PP-Trust-$\mathcal{X}$: A System for Privacy Preserving Trust Negotiations

A. SQUICCIARINI and E. BERTINO
Purdue University
ELENA FERRARI
Universita' degli Studi dell'Insubria, Varese
F. PACI
Universita' degli Studi di Milano, Milano
and
B. THURAISINGHAM
The University of Texas at Dallas

Trust negotiation is a promising approach for establishing trust in open systems, in which sensitive interactions may often occur between entities with no prior knowledge of each other. Although, to date several trust negotiation systems have been proposed, none of them fully address the problem of privacy preservation. Today, privacy is one of the major concerns of users when exchanging information through the Web and thus we believe that trust negotiation systems must effectively address privacy issues in order to be widely applicable. For these reasons, in this paper, we investigate privacy in the context of trust negotiations. We propose a set of privacy-preserving features for inclusion in any trust negotiation system, such as the support for the P3P standard, as well as a number of innovative features, such as a novel format for encoding digital credentials specifically designed for preserving privacy. Further, we present a variety of interoperable strategies to carry on the negotiation with the aim of improving both privacy and efficiency.

Categories and Subject Descriptors: K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; D.4.6 [**Operating Systems**]: Security and Protection—*Accsess controls, information flow controls*

A preliminary version of this paper has been presented in *4th Workshop of Privacy Enhancing Technologies*, Toronto, Canada, May 2004.

Authors' address: A. Squicciarini and E. Bertino, CERIAS and Computer Science Department, Purdue University, West Lafayette, IN 47907; email: bertino@cerias.purdue.edu; squiccia@cs.purdue.edu. Elena Ferrari, Dipartimento di Informatica e Communicazione, Universitá degli Studi dell'Insubria, Varese; email: elena.ferrari@uninsubria.it; F. Paci, Dipartimento di Informatica e Communicazone Universitá degli Studi di Milano, Milano; email: paci@dico.unimi.it. B. Thuraisingham, Department of Computer Science, The University of Texas, Dallas, Texas; email: bhavani.thuraisingham@utdallas.edu.

General Terms: Algorithms, Design, Security

Additional Key Words and Phrases: Access control, attribute-based access control, automated trust negotiation, credentials, privacy, strategy

ACM Reference Format:

Squicciarini, A., Bertino, E., Ferrari, E., Paci, F., and Thuraisigham, B. 2007. PP-trust-$\mathcal{X}$: A system for privacy preserving trust negotiations. ACM Trans. Info. Syst. Secur. 10, 3, Article 12 (July 2007), 50 pages. DOI = 10.1145/1266977.1266981 http://doi.acm.org/ 10.1145/1266977.1266981

1. INTRODUCTION

The recent increase in web-based applications carried out on the Internet has been accompanied by vast amounts of data exchanged and collected by the interacting entities. As the amount of exchanged information grows exponentially, privacy [Westin 1967] has emerged as one of the most crucial and challenging issues. Current researchers are thus focusing on developing systems for supporting on line secure resource sharing [Winsborough and Li 2002b; Yu and Winslett 2003] and on technologies for preserving user privacy in a standardized and automated manner [Agrawal et al. 2003]. The most significant proposal for supporting privacy over the Internet is represented by the Platform for Privacy Preferences—P3P [Cranor et al. 2003]. The designers of P3P have also developed a preference language, called APPEL [Cranor et al. 2002], to allow users to express their privacy preferences, thus enabling automated matching of privacy preferences against P3P policies.

Privacy is crucial when dealing with trust management. Today, among the various approaches that can be adopted for exchanging resources and services on the web, a promising model is represented by trust negotiations [Winslett et al. 2002]. Existing trust negotiation systems are based on the disclosure of a certain amount of sensitive information, usually conveyed by *digital credentials*, required to establish trust. However, although several efficient and powerful negotiation systems have been developed so far [Bonatti and Samarati 2000; Herzberg and J. Mihaeli 2000; Seamons et al. 2001; Yu and Winslett 2003; Winsborough and Li 2002a], none of them provide a comprehensive solution to protect privacy during the negotiation process. In particular, none of them support the P3P standard [Cranor et al. 2003]. Starting from an analysis of the most relevant privacy pitfalls in trust negotiation, in this paper we revise all the key aspects of trust negotiation, which are crucial in order to efficiently and effectively preserve privacy. We cast the proposed features in the framework of Trust-$\mathcal{X}$ system [Bertino et al. 2004b]. However, we believe that they can be easily applied to other negotiation systems as well.

To help the reader in better understanding the Trust-$\mathcal{X}$ negotiation system, we now provide an overview of its main features.

1.1 Trust-$\mathcal{X}$ Overview

Trust-$\mathcal{X}$ is a comprehensive framework for trust negotiations, providing both an XML-based language, referred to as $\mathcal{X}$-TNL, to encode policies and certificates, and a system architecture. Trust-$\mathcal{X}$ certificates describe qualifying properties of the negotiating parties. Such certificates are collected into $\mathcal{X}$-profiles, which are associated with each Trust-$\mathcal{X}$ party. Digital credentials are assertions describing one or more properties of a given subject, certified by trusted third parties.

Protection needs, for the release of a resource, are expressed by *disclosure policies*. A resource can be either a service, a credential, or any kind of data that needs to be protected. Disclosure policies regulate the disclosure of a resource by imposing conditions on the credentials the requesting party should possess. Disclosure policies for a resource can be gradually released according to the degree of trust established, in order to ensure a better protection of the sensitive information exchanged.

Trust-$\mathcal{X}$ also comprises an architecture for negotiation management, which is symmetric and peer-to-peer. A Trust-$\mathcal{X}$ negotiation consists of a set of phases to be sequentially executed. The idea is to disclose policies at first, in order to limit credential release, and then disclose only those credentials that are necessary for the success of negotiation. The key phase of a Trust-$\mathcal{X}$ negotiation is the *policy evaluation phase*, which consists of a bilateral and ordered policy exchange. The goal is to determine a sequence of credentials, called *trust sequence*, satisfying disclosure policies of both parties. Once a trust sequence has been determined, the *credential exchange phase* is executed. Each time a credential is received, the local compliance checker module checks local policy satisfaction and verifies at runtime the validity and ownership of the remote credentials. More details on Trust-$\mathcal{X}$ can be found in Bertino et al. [2004b].

1.2 Main Contributions of the Paper and Applicability to Digital Identity Management

The work reported in this paper is built on top of the Trust-$\mathcal{X}$ negotiation system [Bertino et al. 2004b], summarized above. However, the current work substantially extends Trust-$\mathcal{X}$. In particular, the definition of the actions to carry on a negotiation, the ability of the different strategies to trade-off between privacy and efficiency, and the notion of strategy interoperability are novel aspects that have not been previously considered.

We propose the support for different credential formats and the notion of *context* associated with a policy, which allows one to both express privacy policies and to convey information, which can be used to speed up the negotiation process. We also integrate P3P policies at various steps of the negotiation and present new strategies to carry on a negotiation, which improve both robustness and efficiency, while, at the same time, preserving privacy. We prove correctness of our proposed strategies and give examples of applications in different real-world scenarios. We also illustrate the prototype system we have developed and present the evaluation results for assessing the performance of the implemented strategies. Finally, to enable privacy in trust negotiations, we propose the integration of the system with existing standard privacy technologies, such as P3P and APPEL.

It is also important to note that our approach is in line with current efforts in the area of digital identity management [Microsoft 2004]. An example in this direction is represented by the InfoCard project, currently ongoing at

Microsoft. InfoCard is specifically tailored to protect end user's digital identities against tampering and spoofing and to maintain end-user control. Identities in InfoCard are actually encoded through credentials retrieved from the identity providers and encoded in security tokens. The system we have designed is similar to InfoCard in that our digital credentials have the same function as user identity tokens. Disclosure of such credentials in our system is protected by users' defined policies and can also be selectively executed. In addition, we provide users with the capability of negotiating such identities in order to strongly establish mutual trust using different strategies on the basis of the user preferences. However, our system has several significant additional features with respect to InfoCard. Currently, negotiations envisioned in InfoCard are based on simple matching between parties claims or requirements. Our approach provides a rich set of negotiation features governing the matching process. Moreover, our approach is symmetric; each party has to disclose credentials to the other party. By contrast, in InfoCard, the client has to present identity cards in order to obtain a service from a service provider, whereas the provider is not required to present any identity card to the client. Therefore, our work may potentially be applied as a well articulated and powerful approach to identity management and disclosure of identity information among parties. We can expect that capabilities, such as the ones proposed by our approach, to be part of the next-generation digital identity management systems.

1.3 Paper Roadmap

The remainder of this paper is organized as follows. The next section presents the main privacy pitfalls compromising trust negotiations and introduces possible remedies. Section 3 introduces a running example to illustrate the discussion in the paper. Section 4 provides an overview of PP-Trust-$\mathcal{X}$ main protocols. Section 5 presents the Trust-$\mathcal{X}$ language. Section 6 illustrates the different negotiation strategies supported by Trust-$\mathcal{X}$, while, in Section 7, we discuss possible application domains of the proposed protocols. Section 8 analyzes potential attacks that might thwart the system and discusses possible countermeasures. Sections 9 and 10 deal with the system architecture and its implementation. In particular, Section 10 illustrates the main features of the system architecture and overviews the tests we have carried out on the implemented prototype. Section 11 illustrates related work in the area. We conclude the work in Section 12. The paper also contains three Appendices, reporting the main algorithms developed for trust negotiation management, examples of privacy policies for trust negotiations, and formal proofs of the theoretical results presented in this paper.

2. PRIVACY PITFALLS AND REMEDIES IN TRUST NEGOTIATIONS

Trust negotiation systems by their very nature may represent a threat to privacy. Credentials, exchanged during negotiations, often contain sensitive information, which, thus, need to be adequately protected. A party may also want to minimize the released information, thus enforcing the need to know principle in disclosing its credentials to others. To limit credentials disclosure, a solution is to postpone the credentials exchange until disclosure policies for all the involved resources have been evaluated [Winsborough and Li 2002a] and the set of relevant credentials is identified. Such an approach, however, has several drawbacks. First, when a request for a credential is sent by a party, the counterpart typically replies by sending a counterrequest for credentials necessary to disclose the credential originally requested. Thus, the receiver can infer that the counterpart can satisfy the request, obtaining clues about the possession of sensitive credentials, even if it never actually obtains the credential. Further, during policy exchange it is not possible to determine whether a party is faithfully following the trust negotiation protocol until the credentials are actually disclosed. A naive solution to this inference problem is to disclose credentials as soon as a corresponding policy has been satisfied, thus prior to the end of the policy evaluation phase. However, this strategy may result in unnecessary credential disclosures, as well as needless rounds of negotiation, even when the negotiation cannot succeed. An ideal system should be able to prevent any information leakage, without necessarily extending the negotiation process. A possible solution, proposed by Winsborough et al. [Winsborough and Li 2002a], is to introduce the notion of attribute acknowledgment policies, to uniformly respond whether or not a subject, in fact, possesses a given attribute. These policies are disclosed at each negotiation round in reply to credential requests, independently of the actual possession of the required credential.

