mentioned phases in more detail. Note that articulating the negotiation into different distinct phases results in a multilevel protection mechanism, that avoids the release of unnecessary or unwanted information. Each phase is executed only if the previous ones succeed and sensitivity of the exchanged information increases during negotiation. Indeed, the disclosure of certificates, that can be regarded as more sensitive with respect to policies, is postponed at the end of policy evaluation: only if there is any possibility to succeed in the negotiation, certificates are effectively disclosed.

Introductory Phase. The introductory phase begins when a requester contacts a controller asking for a resource $\mathcal{R}$. The initial phase of a negotiation is carried out to exchange preliminary information that must be satisfied in order to start the actual processing of the resource request. Such exchange of information is regulated by the introductory policies of both parties. Introductory policies are used to verify properties of the counterpart that are mandatory to continue in the negotiation. For instance, a server providing services only to registered clients, before evaluating the requirements for the requested service can first ask the counterpart for the login name. If the requester is not registered there is no reason to further proceed. Prerequisite policies are therefore essentially used to establish whether to enter into the core of the negotiation process or not, but they also can also help in driving the following phases of the process. Introductory policies may also be used to collect information about the requester preferences and/or needs. For instance, in a purchasing of books online, a book store may ask the requester to submit the `book_preferences` declaration, if any, in order to satisfy customer preferences. If the requester does not assume honesty of the controller it can, in turn, send its own introductory policies. Such phase is therefore composed by a short number of simple messages exchanged between the two parties.

Example 1. Consider a Rental Car Service. When Alice asks to rent a car the negotiation starts. Possible prerequisites that the agency may require during introductory phase are modeled by the following introductory policies[3]:
• $Car\_Rental_p \leftarrow Preferred\_customer()$
• $Car\_Rental_p \leftarrow Car\_Preferences()$.
The first policy checks whether the client has the *preferred customer* credential denoting previous business relationship between the parties, whereas with the second policy the Agency asks the requester to submit the `car_preferences` declaration, if any, in order to satisfy requester preferences.

Policy Evaluation Phase. During this phase, both client and server communicate disclosure policies adopted for the involved resources. The goal is to determine a sequence of client and server certificates that when disclosed enable the release of the requested resource, in accordance to the disclosure policies of both parties. This phase is carried out as an interplay between the client and the server. During each interaction one of the two parties sends a set of disclosure policies to the other. The receiver party verifies whether its $\mathcal{X}$-Profile satisfies the conditions stated by the policies, and determines a policy counter request regulating the disclosure of the certificates requested. If the $\mathcal{X}$-Profile of the receiver party satisfies the conditions stated by at least one of the received policies, the receiver can adopt one of two alternative strategies. It can choose to maximize the protection of its local resources replying only for one policy at a time, hiding the real availability of the other requested resources, or, alternatively, it can reply for all the policies to maximize the number of potential solutions for negotiation. Additionally, when selecting a policy each party determines whether its preconditions are verified by the policies disclosed until that point, and, only in this case, the policy is selected. By contrast, if the $\mathcal{X}$-Profile of the receiver party does not satisfy the conditions stated by the received policies, the receiver informs the other party that it does not possess the requested certificates. The counterpart then sends an alternative policy, if any, or it halts the process, if no other policies can be found. The interplay goes on until one or more potential solutions are determined, that is, whenever both the parties determine one or more set of policies that can be satisfied for all the involved resources. The policy evaluation phase is mostly executed by the *Compliance Checker*, whose goal is the evaluation of remote policies with respect to local policies and certificates (certificates can be locally available in the $\mathcal{X}$-Profile or can be retrieved through certificate chains), and the selection of the strategy for carrying out the remainder of the negotiation. To simplify the process a tree structure is used which is managed and updated by the *Tree Manager*.

Note that no certificates are disclosed during the policy evaluation phase. The satisfaction of the policies is only checked to communicate to the other party the possibility of going on with the process and how this can be done. A

[3] Policies are expressed in terms of logical expressions to simplify the comprehension of the corresponding semantic

detailed description of the Trust-$\mathcal{X}$ negotiation, with the associated algorithms for all the negotiation strategies can be found in [3].