Such an approach has the drawback of making the negotiation process longer than necessary, as the policies for nonpossessed attributes/credentials have to be evaluated. Further, it requires a user to specify a number of policies greater than her actual necessity. A further issue related to credentials arises because of the sensitive attributes (e.g., age, credit rating) a credential may contain. A credential may contain several sensitive attributes and, very often, just a subset of them is required to satisfy a counterpolicy. However, when a credential is exchanged, the receiver gathers all the information contained in the credential. Although approaches have been proposed [Brands 2000] for encoding digital credentials, no widely accepted standards exist for supporting partial disclosures of credential contents.

In the current paper, we address some of above issues by extending the techniques introduced in Bertino et al. [2004a]. Instead of introducing a policy for each sensitive attribute or jeopardizing private information by immediately disclosing credentials (as in the eager strategy proposed in Winsborough et al. [2000]), we provide a set of language tools for driving the negotiation, while sending the policy, in a way that better fulfills the user's privacy requirements.

One of the tools allows the user to anticipate, whenever possible, counterpart policy requests, by specifying within a local policy the next credentials the policy sender is willing to disclose as a guarantee for the properties required in the policy just sent. By suggesting subsequent credentials, a form of "barter" may be established.

Since suggestions precede any possible request, this avoids unwanted enquiries without jeopardizing the possibility of successfully terminating the negotiation. To address the issue of credentials conveying multiple sensitive attributes, we propose a new credential format, supporting partial disclosure of credentials, and allowing subjects to give proof of credential possession without actually revealing any sensitive attributes.

Finally, because a relevant issue is related to the lack of control or safeguard of the disclosed personal information, we propose an approach that integrates trust negotiation systems with the P3P platform. Our introduced negotiation system does not, however, mandate the use of P3P. The negotiation parties are free to use such a privacy standard.

3. RUNNING EXAMPLE

The scenario we refer to throughout the paper is that of a social network. A social network comprises a set of people with a pattern of interactions among them. Nodes of the network are individuals within the same social context and edges represent interactions and collaborations between entities. Social networks are highly dynamic, growing and changing very quickly over time through the addition of new edges and/or nodes representing the appearance of new interactions/members in the underlying social structure. Typically, nodes do not know each other and they interact to share knowledge, services, and resources. Some of the interactions among nodes can be well modeled according to a trust negotiation scheme, as we will explain in what follows, and nodes can follow different approaches to negotiate on the basis of their security requirements and relationships.

For our work, we refer to the specific network of a mentoring program (referred to as WICE, Women in Chemical Engineering), aimed at providing a social network and environment of understanding and support for students dealing with the cultural issues associated with being a woman in Chemical Engineering today. WICE organizes lecture series about women in chemical engineering, describing the stories of their research and academic interests in the context of their lives as students, academics, and women in chemistry. It also provides the possibility for the nodes to share, in a peer-to-peer fashion, data, research reports, and other pieces of information. Users of the network can also supply some services under payment, such as students mentoring, proofreading of thesis or project reports, and sale of used textbooks. The network involves different organizations and schools at different levels and the exchanged information among nodes is often confidential or private. Nodes can also form smaller networks or groups, related to projects or particular areas of interest. Principal institutions of *WICE* are three chemical engineering departments at three universities (*Milano, Western, ETNH*), two colleges (*IMR, HR*) and two chemical research centers (*AbCent, ChemyDrug*). Subjects taking part in the network are qualified by a specific credential, issued by a trusted network coordinator. In particular, throughout the paper, we consider two users, Alice, a professor at the chemical engineering department of the Western University, and Mary, a PhD student at IMR. Suppose that Mary wants to apply for a job at *AbCent* research labs. As the position supports diversity and WICE provides help and support for applicants being WICE members, Mary needs to engage in a negotiation to have her application reviewed by a consultant of the WICE network. Alice is involved in several research projects. In addition, as a professor, she also provides consulting activities to help WICE members to write project proposals. Western University reports are disclosed by Alice only to other WICE investigators, proving their partnership to some specific projects and providing evidence of the need for the reports to proceed with the related study. Suppose that Western University web server adopts specific privacy policies on the data of the requesting subjects. In addition, it also specifies stringent privacy preferences to prevent nonauthorized users from collecting any type of information related to the Western University projects. Finally, since Alice and Mary are privacy-conscious subjects, they want their personal information to be used only to complete their negotiations and thus they will carry on privacy-preserving trust negotiations. In addition, they will negotiate only after being informed of and having agreed on the privacy practices of the entities they negotiate with.

As the paper proceeds, we will use this scenario to illustrate how PP-Trust-$\mathcal{X}$ carries out trust negotiations while enforcing privacy preferences of both negotiation parties.

4. OVERVIEW OF PRIVACY-ENHANCED TRUST NEGOTIATIONS WITH PP-TRUST-$\mathcal{X}$

A Trust-$\mathcal{X}$ negotiation consists of two main phases: the policy evaluation phase, devoted to policy exchange and the credentials disclosure phase. In addition, a Trust-$\mathcal{X}$ negotiation includes an *introductory phase*, which is an optional phase to let the negotiators exchange preliminary information about the process to be executed (e.g., the specific resource to be negotiated, technical details of the negotiation). Before going into the details of PP-Trust-$\mathcal{X}$, we give an overview of the extensions, we propose to this scheme to preserve privacy. To enable privacy-preserving trust negotiations, we have enhanced the introductory phase with the option of exchanging privacy policies on the data to be negotiated. More precisely, the introductory phase contains a specific subphase, referred to as *privacy agreement* subphase, whose goal is to reach a preliminary agreement on personal data collection and usage before starting the actual negotiation. The agreement, resulting from the mutual exchange of information characterizing a negotiation, is reached by communicating to the counterpart both privacy practices and preferences, using coarse-grained P3P policies and privacy preferences rules. Note that this approach is also valuable for asymmetric scenarios (typical of e-commerce transactions), where only one of the two parties actually enforces privacy policies to be matched against the other party's privacy preferences. We assume that the privacy preferences are expressed using APPEL [Cranor et al. 2002], although other languages can be used as well. Coarse-grained privacy policies specify to the counterpart the types of data the negotiator will collect without enumerating every piece of individual data. These types of policies can be implemented using the P3P syntax by describing the data using the `<dynamic><miscdata/><dynamic>` element and the categories to which the information to be exchanged belongs to. Once a prior agreement on privacy is reached, parties can enter into the core process and start the policy evaluation phase.

Example 1. With respect to our reference scenario, suppose Helen, a researcher at Milano University, is contacting Alice at Western University to get a project proposal.

Alice's privacy policies are summarized as follows. To accept any incoming request, Alice needs to obtain information concerning the involvement of the requester in the project. Alice stores the information gathered for a certain time interval before discarding it, to monitor the advances of the project. As Western University representative, Alice's privacy policy also states that she collects personal information of users contacting and downloading information from the server of the university. The resulting P3P policy to be matched against Helen's privacy preferences is shown in Figure B.1 of Appendix B. The first STATEMENT says that personal information and miscellaneous data (like negotiation timestamp and project tasks) will be used for completing the negotiation. The second STATEMENT allows Alice's site to use miscellaneous data for creating personalized user profiles. Under a user perspective, Alice also has her own privacy preferences to be evaluated against the other party's privacy practices. Alice's privacy preferences assert her concerns in revealing personal data, as she wants to carry on negotiations following a parsimonious approach. That is, she has no refrain in negotiating on behalf of the university she represents, but she does not want her identity to be revealed. As a university representative, she also does not allow any user to collect information about the exact data to be downloaded for longer than required by the specific purpose of the negotiation. All the above privacy preferences will be matched against the counterpart privacy policies during the privacy agreement phase.

Figure 1 illustrates possible approaches to privacy protection during the different steps of a negotiation. As shown, a negotiation starts with the introductory phase (step (1)), which may or may not include a *privacy agreement* subphase (step 1a). The subsequent phases are: policy evaluation phase (2), credential disclosure (3), and resource disclosure (4). Phase (2), which is the core of all the trust negotiation systems, has been enhanced in PP-Trust-$\mathcal{X}$ to include privacy policy exchange. Privacy policies are exchanged during such a phase only if a prior privacy agreement subphase has not been executed or specific privacy policies have to be applied to some of the requested credentials. Indeed, during the introductory phase, the parties are not aware of the exact credentials to be negotiated. As such, disclosure policies requiring credentials having specific privacy policies might be involved. If desired by the parties, disclosure policies can thus be sent together with the related P3P policy using a specific field of the policy, as we will illustrate in Section 5. The P3P policy will specify how the information collected by means of the requested credentials will be managed and for what purposes. We refer to such policies as *fine-grained* privacy policies. An example of such types of policies is given in Figure B.2 of Appendix B. Each time a P3P policy is received, the receiver has to check whether his/her privacy preferences comply with the P3P policy.

Like privacy policies, privacy preference rules can either express coarse-grained preferences to be exchanged in the agreement phase, or finer-grained preferences associated with credentials considered privacy sensitive.

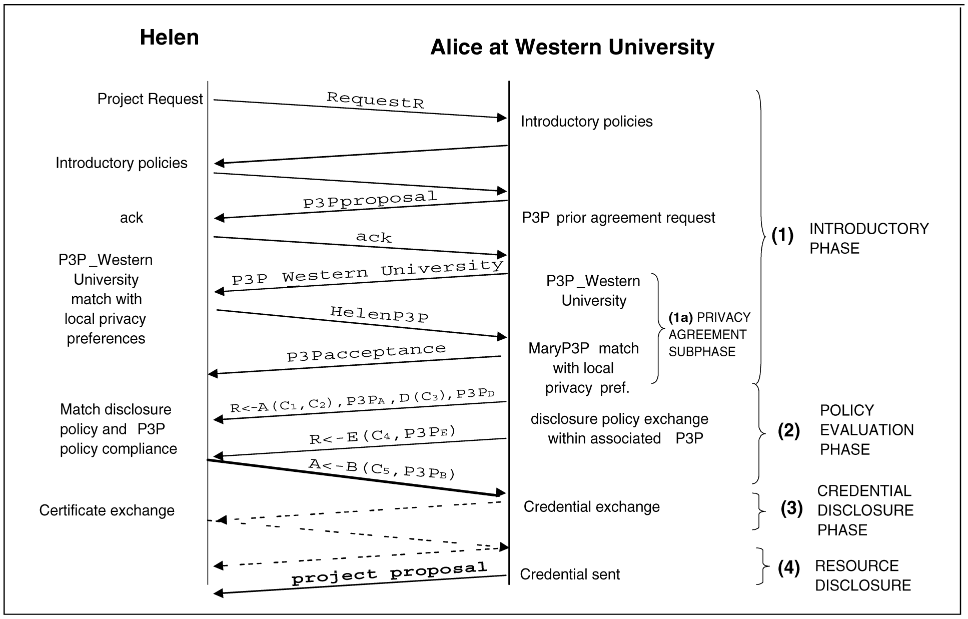
Helen
Alice at Western University
Project Request
RequestR
Introductory policies
Introductory policies
P3Pproposal
ack
P3P prior agreement request
ack
P3P _Western University match with local privacy preferences
P3P Western University
P3P _Western University
HelenP3P
(1a) PRIVACY AGREEMENT SUBPHASE
P3Pacceptance
MaryP3P match with local privacy pref.
(1) INTRODUCTORY PHASE
Match disclosure policy and P3P policy compliance
R<-A(C1,C2),P3PA,D(C3),P3PD
disclosure policy exchange within associated P3P
R<-E(C4,P3PE)
A<-B(C5,P3PB)
(2) POLICY EVALUATION PHASE
Certificate exchange
Credential exchange
(3) CREDENTIAL DISCLOSURE PHASE
project proposal
Credential sent
(4) RESOURCE DISCLOSURE

Fig. 1. Sketch of a negotiation process.

Coarse-grained preference rules are high-level assertions stating how the data to be released should be preferably treated by the counterpart. By contrast, fine-grained preference rules will be targeted to specific credentials and might be communicated to the counterpart while the related credentials are involved in the process.

Example 2. Suppose that during the negotiation illustrated in Example 1, Alice requires Helen's ID card and she collects from this card her mailing address to contact her later for issues related to the project. As mentioned, Alice (as for the WICE network rule) wants to retain Helen's email and telephone number for a long time. As such, the coarse-grained policy that the parties agreed upon is not enough for this specific data. Therefore, the disclosure policy requiring Helen's ID Card will be sent along with a fine-grained P3P policy, represented in Appendix B, Figure B.2, specifying Helen's ID card specific handling by Alice. Unlike the coarse-grained policy of Figure B.1, the retention time of the ID Card information is set to `indefinitely`, implying that the information will be retained for an indeterminate period of time.

5. A PRIVACY PRESERVING SPECIFICATION LANGUAGE

In what follows, we first present an extension to the standard credential format to support our approach to privacy. Then, we introduce the notion of *context* associated with a policy, which can be used to attach privacy rules to a policy, as well as to protect policy disclosure and to speed up the negotiation process.

5.1 Privacy-Enhanced Credentials

During trust negotiations credentials play a key role, in that they represent the means to prove properties of parties. Thus, credentials must be unforgeable and verifiable. Typically, a credential contains a set of attributes specified using name/value pairs and is signed by a trusted issuer.

Our system supports two types of credentials. The first one, called *basic format* (see Bertino et al. [2004b]), represents the standard approach for credential encoding, that is, a digitally signed document containing a set of subject properties [Housley et al. 2002]. The second scheme, which we propose in this paper, is a *privacy-enhanced* format and is based on a credential template supporting the disclosure of the credential in two different steps, to keep the sensitive content of the credential secret until the end of the negotiation. The proposed format also supports partial disclosure of credential content to protect the privacy of sensitive attributes. In the subsequent sections, we first illustrate the technique used to support partial credential disclosure and then we present the privacy-enhanced format. We do not further elaborate on the basic format and refer the reader to Bertino et al. [2003].

5.2 Protecting Credential Attributes

An interesting approach to maximize privacy protection is to selectively disclose attributes in a credential, so that only the required subset of information is made available to the recipient of the credential. The best approach, currently available, supporting partial disclosure of credentials, relies on the use of the bit-commitment technique [Naor 1990], which enables users to commit a value without revealing it. By exploiting this technique it is possible to actually send credentials by only revealing the minimal set of attributes required during the negotiation. Instead of using the bit-commitment technique, we adopt a multibit hash-commitment technique for attribute encoding, as the length of the attributes will likely be longer than 1 bit. The general protocol that is followed to issue credentials with protected attributes is briefly summarized in what follows.

An entity wishing to obtain a credential from a Certification Authority (CA) either generates the set of attribute values for the credential or it asks the CA to generate such values, depending on the credential content and type. In order to create a credential with protected attributes, the requester has first to create the corresponding private values to be used in place of the sensitive ones. Given a sensitive attribute `a` with value `va` the operations needed for its protection are: (1) generate a random string `r`; (2) compute `p=va|r`, that is, the concatenation of `va` with `r`; (3) compute `v=hash(p)`, where hash is a one-way multibit hash function. Upon the completion of the above process, the requester submits the attribute values to the credential authority, which verifies them, generates the corresponding credential, and, finally, signs the credential.

During a negotiation, the credential can be sent by keeping the content of some of the attributes secret. The disclosure of each private attribute is executed by sending the counterpart both `va`, the original value, and `r`, the random value, so that the receiver can compute the hashed value using the same hash function and verify the attribute validity. The remaining sensitive attributes of a credential that are not relevant for the negotiation are hidden.

A credential is an instance of a *credential type*, which is a DTD (document-type definition) used as a template for credentials having a similar structure. Although the content of a credential is determined by the corresponding type, each credential, beyond the specific language used for its encoding, must always convey some general reference information about the corresponding credential type, the issuer, and its temporal validity. This set of information is crucial for proving that the credential, besides its specific content, is a signed and valid digital document issued by an entity that is trusted. The credential format we have devised captures this reference information into a specific portion of the document, called the *header*, which is kept separate from the private *content* of the credential. Further, the credential content is structured by using the multibit technique discussed above, so that partial disclosure of attributes can be achieved. Such an approach enables the negotiating parties to adopt new strategies to gradually establish trust in that the header and content can be disclosed at different times during a negotiation. For instance, one possible strategy is to disclose the header to prove credential possession as the credential is involved in a negotiation, and keep the credential content secret until the end of the entire process. An alternative approach can be that of requiring attributes to be disclosed as soon as they are requested by a policy. The header disclosure can then be immediately followed by the disclosure of the required attributes and the corresponding random values.

Although Trust-$\mathcal{X}$ provides an XML-based encoding of credentials, a privacy-enhanced credential template is a language-independent mechanism for encoding credentials. Thus, in what follows, we give a logic definition, abstracting from the specific Trust-$\mathcal{X}$ syntax. Formally, a credential type ct can be represented as a pair $\langle n_{ct}, p_{ct}\rangle$, where n_{ct} is the name of the credential type and p_{ct} is the set of corresponding attribute specifications. Each attribute specification contains a name and a domain. We represent a privacy-enhanced credential template as follows.

Definition 5.1. (**Privacy-enhanced credential template**). Let $ct = \langle n_{ct}, p_{ct}\rangle$ be a credential type. A *privacy-enhanced credential* template (pt_{ct}) for ct is a pair $\langle header\text{-}template, content\text{-}template\rangle$ where:

- *header-template* is a set containing the specification of the following attributes:
  1. `credID`, specifying the credential identifier;
  2. `CredType`, identifying the type of the credential n_{ct};
  3. `notBefore`, specifying the credential validity starting date;
  4. `notAfter`, specifying the credential expiring date;
  5. `IssueRep`, denoting the unique address of the issuer's server.[1]
- $content\text{-}template$ is a list of attribute specifications in p_{ct}.

Throughout the paper we denote with $sign(doc)$ and $hash(doc)$ the signature and the hash value computed over document doc.

[1] Identified by a URI [World Wide Web Consortium b].

We are now ready to define a privacy-enhanced credential.

Definition 5.2. (**Privacy-enhanced credential**). Let *ct* be a credential type and let pt_{ct} =<*header-template, content-template*> be the privacy-enhanced credential template for *ct*. A privacy-enhanced credential *pc* instance of pt_{ct} (where *ct* is a credential type) is a tuple <*header, content, sign(header|content)*>, where:

- For each attribute specification *p* in *header-template*, *header* contains a pair (*p-name*, *p-value*), where *p-name* is the name of the attribute specified by pt_{ct}, and *p-value* is a value compatible with the domain specified in pt_{ct};
- For each attribute specification in *content-template*, *content* contains *hash(p-name|p-value|random)*, where *p-name* is the name of the attribute specified by *p*, *p-value* is a value compatible with its domain, and *random* is a random string of bits;
- *sign(header|content)* is the signature of the header concatenated with the credential content, generated by the issuing CA.

Example 3. Figure 2 shows an example of the XML encoding of a privacy-enhanced credential. The `<Header>` and `<Content>` elements represent, respectively, the *header* and *content* components in the privacy-enhanced credential tuple. The `<Signature>` element represents the signature affixed on the `<Header>` and `<Content>` elements. `<Signature>` is composed of several subelements to specify how the signature is generated; that is, the `<SignatureMethod>`, `<DigestMethod>`, and `<Reference>` elements to specify which XML elements have been signed.

`<SignatureValue>` subelement contains the signature value, while `<KeyInfo>` element contains the CA's public key that must be used to validate the signature.

A *credential proof* corresponds to a particular view of a privacy-enhanced credential, in which the header is in clear, whereas the whole content is hidden, but the signature over the credential can be verified.

During the policy evaluation phase, when a credential is requested and the corresponding policy is satisfied, the corresponding credential proof can be safely disclosed. In this way, the recipient is ensured that the other party possesses the requested credential, even if it cannot immediately access its sensitive content. The protocol for proving a credential is illustrated in Figure 3.

At the end of the policy evaluation phase, when a trust sequence has been found, all or a subset of the random values are disclosed, allowing the receiver to verify credential properties and the actual attribute values. If the verification process fails, the receiver can notify this to the credential issuer and abort the negotiation. The protocol to reveal attribute content is illustarted in Figure 4.