Certificate Exchange. This phase begins when the previous phase ends successfully, determining one or more trust sequences. Several sequences of credentials can be determined to succeed in the same negotiation. Once the parties choose the sequence of certificates to disclose, the certificate exchange starts. Each party discloses its certificates, observing the order defined in the sequence. Upon receiving a certificate, the counterpart verifies the satisfaction of the associated policies, checks for revocation, checks validity dates and authenticates the ownership (for credentials). Eventually, if further information is needed for establishing trust, it is the receiver responsibility to check for new certificates using credential chains. For example, if a medical certificate was requested and the issuer is an unknown hospital, the receiver party has to check the validity of issuer certificate by collecting new certificates from issuer repository. The receiver then replies with an acknowledgment expressed with an `ack message`, and asks for the following certificate in the sequence, or if it has received all the certificates of the set, it sends a certificate belonging to the subsequent set of certificates in the trust sequence. If no unforeseen event happens, the exchange ends with the disclosure of the requested resource. Figure 5 shows an example of the messages exchanged by two parties performing a car rental negotiation. Suppose that parties have successfully completed policy evaluation phase and have determined the subsequent sequence (each name denotes a certificate): [{Corrier_Affiliation}, {Corrier_ Employee}, {Rental_Car}]. The disclosure of certificates begins with submission of `Corrier_Affiliation` credential, which should satisfy the conditions specified during policy evaluation phase. The subsequent certificate (`Corrier_Employee`) is disclosed from the counterpart after checking the remote certificate received. *Rental Service* is finally provided and the negotiation succeeds.

4 Additional Features of Trust-$\mathcal{X}$

In this section we discuss two additional aspects of Trust-$\mathcal{X}$ system, that is, the negotiation of multiple resources and the possibility of caching the sequence of certificates as a mechanism to speed up negotiation processes.

4.1 Negotiation of Composite Resources

As previously introduced, Trust-$\mathcal{X}$ resources can be either simple or composite, that is, obtained as the composition of several resources. The negotiation of composite resources can be considered as an extension of negotiation of simple ones and it is supported as follows. Each component of the main resource is sequentially negotiated. When a party asks for a composite resource, the negotiation is

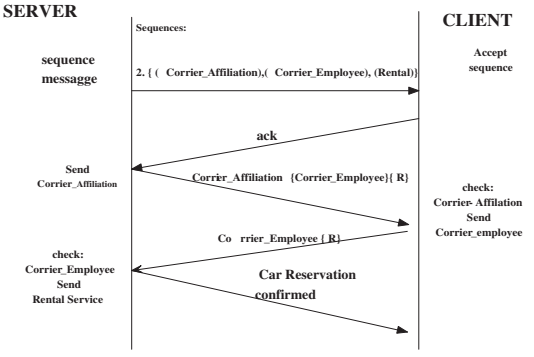

Fig. 5. An example of certificate exchange phase

executed evaluating first the policies for the root resource[4] and then negotiating each component at the time. Each time the policy evaluation phase of a resource component succeeds, the corresponding trust sequence is determined. Note that, although it is possible to determine more than a sequence for the same resource, we now refer to the sequence on which parties agree. Instead of immediately executing it, both parties store it and start evaluating requirements for negotiating the next component resource. Intuitively, the information obtained from the previous process can be used to make this phase simpler and faster, by avoiding to ask for the same certificates or properties again. This simplification consists in not asking again the satisfaction of disclosure policies that were already satisfied in previous policy exchange. This simple strategy reduces the number of party requests and optimizes the number of exchanges. In addition, the controller can choose whether allow partial disclosure of some of the components or not. Indeed, suppose that during a negotiation of a composite resource (composed by three atomic resources, say), a party results to be in order to obtain only two of the three resource components of the resource originally requested. What should the controller do? It can choose to disclose the two resources anyway or deny the entire access. Intuitively, this decision strongly depends from the context and from the kind of the resources parties are negotiating. If parties are part of a collaborative environment and resources are not logically dependent from each other, the disclosure can be granted, otherwise, it is denied.