5.2.1 *Negotiating Trust with Privacy-Enhanced Credentials.* With privacy-enhanced credentials, trust negotiation has the potential to increase its success rate and performance. The following examples show how privacy-enhanced credentials may be used to successfully complete a negotiation by strongly protecting privacy and, at the same time, deal with situations which

```
  <Credential>
  <Header>
    <CredType>Wice_Card</CredType>
    <CredID>210</CredID>
    <Issuer HREF="http://www.ItalyCountry.com" Title=" KTHUUniversity_Repository "/>
    <Expiration_Date>
      <NotBefore>6-07-2007 </NotBefore>
      <NotAfter>18-12-2007g </NotAfter>
    </Expiration_Date>
  </Header>
  <Content>
    <Attribute Id="1">TmFtZXxPbGl2aWE=</Attribute>
    <Attribute Id="2">U3VybmFtZXxTdWxsaXZhbg==</Attribute>
  </Content>
  <Signature>
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
      <Reference Id="1" URI="">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
          <Transform Algorithm="http://www.w3.org/2002/06/xmldsig-filter2">
            <XPath Filter="union">Header|Content</XPath>
          </Transform>
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <DigestValue>tbUZmXYOB2G2EqKO2+mx45o19nQ=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>
     MCwCFHTVf89qw5PJ+K9CZn5eFJATQRzJAhQRvto+TKA9McqqGDsnsaNWiswWGw==
    </SignatureValue>
    <KeyInfo>
      <KeyValue>
        <DSAKeyValue>
          <P>17801190547854226652823756245015999014523215636912067427327445031444286
             57887370207706126952521234630795671567847784664499706507709207278570500
          </P>
          <Q>864205495604807476120572616017955259175325408501</Q>
          <G>174064887611810308304375498519098347260155094691329488083395492313850000361646 4</G>
         <Y>159213235452455555555555553135455865553555554761089660154512686954921676956314 56</Y>
        </DSAKeyValue>
      </KeyValue>
    </KeyInfo>
  </Signature>
</Credential>
```

Fig. 2. An example of a privacy-enhanced credential.

would result in negotiation failures if conventional trust negotiation protocols were adopted.

Example 4. Eve is an active member of the *WICE* association and is in charge of reviewing and counseling the job applications. To offer such a service, Eve needs to obtain Mary's WICE card and additional information about her working and personal skills. (Rules expressing such requirements are formalized in Example 6.) Mary, on the other hand, in order to release the required information wants to have proof of the recipient's (Eve, in the specific case) role in the WICE network. Thus, Mary is willing to disclose the requested credentials only if Eve first shows her credential stating her membership to the WICE network as a consultant. In addition, in order not to be rejected without consideration of merit, her application cannot be accessed to members of the hiring committee before the official submission to e *AbCent*. Thus,

```
Requester::    Request $cred$
Cred_owner::   Send $\langle cred.header$, hash$(cred.p\text{-}name_1|cred.p\text{-}value_1|random_1)$, ...,
                 hash$(p\text{-}name_k|p\text{-}value_k|random_k)$), sign(($cred.header$,
                 hash$(cred.p\text{-}name_1|cred.p\text{-}value_1|random_1)$, ...,hash$(cred.p\text{-}name_k|$
                 $cred.p\text{-}value_k|random_k))\rangle$
Requester::    Check $cred.header$
Requester::    Verifies sign(($cred.header, hash(cred.p\text{-}name_1|cred.p\text{-}value_1|random_1)$,
                 ..., hash$(cred.p\text{-}name_k|cred.p\text{-}value_k|random_k))$
```

Fig. 3. Protocol for credential verification.

```
Cred_owner::  Send $name_j, value, random$
Requester::   Compute $H = hash(value|random)$
Requester::   Compare $hash(name_j|value_j|random_j)$ previously received with $H$
Requester::   Evaluation of $value_j$
```

Fig. 4. Protocol to reveal the value of attribute with name $name_j$.

she needs to make sure about the affiliation of the recipient of her application material.

Her policy rules will request Eve's ID not to be issued by *AbCent* and her role at WICE corresponding to "Senior Consultant."

In conventional negotiation, rules such as the ones illustrated in Example 4, will result in a *negotiation deadlock*, since each of the negotiators adopt policies which require the recursive disclosure of the same credential (i.e., the WICE_Card). This would result in a negotiation failure, even if both parties possess the requested properties and associated credentials. Such deadlock can be avoided using privacy-enhanced credentials. During the policy evaluation phase, parties may prove to each other the possession of the credentials without actually revealing the credential content until they receive all the requested credential proofs. Alternatively, they might reveal upfront the relevant attributes and skip the credential exchange phase. The hash commitment technique can, indeed, be used each time the selective disclosure of a credential is required.

Example 5. Following from Example 4, Eve's trust negotiation agent[2] requires Mary's WICE Card. Mary's WICE card, in turn, cannot be disclosed until the counterpart's WICE card is received, as explained in the previous example. Eve can solve the deadlock by sending her card with all her sensitive information hidden (her identifying data, for example) not strictly required for satisfying Mary's policy requirements. This would satisfy Eve's policy and let the negotiation advance. Finally, note also that Eve may choose to only disclose the WICE Card header to Mary, so that Mary can safely disclose hers and the parties can, later on, selectively disclose the remaining attributes required for satisfying the policies.

[2] As we will discuss in the concluding section of the paper, a trust negotiation can be conducted either directly by the end user or it can be carried out by system agents.

5.3 Policy Language

Besides credentials, trust negotiations rely on disclosure policies, specifying trust requirements that must be satisfied in order to access the requested resource. Disclosure policies are assumed to be expressed as finite set of statements in a formal language with well-defined semantics. Beyond the specific formalism adopted, disclosure policies can be modeled as logic rules specifying two types of information: the target resource for which the policy is specified and the credentials to be disclosed, eventually specifying conditions against them. The ensuing definition formalizes this concept, by using the Trust-$\mathcal{X}$ syntax. The target resource is denoted as R, whereas the requested credentials are denoted by means of *terms*. A term specifies a credential name and eventually conditions against some of its attribute values. Resources are described using expressions, named *R-Term*, of the form *ResName*(*attrset*). *ResName* denotes a resource name whereas *attrset* denotes a set of attributes, specifying relevant characteristics of the resource used when specifying disclosure policies. In our reference scenario, example of R-Terms are *JobApplication*(*ApplicantName*, *ApplicationDate*, *Skills*) and *Project_Proposal(Partners-List, Date_of_Issue, ActivityLines)*, where the first summarizes the request of a review of a job application to WICE member. The latter R-term denotes a classified report part of a WICE project proposal at Western University. The attributes within brackets represent examples of possible attributes relevant for the service/resource disclosure. For instance, in *JobApplication*, attributes *Skills* correspond to certified skills the applicant has, while in *Project_Proposal*, the attributes qualify the validity and the structure of the project.

Definition 5.3. [Bertino et al. 2004b] **(Rule).** A rule is an expression of one of the following forms:

1. $\mathbf{R} \leftarrow \mathcal{T}_\mathbf{1}, \mathcal{T}_\mathbf{2}, ., \mathcal{T}_\mathbf{n}$, n $\geq$ 1, where $\mathcal{T}_\mathbf{1}, \mathcal{T}_\mathbf{2}, ., \mathcal{T}_\mathbf{n}$ are terms and **R** is an *R-Term* identifying the name of the target resource.
2. $\mathbf{R} \leftarrow$ **DELIV**. Rule of this form is called *delivery rule*.

A rule is *satisfied* if the stated credentials are disclosed to the policy sender and their conditions (if any) evaluated as true, according to the specific credential content. A delivery rule implies that resource R is ready to be released and no specific requirement is to be satisfied.

Example 6. With respect to our reference scenario, example of rules are the following:

r_1: JobApplication(applicantName, applicationDate, skills)← WICE_Card(), AwardStatement(Issuer=AmChemAss, Name=WICE_Card.LName, IssueYear>2004), RecommendationLetters()

r_2: Project_Proposal(Partners-list, date_of_issue, ActivityLines)←WICE_Card(Institution ∈ Project_Proposal.Partners), ChemistryAssociation(Lname= WICE_Card.Lname, dep= Chemical Engineering).

Rule r_1 states that to obtain a free consult from WICE counsellors, job applicants have to be part of the network and provide signed recommendation letters. In addition, applicants have to be recognized as good standing researchers from the American Chemistry Association in the last 2 years.

Rule r_2 regulates the disclosure of project proposals and requires disclosure of a WICE membership card to verify members' eligibility to access this kind of report. Precisely, the name of the institution and the department in which the member is employed are required, along with the proof of the ChemistryAssociation affiliation.

Although the above specification captures all the basic information required to carry on a negotiation, it is not expressive enough to specify other crucial information that may be associated with a policy, such as, for instance, its usage, its prerequisites, or the privacy policies for the requested credentials. For this reason, we enhance the notion of policy by adding a set of information referred to as *policy context*. The goal is to integrate the rules defined according to Definition 5.3 with a structured set of data to be used during trust negotiation. Before formally defining a PP-Trust-$\mathcal{X}$ policy, we thus need to introduce the notion of policy context.

Definition 5.4. **(Policy context)**. Let $r = R \leftarrow \mathcal{T}_1, \mathcal{T}_2, ., \mathcal{T}_n$ or $r = R \leftarrow$ *DELIV* be a rule. A **context** for r is a data object with the following structure: $\langle pol\_prec\_set, sugg, priv\rangle$ where:

- *pol_prec_set* is a possibly empty set of policy identifiers,[3] named policy precondition set, where all policies $p \in pol\_prec\_set$ are specified for the same target resource R;
- *sugg* is a pair **(list, op)**, where:
  $*$ *list* is a list of pairs of the form $(\mathcal{T}_w; \mathcal{T}_{i1}, .., \mathcal{T}_{il})$ where:
  $\diamond$ $\mathcal{T}_w$ is a term in $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n\}$;
  $\diamond$ $\mathcal{T}_{i1} ... \mathcal{T}_{il}$, $l \geq 1$, are the suggested terms for $\mathcal{T}_1$;
  $*$ *op* $\in \{+, -\}$ is a boolean value;
- *priv* is a privacy policy;

All the components of a context are optional.

The context of a policy is, thus, a set of information, to be used when processing the policy. Policy preconditions are used to protect sensitive policies by introducing an order in policy disclosure.

More precisely, the policy precondition set component is a set of policy identifiers, with the meaning that at least one of the policies in this set has to be satisfied by the counterpart before the policy conveying the context can be disclosed. More details can be found in Bertino et al. [2004b].

Example 7. With respect to the reference scenario introduced in Section 3, consider the release of project proposal reports. Since also the disclosure policies controlling project reports are sensitive, only WICE engineers working

[3] We assume that each policy is identified by a unique identifier.

on the project proposal are allowed to view them. As such, satisfaction of the requirement modeled through rule r_2 of Example 6 is a precondition for additional policies requiring more detailed data on the requesting chemical engineer and her involvement in the project. Thus, let p_3 be the policy identifier associated with rule Project_Proposal ←(Project_Badge(date_of_issue <Project_Proposal.date_of_issue, ActivityLine=W_U_Report.ActivityLine)). The *pol_prec_set* of p_3 will be of the form $\langle\{p_2\}, \ldots, \ldots\rangle$, where p_2 is the id of the policy containing rule r_2 of Example 6.

Policy conveying rule r_1 of Example 6 can instead be a precondition for the policy requiring additional information on certified skills of the applicant and personal information like the marital status and the residency. Let p_4 be the policy identifier corresponding to the rule JobApplication(...)←MaritalStatus(), IdCard(City, ZipCode), ChemistryAssCert(). The *pol_prec_set* field of p_4 will be of the form $\langle\{p_1\}, \ldots, \ldots\rangle$, where p_1 denotes the id of the policy containing rule r_1.

The *sugg* component is a new concept in our language; it has been introduced to facilitate the negotiation process. The motivation behind this component is that commonly used resources are nowadays regulated by standard off-the-shelf policies, which are public. When such types of resources are involved, the credentials of the ensuing negotiation rounds can be easily predicted.

To speed up the process, the negotiator can refer to these policies and directly list the name of the credentials it is willing to disclose next. The list of credential names is reported in the *list* field of *sugg* component.

When the policy is obvious (for instance, because of previous negotiations involving the same resources), its disclosure is not really required and one single round of negotiation can be enough. The *sugg* component is, therefore, a way to avoid computational resource waste in those cases in which the subsequent policy exchange would not add anything to privacy preservation. Finally, in the definition we defined *op*, an operator contained in the *sugg* component that has to be interpreted as follows. If op = "+" then the proposed terms are a mere suggestion to proceed in the process, otherwise, if op = "−," the suggested terms represent the only local terms the party can provide. Note that suggesting the next credentials (defined by the terms of the *sugg* component), the negotiator is willing to disclose with op = "−" avoids that the negotiator is asked for other credentials, which may be sensitive and not to be revealed in the negotiation. This is a way of preserving privacy of the negotiator specifying the policy, which explicitly claims what it is willing to show next for advancing the negotiation without having to reply to any other policy. The following example clarifies this concept.

Example 8. Rule r_1 of Example 6, states that, among the other credentials, a WICE_Card is required for submitting the application to a WICE consultant. Since, for any offered service, the WICE membership card is a firm requirement, the disclosure policies exchanged after such a request are known and predictable. WICE members typically only disclose their own cards either to other network members or to the institutions the WICE network cooperates with. WICE partners are recognized by the possession of a certified statement issued by the network referred to as "Wice_Partner." Recall that p_1 is the policy identifier of rule r_1. The *sugg* component of p_1 context will be $((\mathit{WICE\_Card};\ \mathit{WICE\_Card}, \mathit{WICE\_Partner}); +)$, where *WICE_Card and WICE_Partner* are terms denoting credentials. The suggestion says that the negotiator counter request for the WICE_Card is either the disclosure of the WICE_Card or, alternatively, the disclosure of the WICE_Partners card. "+" denotes that the counterpart may disclose other alternative credentials, if the listed ones cannot be disclosed.

For what concerns rule r_2, suppose that ChemistryAssociation is a credential known to be released to certified members of the "Chemical Engineers Association" (CeA). Since such credential is not sensitive and does not require to be protected, Alice can directly state CeA possession along with the rule, using the sugg component of p_2. As a result, the sugg component will be: ((ChemistryAssociation; CeA); -).

Finally, the *priv* component of a context denotes a P3P privacy policy. Such a policy complements the disclosure policies, specifying whether the information conveyed by the required credentials will be collected and/or used. Privacy policies may also specify the management of the portions of credential content (if any), not explicitly requested by the associated policy, but anyhow obtained as a result of the credential disclosure. Indeed, upon receiving a credential, unless protected attributes are used, the recipient obtains the entire credential, in addition, the attributes requested to satisfy the policy. We further discuss P3P policy encoding in Section 9.

We are now ready to formally define a disclosure policy.

Definition 5.5. (*Disclosure Policy*). A disclosure policy *dp* is a pair: **(rule, context)** where **rule** is a rule defined according to Definition 5.3 and **context** is the policy context defined according to Definition 5.4.

Example 9. In our job placement scenario through the help of the WICE network, the applicant is asked to provide her digital ID card and to provide other certificates testifying her working skills, along with the residency status. Suppose that *WICE* network maintains a database of members personal data. In particular, it collects information about the WICE members who are currently on the job market to promote events and inform them of possible job opportunities on the basis of the information gathered from members' interactions. Thus, a possible context for policy p_3 of Example 7 is: $\langle\{p_2\}$((WICE_Card; WICE_Card, WICE_Partners); +), priv$\rangle$, where *priv* is the P3P privacy policy informing a user about ID Card management (refer to Appendix B, Figure B.2 for an example of P3P policy for the ID Card).

We denote with the term *Policy Base* ($\mathcal{PB}$) the encoding of all the disclosure policies associated with a party.

6. PROTOCOLS AND STRATEGIES

Trust negotiations can be carried out according to several approaches, referred to as *strategies*. Prior work on trust negotiation associates with the term *strategies* the following meaning: "a strategy controls the exact content of messages, that is, which credentials to disclose, when to disclose them, and when to terminate the negotiation" [Yu et al. 2003]. Our use of the term strategy is quite similar to this definition. By trust negotiation strategies, we mean the approach a negotiating party can adopt in carrying on a negotiation, exploiting the tools provided by our policy language. The language extensions presented in Section 5, indeed, are the basis upon which a variety of novel trust negotiation strategies can be developed, with the goal of protecting privacy of the parties, according to their specific needs. In this section, before focusing on PP-Trust-$\mathcal{X}$ negotiation strategies, we introduce the notion of negotiation tree, a data structure that keeps track of the advances in the policy evaluation phase.

6.1 Negotiation Tree

A negotiation tree [Bertino et al. 2004b] is rooted at a node labeled with the identifier of the requested resource and it is initialized when the policy evaluation phase of a negotiation starts. It is dynamically built and grows as the policy evaluation phase proceeds. Formally, a negotiation tree *NT* is a labeled tree in which each node corresponds to a term, and edges correspond to policy rules. A negotiation tree is a tuple $NT = \langle \mathcal{N}, \mathcal{R}, \mathcal{E}, \phi \rangle$, where $\mathcal{N}$ denotes the set of nodes, $\mathcal{R}$ denotes the root of the tree, and $\mathcal{E}$ and ϕ correspond to the set of edges and the labeling function, respectively. Function ϕ associates a label with each edge in the tree, in order to give information on the order according to which the credentials associated with the corresponding nodes must be disclosed. The order is implied by the policy precondition set of each rule. Set $\mathcal{E}$ contains two different kinds of edges: *simple* and *multi*. A simple edge is used to model policies having only one term on the left-side component of the associated rule. By contrast, a multi edge links several simple edges in order to represent policy rules having more than one term on their left-side component. Nodes belonging to a multi edge are thus considered as a whole during the negotiation. Each node[4] of a negotiation tree can assume four different states: *deliv* state denoting a delivery resource, that is, a credential/resource ready to be disclosed without further requirements; *open* state, meaning that the credential/resource denoted by the node is not yet ready to be delivered. The tree nodes can also assume state *cred_proof_disclosed* and *attr_discl*, denoting, respectively, the disclosure of the credential proof referring to the credential referred by the term stored in the node; and the disclosed attributes of the credential, as required by the corresponding term associated with the node. *cred_proof_disclosed* and *attr_discl* states are used according the strategy adopted by the party. We elaborate on these aspects in the next section.

The policy evaluation phase is successfully completed if the tree contains a subtree rooted at R having all nodes with a *deliv*/*cred_proof_disclosed*/*attr_discl* state, depending on the strategies adopted by the parties. We refer to such a subtree as *valid view*. When the negotiation tree includes a valid view, it is possible to determine a trust sequence for the considered negotiation. The

[4] Given a node $n \in \mathcal{N}$, we use the notation $\mathcal{T}(n)$ to denote the term $\mathcal{T}$; *state*(n) to denote the state of $\mathcal{T}$; and *party*(n) to denote the owner of the term in n.

Table I. PP-Trust-$\mathcal{X}$ Actions and Possible Replies

Action	Meaning	Positive Reply	Negative Reply
$Cred\_Proof\_req(\mathcal{T})$, where $\mathcal{T} = P(c)$	to request cred. proof	$Cred\_proof\_send(cred)$ *cred*, is a cred. of type P satisfying c	$Cred\_proof\_refuse(\mathcal{T})$
Send(p), $p.sugg \neq \emptyset$	to send policy with *sugg*	$Accept(p.sugg)$	$Refuse(p.sugg)$
Send(p), $priv \neq \emptyset$	to send policy with privacy policy *priv*	$Accept(p.priv)$	$Refuse(p.priv)$
Attr_req(attr)	to request attribute	$Attr\_send(C.attr)$ where C is a cred. containing attribute *attr*	$Attr\_refuse(attr)$

trust sequence can be built by traversing the view according to a specified order defined by the labeling function associated with the tree.

6.2 Strategies

Parties build the negotiation tree by following a protocol, which consists of an alternate transmission of messages containing a sequence of update operations and a set of policies to be used to proceed in the phase. The use of the policy context, as well as the possibility of partially disclosing a credential, results in a number of different options for carrying on the policy evaluation phase. As a result, the legal messages that may be exchanged may convey information of different types and nature. More precisely, the messages that can be transmitted are not only limited to mere policy exchanges, but can also be related to credential proof, or privacy policy exchanges. We distinguish between two kinds of messages referred to as *actions* and *replies*. The former kind concerns messages for enquiries about resources. Table I shows possible actions provided by our protocols and corresponding replies.

Sending a disclosure policy entails a variable number of messages as a reply, which depend on the associated components of the policy context. For instance, in addition to the basic reply, that is, sending a counterpolicy, the negotiator may notify acceptance or refusal of the specified suggestion, if present. Similarly, if a rule is accompanied by a privacy policy, a reply specifying either its acceptance or refusal is required. As shown in Table I, each action has a corresponding set of replies. The function in charge of selecting valid replies among all possible ones is presented in Appendix A. When more than one option is possible, the selected valid replies are chosen on the basis of the negotiator local setting and also on the type of negotiation. In fact, in order to define a framework that is adaptable and flexible, we do not define a unique strategy for building the tree and, thus, for carrying on negotiations.

Our framework supports four basic general-purpose strategies that reflect four different approaches to a negotiation. We illustrate them in the remainder of the section. In addition, the system provides a mixed strategy presented in Section 6.2.5, which results from the combination of the basic ones.