4.2 Sequence Caching

It is really likely that some negotiations will be performed by an entity with different counterparts having a similar profile. In such cases it might be useful to keep track of the sequences of certificates most frequently exchanged, instead

[4] Policies may be specified for both the main resource and for resource components too.

of recalculating them for each negotiation. For instance, some books of a digital library might be often asked during exam sessions. The trust negotiation process will be very similar for different students. Or, better, the asked information will be exactly the same and the only differences will concern the type of certificates disclosed. As an example, consider a student attending a certain University Y. Students might have a card issued by the main secretary office of the university, whereas students of a branch department might have only a badge issued by the departmental secretary office. Suppose that the properties required to obtain the authorization to access the library can be proved by presenting either the card issued by the main secretary office or by presenting both the badge issued by the departmental secretary office and the student library badge. Suppose moreover that students usually require privacy guarantees before disclosing certificates, and that the library proves its honesty by a set of proper certificates. Alternative sequences can therefore be generated to disclose the same resource, but all of them are quite intuitive and easy to determine. The controller can cache and suggest them upon receiving a request from a student. The student can cache the most widely used sequences for negotiating access to digital libraries as well as the library, and suggest them upon sending a request to a library.

Intuitively, this approach does not ensure a complete protection of policies. However, in many contexts, protection of policies and associated certificates is not the main goal for parties. Moreover, we expect that in many scenarios there will be standard, off-the-shelf policies available for widely used resources (e.g., VISA cards, passports). In case of negotiations involving such common resources the sequences of certificates to be disclosed will be only regulated by such standard and predictable policies. In this case, certificates represent only a means to easily prove parties properties, and it is not unsafe to suggest the sequences at the beginning of the process. If the counterpart can not satisfy the proposed sequence, the negotiation can continue by executing the policy evaluation phase or by suggesting the another sequence. The module of Trust-$\mathcal{X}$ in charge of caching and suggesting the sequences is the *sequence prediction module*.

5 Conclusion

In this paper we have introduced Trust-$\mathcal{X}$, a comprehensive XML-based framework for trust negotiations. We have focused on the various phases in which a Trust-$\mathcal{X}$ negotiation is articulated. Future work includes the extension of $\mathcal{X}$-TNL among several directions, such as for instance the possibility of specifying the credential submitter. Another extension we are currently working on is the possibility of disclosing only portions of a credential during the negotiation process. This will allows us to protect the elements of a credential in a selective and differentiated way. Finally, we are developing techniques and algorithms for credential chains discovery and for recovering from a negotiation failure and we are optimizing caching strategy to increase the efficiency of negotiation.

References

1. E. Bertino, S. Castano, E. Ferrari. "On Specifying Security Policies for Web Documents with an XML-based Language". *Proc. of SACMAT 2001, ACM Symposium on Access Control Models and Technologies*, Fairfax, VA, May 2001.
2. E. Bertino, E. Ferrari, A. Squicciarini, "$\mathcal{X}$-TNL - An XML Based language for Trust Negotiations". *Proceedings of Fourth International Workshop on Policies for Distributed Systems and Networks*, Como, Italy, June 2003.
3. E. Bertino, E. Ferrari, A. Squicciarini, "Trust-$\mathcal{X}$ A Peer to Peer framework for Trust Establishment". *Submitted for Publication.*
4. M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis, The KeyNote Trust-Management System *RFC 2704*, September 1999.
5. T. Yu, M. Winslett, K. Seamons, "Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation" *ACM Transactions on Information and System Security, volume 6, number 1*, February 2003.
6. P. Bonatti, P. Samarati, "Regulating Access Services and Information Release on the Web" *7th ACM Conference on Computer and Communications Security*, Athens, Greece, November 2000.
7. K. E. Seamons, M. Winslett, T. Yu, B. Smith, E. Child, J. Jacobson, H. Mills, and L. Yu. "Requirements for Policy Languages for Trust Negotiation". *International Workshop on Policies for Distributed Systems and Networks (POLICY 2002)*, Monterey, CA, June 2002.
8. A. Herzberg, Mihaeli, Y. Mass, D. Naor, and Y. Ravid, "Access Control System Meets Public Infrastructure, Or: Assigning roles to Strangers" *IEEE Symposium on Security and Privacy*, Oakland, CA, May 2000.
9. World Wide Web Consortium. Available at `http://www.w3.org/`