6.2.1 *Standard Strategy.* This is the traditional strategy for carrying on a negotiation, based on an informed strategy, where policies are exchanged at first, and only the credentials relevant for successfully completing the negotiation are subsequently disclosed, if a trust sequence is found.

6.2.2 *Suspicious Strategy.* Under this strategy the credential proof is always requested during the policy evaluation phase for each of the involved credentials. More precisely, the *cred_proof_disclosed* of a credential is requested when the corresponding policy is satisfied, that is, as the corresponding node becomes a delivery node. Thus, the *cred_proof_disclosed* state for each involved credential is mandatory in order to successfully complete the policy evaluation phase. As a result, this phase ends when a valid view of the tree is found, where the state of each node of the view is *cred_proof_disclosed*. Finally, when the credentials content is disclosed, the protected attributes not required by the process may be kept secret.

6.2.3 *Strongly Suspicious Strategy.* This strategy is a specific case of the suspicious one. Under this scheme, parties require attribute disclosures as soon as the corresponding policies are satisfied (i.e., the state of the node storing the related term is *deliv*). Partial credential disclosure is indicated in the tree by the *attr_discl* state of the corresponding node. As result, the process ends when a valid view of the tree is found, where each node of the view has the *attr_discl* state. This approach, although more onerous than the suspicious one, is more effective, since it eliminates the need of the credential exchange phase[5] and results in a greater control over private information.

6.2.4 *Trusting Strategy.* The goal of this strategy is to speed up the process whenever possible through credential suggestions, stored in the *sugg* component of a policy context. The main advantage of this strategy is that if one of the parties uses the suggestions for all the involved policies, the number of negotiation rounds is reduced by one-half with respect to a standard negotiation.

By suggesting the next step, a party can drive the counterpart reply even if many other alternatives are possible. As a result, if iterated for a significant number of times, the suggesting party can actually drive the negotiation toward a precise solution, thus limiting counterpart freedom. However, this is the ideal approach in scenarios characterized by counterparts with limited resources, such as a mobile user. A subject can choose to speed up the negotiation process by simply accepting suggestions, thus alleviating its task of retrieving and evaluating policy compliance at each round. In addition, suggestions can be useful in all the cases in which there is only one (or few) standard ways of carrying on the negotiations. In this case, suggestions can simply codify such a standard way. The next theorem proves the correctness of these four strategies.

THEOREM 6.1. *Let $\mathcal{CN}$ be a resource controller and $\mathcal{RQ}$ be a requester. Let $\mathcal{PB}_{cn}$ and $\mathcal{PB}_{rq}$ be the policy bases associated with $\mathcal{CN}$ and $\mathcal{RQ}$, respectively. Let*

[5] Recall that, as explained in Section 1.1, a negotiation usually ends by disclosing parties credentials according to the order determined in the policy evaluation phase.

X-$Prof_{cn}$ and X-$Prof_{rq}$ be the $\mathcal{X}$-Profiles associated with $\mathcal{CN}$ and $\mathcal{RQ}$, respectively. Let R be the resource requested by $\mathcal{RQ}$ to $\mathcal{CN}$. Let NT be a negotiation tree for the resource R and let $\mathcal{WT}$ be a view on NT determined using a strategy in {standard, trusting, suspicious, strongly suspicious}. $\mathcal{WT}$ is a valid view, that is, there is a corresponding trust sequence TS containing all and only the credentials corresponding to terms in the view, ending with disclosure of R.

Formal proof is reported in Appendix C.

6.2.5 *Strategy Interoperability.* As recognized by many researchers [Yu et al. 2003], in order to be effective, negotiation strategies should interoperate, in the sense that negotiators should be free of using different strategies and combine them, when needed. To support strategy interoperability, PP-Trust-$\mathcal{X}$ supports a mixed strategy in addition to the four basic strategies presented in previous section. This strategy is characterized by the possibility of dynamically switching among the PP-Trust-$\mathcal{X}$ basic strategies. The motivation of a mixed strategy is intuitive. Trust is not static and changes over time, even during a single negotiation. The type and content of the messages exchanged during trust negotiations should, therefore, be adapted to the level of trust reached at each round, by switching among strategies, if needed. Further, parties should not be forced to stick to a fixed strategy during the entire negotiation process, but they should be able to dynamically change the adopted strategy, as the negotiation proceeds. Finally, it is likely that for negotiations involving a considerable number of heterogeneous resources, the adoption of a single strategy is too restrictive. As the policy evaluation phase evolves and new policies are exchanged new protection requirements may arise and thus other strategies than the one selected may be better suited.

Example 10. In our running example, Alice at Western University and Helen at Milano University may start negotiating with a suspicious strategy. After the disclosure policy requiring proofs of the involvement in the project is given and related credential proofs disclosed, Alice may choose to proceed with a standard strategy, as a sufficient level of trust on the counterpart is reached.

As a further example, during on-line purchases, when preliminary information about customers' identity and preferences are exchanged, a trusting strategy may be adopted, whereas when payment information is negotiated, a stricter control over the counterpart policies is to be preferred; thus, a suspicious strategy can be used.

Correctness of the mixed strategy is ensured by the following theorem.

THEOREM 6.2. *Let $\mathcal{CN}$ be a resource controller and $\mathcal{RQ}$ be a requester. Let $\mathcal{PB}_{cn}$ and $\mathcal{PB}_{rq}$ be the policy bases associated with $\mathcal{CN}$ and $\mathcal{RQ}$, respectively. Let X-$Prof_{cn}$ and X-$Prof_{rq}$ be the $\mathcal{X}$-Profiles associated with $\mathcal{CN}$ and $\mathcal{RQ}$, respectively. Let R be the resource requested by $\mathcal{RQ}$ to $\mathcal{CN}$. Let NT be a negotiation tree for R and let $\mathcal{WT}$ be a view on NT determined using the mixed strategy. $\mathcal{WT}$ is a valid view, that is, there is a corresponding trust sequence TS containing all and only the credentials corresponding to terms in the view ending with disclosure of R.*

The proof is reported in Appendix C.

Not all the proposed strategies may be adequate in all possible negotiation scenarios that may occur, although technically possible. One of the two parties may have specific requirements on the counterpart's adopted strategy. In the following, we refer to a party's requirements over selected strategies as party *local settings*.

In our reference scenario, Helen may want to negotiate with researcher Alice under either a suspicious or a strongly suspicious strategy. As a further example, clients negotiating with well-known servers may accept that servers adopt a trusting strategy, whereas a server may adopt an internal regulation stating that only entities adopting either a suspicious or a standard strategy may be accepted for negotiation.

Parties may reach an agreement about the strategies to use either by directly communicating the allowed strategies or by autonomously selecting them while the negotiation is being executed. The strategy to be adopted is usually selected before entering the process on the basis of the estimated sensitivity and type of credentials to be negotiated. Once an (implicit or explicit) agreement on the adopted strategies is reached, parties may also accept to relax the constraints provided by the selected strategy under specific circumstances. For instance, the suggestions associated with policies cannot be used under a suspicious or a strongly suspicious strategy. However, to speed up the evaluation, if a policy, containing a nonempty *sugg* component, is received and *sugg.list* corresponds to the local policies for the requested credentials, the suggestion should anyway be accepted.

Example 11. With respect to our reference scenario, suppose that a negotiation between Helen and Alice concerning project proposals is executed. Suppose Alice at Western University adopts a trusting strategy, while only a suspicious strategy is accepted by the counterpart. Suppose Helen's trust negotiation local settings let her adopt either a suspicious strategy or a standard one and accept a trusting strategy by the counterpart. Suppose that disclosure policy: $\langle r_1, <\{\}, ((\mathit{WICE\_Card};\ \mathit{WICE\_Card}, \mathit{WICE\_Partners}); +), \{\} > \rangle$ is sent to Helen, where r_1 is the rule of Example 6. If the suggestion of a *WICE_Card* credential is compliant with Mary's local policies, her next message contains the following valid replies: $\{\mathit{Accept}(\mathit{WICE\_Card}), \mathit{Cred\_proof\_send}(\mathit{WICE\_Card})\}$.

The following policy sent by Alice is policy p_3 of Example 7, the related context of which contains the privacy policy priv. Again, Helen evaluates such a policy and matches the associated privacy policy with her own privacy preferences. Her reply may convey an action message requiring a counter proof of Alices involvement in the *WICE*. Helen may ask for a credential proof of such a credential, as it is the cornerstone for proceeding with the negotiation. As such, parties can keep negotiating by exchanging messages conveying actions related to both trusting and suspicious strategies.

Ideally, a negotiation system should always assure the parties that all the *trust requirements* they have with respect to the negotiation currently being carried on are met. The notion of trust requirements in PP-Trust-$\mathcal{X}$ is not only related to protection of resources (e.g., credentials, services, and policies), as typically intended in prior trust negotiation work [Bonatti and Samarati 2000; Seamons et al. 2001; Winsborough and Li 2002a]. Trust requirements in our framework also includes entities privacy policy satisfaction and parties adopted strategies agreement. More precisely, satisfaction of trust requirements for a PP-Trust-$\mathcal{X}$ negotiation can be defined as follows. In the definition, we use the concept of *X-Profile*, which, as introduced earlier, corresponds to the profile of credentials possessed by a negotiation party.

Definition 6.1. (*Satisfied Trust Requirement*). Let ξ be a PP-Trust-$\mathcal{X}$ entity. Let PB_ξ and $X\text{-}Prof_\xi$ be the policy base and the $\mathcal{X}$-Profile associated with ξ, respectively. Let *NT* be a negotiation involving ξ. *NT* satisfies the trust requirements of ξ if the following conditions hold:

1. Each credential *cred* requested by the counterpart during *NT* is disclosed if, and only if, one of the following conditions hold:
   (a) *cred* is disclosable;
   (b) At least one of *cred*'s associated disclosure policies is satisfied before its disclosure.
2. Each time *Privacy_matching*()[6] function is invoked, it returns an *Accept*(*priv*) reply action.
3. The strategies adopted by the counterpart are compliant with ξ's local settings.

Note that in the above definition we do not specify whether ξ is acting as a resource controller or resource requester, since, in PP-Trust-$\mathcal{X}$, both parties protection of trust requisites are equally important. Further, since PP-Trust-$\mathcal{X}$ is a privacy aware system, asking for disclosure policy satisfaction is not sufficient for the negotiation success. Thus, condition (2) requires that privacy policies of the counterpart are accepted, that is, all received remote privacy policies are compatible with local privacy preferences. Finally, condition (3) is related to the possibility of selecting among different negotiation strategies and it requires ξ's selected strategy to be acceptable by the counterpart.

Another desirable property of negotiations, besides correctness, concerns completeness. By completeness we mean that, if the negotiating parties adhere to the PP-Trust-$\mathcal{X}$ protocol and their trust requirements are compatible, the negotiation always succeeds. The following theorem states the completeness of PP-Trust-$\mathcal{X}$ negotiations. In the theorem, we use the notion of party's $\mathcal{X}$-Profile, which correspond to the collection of credentials associated with a negotiation party (see Section 1.1 and Bertino et al. [2004b] for more details).

THEOREM 6.3. *Let $\mathcal{CN}$ be a resource controller and $\mathcal{RQ}$ be a requester. Let $\mathcal{PB}_{cn}$ and $\mathcal{PB}_{rq}$ be the policy bases of $\mathcal{CN}$ and $\mathcal{RQ}$, respectively. Let $X\text{-}Prof_{cn}$ and $X\text{-}Prof_{rq}$ be the $\mathcal{X}$-Profiles associated with $\mathcal{CN}$ and $\mathcal{RQ}$, respectively. Let R be the resource requested by $\mathcal{RQ}$ to $\mathcal{CN}$. If trust requirements for R by $\mathcal{RQ}$ and $\mathcal{CN}$ are satisfied (as by Definition 6.1) according to the associated $\mathcal{PB}_{cn}$, $\mathcal{PB}_{rq}$,*

[6] Privacy_matching is the function in charge of matching remote privacy policies with local privacy preferences, and vice versa. We report the function code in Appendix A for lack of space.

Table II. PP-Trust-$\mathcal{X}$ Strategies and Domain Applicability

Privacy Req.	Standard	Suspicious	Strongly Susp.	Trusting
Low	B2C Medical Research			B2B Joint coop
Medium		B2B B2C Joint coop		Medical Research
High	Research	Joint coop	B2C Medical B2B	

X-$Prof_{cn}$, and X-$Prof_{rq}$, then the negotiation succeeds, that is, a trust sequence TS ending with the disclosure of R is found.

Formal proof is reported in Appendix C.

7. APPLICABILITY OF TRUST NEGOTIATION STRATEGIES

The availability of a number of negotiation strategies empowers subjects to determine how, when, and to what extent information is communicated to the other party during trust negotiation. As illustrated in the previous section, the proposed strategies differ in the type of messages to be exchanged and achieve different levels of privacy protection. The applicability of a specific strategy depends on the profile of the negotiating parties and on their specific privacy needs. To show how they can be usefully adopted in the different scenarios that may occur, we have compared and evaluated PP-Trust-$\mathcal{X}$ strategies along two different dimensions: the negotiation domain and the computational resources of the negotiating parties. For the first dimension, we have identified the best strategy with respect to the required level of privacy. In Table II we have classified negotiations on the basis of the main domains in which they occur. The possible domains where negotiations are likely to be executed are related to e-commerce transactions, exchange of medical (or other sensitive)-related data, research environments, and joint collaborations among different enterprises (abbreviated as "joint coop," in the table). As far as e-commerce is concerned, we have further distinguished between business-to-business (B2B) and business-to-consumer (B2C) transactions, since they are usually characterized by different privacy requirements. Example of trust negotiations in each of the above domains are: on line shopping from web sites for B2C transactions, supply of basic commodities among factories for B2B applications, exchange of medical records about patients among collaborating clinics, and sharing of research reports among researchers and/or students.

In current B2B transactions, enterprises need to execute complex operations to purchase supplies in a timely and privacy-preserving fashion, to avoid competing sellers to learn the adopted business strategies. The use of credentials proving partnerships, ensuring the liability of the dealers and other properties of the negotiating entities, can make business transactions more effective and help both buyers and sellers in adopting customized strategies based on the counterpart properties. In addition, the gradual and controlled disclosure of digitally signed credentials provided by the negotiation strategies increase the security of the interactions. As such, this type of business can benefit from trust negotiations carried out using systems like PP-Trust-$\mathcal{X}$. Negotiations carried out in the framework of joint collaborations usually concern heterogeneous organizations pooling together data and computational resources for a common purpose, as in our running example. Here, by using PP-Trust-$\mathcal{X}$, we can imagine competing companies collaborating for classified projects in a private manner, based on the trust established on line via trust negotiations.

An important domain of applications for PP-Trust-$\mathcal{X}$ is also represented by the health care domain. Health industry payers and providers collect and maintain large volumes of protected health information. Furthermore, they collect other sensitive personal and financial data and conduct many transactions electronically. Several security and privacy issues arise in this context. Specifically, an issue to be addressed is related to the individual's identity, which, in this specific environment, includes his/her medical history, and is made up of disjoint medical records from different health institutions. Protecting this information is compelling, since the mere knowledge of the existence of the medical information may leak valuable information about an individual.

Nowadays these kind of transactions are very limited and rely on preestablished trust and agreements among parties. Interactions are very limited and often slowed by the lack of adequate technologies supporting transactions of this specific type. In such a context, privacy-preserving trust negotiations could be of great advantage, as they help in protecting access to sensitive medical information related to the individual and do not require any static trust between parties. In addition, by using credentials, data provenance and integrity are also ensured, reducing the risk of false claims or forgery.

For classifying privacy requirements, we distinguish among a low, medium, and high level. A subject that does not have particular concerns in revealing his/her personal information, has *low* privacy requirements. A subject has *medium* privacy requirements if the disclosure of private information is not prevented, but the subject is very careful in revealing his/her credentials and he/she adopts ad hoc approaches to detect counterpart's misbehavior. Finally, a *high* level corresponds to the behavior of subjects considering privacy of personal data as a first-class requirement in a trust negotiation. Such privacy-conscious subjects will adopt and ask the counterpart to enforce the "minimum disclosure" principle, which implies that information disclosure is to be limited to the data actually required for succeeding in the negotiation.

Of course the actual level of privacy to be reached is tightly coupled with the specific purpose of the negotiation and the privacy preferences of the parties. For instance, privacy requirements of a business-to-consumer transaction for purchasing food at an on-line grocery shop are very different from those of B2C transactions carried on for buying medicine and drugs.

As shown in Table II, suspicious and strongly suspicious strategies are, in general, preferable when privacy requirements are stringent (e.g., set to high). In contrast, when privacy requirements are low, a standard strategy is adequate, as this strategy effectively protects disclosure of sensitive information while at the same time, maximizes the negotiation success rate (see [Bertino et al. 2004b]). A trusting strategy is also effective, especially in B2C negotiations in which one of the parties is usually a known and trustworthy server. In the framework of joint cooperation, since the parties are not usually strangers, they can also take advantage of a trusting strategy to speed up and facilitate the negotiation process. Finally, in case of medium privacy requirements, any strategy can be adopted in any context. However, we suggest a suspicious strategy to be adopted when the parties are strangers, as it often happens in B2B and B2C transactions. Our choice is motivated by the vulnerable nature of open environments like the Internet, where such negotiations take place. In collaborating environments—like research or educational organizations—a trusting strategy can, instead, be successfully adopted, as parties are usually not totally unknown to one another, but they always have an enterprise or organization to which they can be referred to (e.g., the hospital or university which they are affiliated with or which they represent).

It has to be noted, however, that the mixed strategy because of its adaptive nature is suitable for any of the possible scenarios and domains.[7] For instance, a mixed strategy is well suited for trust negotiations when accessing research data. This is because these types of negotiations often have conflicting requirements: they should always succeed whenever possible, because of the need for sharing research data as much as possible. However, very often, interacting parties are not willing to reveal their personal credentials for this purpose and thus may have stringent privacy requirements, which may compromise the success of the process.

Another dimension we have considered in analyzing our strategies is related to the computational resources of the interacting parties. Trust negotiations can be quite expensive in terms of computational and communication resources. Parties with limited resources are usually more interested in fast negotiations. For wireless clients, it is not practical to perform all the phases required to complete a trust negotiation, such as the storage of all required credentials, the processing steps that include costly cryptographic verifications, and the network communications with the other negotiation participants. With PP-Trust-$\mathcal{X}$ it is possible to reduce the overhead in certain circumstances. For instance, the adoption of ad hoc strategies, such as the strongly suspicious one, helps in reducing the information to be exchanged and limits the number of rounds. If both the parties adopt this strategy, the credential exchange phase is not required and the number of message exchanges is considerably reduced. For negotiations where one of the parties is connected through a mobile device, a trusting strategy can be more suitable. Mobile devices are often poor in memory capacity and can only use slow connections. In such a case, the negotiating party, connected through a desktop computer or a server can drive the negotiation process, by adopting a trusting strategy. As remarked earlier in the paper, a trusting strategy is cost-effective, if the stronger negotiation party (or driver) is able to prove its trustworthiness, such as a company supplying services in a B2C transaction. Under this scheme, once the driver trustworthiness is ensured, the driver can keep track of the progress of the negotiation, freeing the mobile party from carrying this burden.

[7] As such, we chose not to report it in Table II.

8. ATTACK ANALYSIS

In this section, we analyze the potential attacks that the PP-Trust-$\mathcal{X}$ system might be subjected to and discuss possible countermeasures that may be adopted. We can identify several types of attacks that the PP-Trust-$\mathcal{X}$ system might need to thwart. Some attacks are similar to those that can be performed against any server available on the web, like denial-of-services (DoS), timing and replay attacks, and man-in-the-middle attacks. These attacks usually alter the normal behavior of the system and make it unable to work normally. DoS attacks may be launched at the application layer by initiating a large number of trust negotiations with a same party. The attacked system is flooded either with complex disclosure policies to be simultaneously evaluated or with a number of revoked or unneeded credentials to verify and validate. As a result, the system slows down or becomes unavailable to honest negotiation subjects. To prevent and mitigate the effect of DoS attacks, some ideas can be borrowed from conventional on-line systems and application. For example, one possible approach is to fix an upper bound to the number of negotiations that each PP-Trust-$\mathcal{X}$ system can simultaneously handle. Upper bounds can also be applied to each single negotiation. A party can establish *a priori* the maximum duration of a negotiation, in terms of processed policies and/or credentials to be evaluated, and tighten negotiation timeout. Further, an intrusion-detection system (IDS) can be employed, to estimate the risk of an attack while negotiations are being executed. In addition, negotiation protocols are secure against man-in-the-middle attacks, since parties are connected using SSL sockets and thus messages are always ciphered. Disclosure policies may be digitally signed in order to prevent accidental or malicious modification of their content. Therefore, each policy receiver will be able to detect any modification of the policy, as long as the policy sender's private key used to sign the policy has not been modified. Timing and replay attacks may be prevented with the help of timestamps to be included in the exchanged messages and random challenges sent by the parties. An attacker in these cases may not be able to retain the freshness of the messages, since they are encrypted and, therefore, it cannot replay them to the recipients. Other kinds of attack are related to stealing of personal information. An attacker may carry on negotiations with the goal of inferring protected information and not for establishing mutual trust. For instance, the attacker might try to use ad hoc disclosure policies to gather personal information to be reused later on in different negotiations. Our suspicious and strongly suspicious strategies mitigate the risk of those attacks in many ways. By forcing the disclosure of the credential header as soon as a credential is involved (see Section 6.2) in the negotiation, a party cannot declare false possession of a credential, as it has to give immediate proof of its ownership. As a result, the possibility of keeping the negotiation open is constrained by the actual profile of credentials and its compatibility with the counterpart disclosure policies. Further, information leakage is reduced by blinding the nonrequired attributes with multihash technique, thus vanishing the attempt of a negotiating party of collecting unneeded counterpart information. Of course, the attacker can still collect sensitive information by sending disclosure policies having numerous

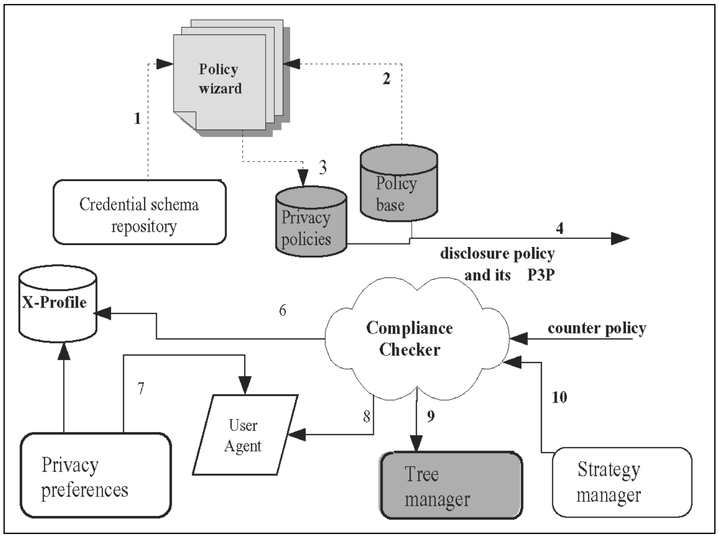

Fig. 5. Trust-$\mathcal{X}$ framework. Dashed lines denote off-line operations. Arrow labels denote the flow of the operations.

credential terms and then failing the negotiation once the interested information is inferred. To limit this risk, a party may also take advantage of the *sugg* field in the policy context. Informing in advance what the party is willing to disclose frees him/her from responding to any other disclosure policy. Also, we need to consider the scenario of a malicious party launching a *property theft* attack, a variant of the common *identity theft* attack that could easily exploit trust negotiation systems. An attacker may use policies and credentials not belonging to him/her, eavesdropped or stolen from a third party. This attack can be prevented by checking credential ownership and using credential chains to retrieve information about the owner. Also, an effective way to mitigate the risk of this attack is to use short-term credentials. In this case, even if a credential is stolen, it cannot be reused for many negotiations before its expiration. Finally, it is worthy to note that disclosure policies requesting for combination of credentials from different issuers might help in detecting stolen credentials. While it might happen that a credential is stolen, it is very unlikely that an attacker is able to steal the entire set of credentials a user has. An attacker may have to compromise several public/private keys before being able to successfully forge multiple credentials and gain illicit access.

9. A PRIVACY-ENABLED TRUST-$\mathcal{X}$ ARCHITECTURE

The Trust-$\mathcal{X}$ architecture is composed of several components, sketched in Figure 5. As illustrated, the main components of the system are a *policy base*, storing disclosure policies, the *$\mathcal{X}$-Profile* associated with the party, a *tree manager*, managing the negotiation tree. The system also includes a *compliance checker*, testing policy satisfaction and generating request replies, and a *strategy manager*, in charge of dynamically selecting the negotiation strategy and managing the messages exchanged during negotiations, according to the adopted remote and local strategies. The compliance checker also checks local policy satisfaction and verifies, at runtime, the validity and ownership of remote credentials. The goals of the system components are essentially to support policy and credential exchange and to test whether a policy is satisfied. In addition to the above elements, a set of modules for the management of privacy policies is included.

Current implementation of privacy systems [World Wide Web Consortium a] usually have two components deploying P3P. Web sites install privacy policies and reference files, at their sites, using various tools [IBM; JRC 2002]. Then, as users browse sites, their preferences are checked against site policies. This simple scheme is not adequate in our context, in that we are dealing with both user and server sides. Indeed, each Trust-$\mathcal{X}$ entity acts, during a negotiation, as a server requesting personal data as well as a user disclosing personal information. The framework should thus support both sides during negotiations. In what follows, we illustrate users and server modules separately. We then merge these two into a unified framework and show how they fit together. In presenting the modules, we mainly focus on P3P policies to be exchanged during policy evaluation phase. We recall that P3P policies are also exchanged during the policy agreement phase. However, such policies are coarse-grained, because they refer to the entire negotiation and not to a specific credential. As such, they can be specified and evaluated using standard mechanisms adopted by web sites.

For a policy sender, the system should support the following actions:

1. mapping credential types onto data schema usable for privacy policy specification;
2. specifying privacy policies about these credentials using P3P;
3. supporting shipping and evaluation of privacy policies during negotiations.

P3P policies can be specified off-line before negotiations start. The process is sketched at the top of Figure 5 (dashed lines). As shown, the module in charge of encoding P3P policy is the `policy wizard`. Given a disclosure policy *dp*, the module extracts the corresponding credential schema[8] (see Figure 5, arrow 1) required by *dp* from the `credential schema repository`. This module is implemented as a credential chain tool, to retrieve credential schemes from public issuer repositories and a local cache storing the most widely used schemes.

Upon retrieval of the required schemes, the credentials contents are analyzed in order to identify data to be collected. Credentials content can be analyzed under two different perspectives. If the information to be collected is a set of properties and the credential actually represents only the envelope to transmit these data, then the policy can be specified as a conventional P3P policy, that is, using built-in data schemes and categories provided by the standard, without referring to the particular credential collecting the requested attributes. In contrast, if the key information is the credential itself, then the policy should

[8] We use the term schema and type interchangeably, in this context.

refer not only to the attributes in the credential, but also to the credential itself. For instance, if a web server wants to cache an entire credential to create a database collecting customer's data, it has to refer to the specific credential, specifying its ID and issuer public key. In such cases, it is mandatory to extend P3P data schema to encode the data structure underlying the credential. Privacy policies are encoded according to version 1.1. of P3P [Cranor et al. 2003] that provides a new format for expressing P3P data schema in a simpler way than the previous one. The new format uses the XML schema definition (XSD) format, which can be validated against an XML schema. Since Trust-$\mathcal{X}$ credentials are defined in terms of DTDs, it is possible to encode the policy directly referring to the schema corresponding to a credential, by simply translating the DTD with XSLT [Clark 1999] in XSD.

An example of P3P policy referring to a credential schema is shown in Figure B.2.

Once the corresponding data schema has been encoded, the policy creation wizard can complete policy encoding, specifying how data is to be managed (for which purpose the data will be collected, for how long, and so forth), according to the local privacy practice. The privacy policy is finally linked to the corresponding disclosure policy.

On the other hand, the policy receiver must be equipped with tools for describing privacy preferences and matching policies against privacy preferences. Tools for describing privacy preferences can be implemented as ad hoc policy editors, able to encode user preferences into a set of rules to be used to evaluate remote P3P policies. Policy matching, by contrast, can be executed by an agent integrating the compliance checker module. Such an agent performs the task modeled by function *Privacy_matching*(), reported in Figure A.2, in Appendix A. As shown, policy evaluation is executed by checking the received local preference rules against remote privacy policies referring to the credentials requested by the disclosure policy. If no privacy policy is associated with the disclosure policy, then the privacy policy is checked against the preferences rules exchanged during the privacy agreement phase, denoted as *Priv_nego*, in Figure A.2. Similarly, if no privacy policy is associated with the disclosure policy, but a preference rule has been specified for the credentials requested by the policy, the preference rule is checked against the coarse-grained privacy policy exchanged during the privacy agreement phase.

Modules characterizing the two sides are merged into a unified framework and are complementary to each other. The first set of modules is used to integrate conventional disclosure policies sent to the counterpart with specific privacy policies. The latter, in contrast, acts as a further filter when remote policies requesting credentials conveying personal information are received.

As new features are added to Trust-$\mathcal{X}$, an increasing computational effort is required to carry out negotiations. For example, if a disclosure policy is sent together with a privacy policy, the compliance checker has actually to make one or two additional operations (arrow 7 in Figure 5) in order to check if the P3P policy applies. However, because of the simplicity underlying the P3P platform, the time required to perform this kind of checking is minimal and it does not really impact system performance. Moreover, it is expected that in most cases the overhead resulting from the P3P policy exchange will actually be very limited and confined to the privacy agreement phase. Indeed, during the subsequent phase, only additional privacy policies not covered by the previous ones may be exchanged, if required by the parties. We analyze system performance of the PP-Trust-$\mathcal{X}$ prototype system implemented in the following section.

10. SYSTEM IMPLEMENTATION AND EXPERIMENTAL RESULTS

In this section we first describe the implementation of the PP-Trust-$\mathcal{X}$ prototype system. Then, we describe the performance results we have obtained from the prototype.

10.1 The PP-Trust-$\mathcal{X}$ Prototype

The system main component is a Web service supporting the operations to carry on a trust negotiation according to the different negotiation strategies presented in this paper and of a client application that invokes the Web service operations. The PP-Trust-$\mathcal{X}$ Web service has been developed in Java, using the Tomcat Application Server and the Axis Soap Engine. The client application has also been implemented using Java. The Oracle database version 10g has been used to store disclosure policies and credentials necessary to carry on a trust negotiation. The integration with P3P platform is still under development. As such, we do not elaborate on it in this section.

The PP-Trust-$\mathcal{X}$ Web service provides three different operations, `StartNegotiation`, `PolicyExchange`, and `CredentialExchange` corresponding to the main phases of the negotiation process.

`StartNegotiation` has, as input message, `StartNegotiationRequest`, that specifies the negotiation strategy selected by the user who invokes the operation, the URL of the counter party in the negotiation process, and the parameters to connect to the Oracle database containing the disclosure policies and credentials of the invoker. `StartNegotiation` assigns a unique id to the negotiation process and opens the connection with the Oracle database. The negotiation id is returned by the output message `StartNegotiationResponse`. `PolicyExchange` checks if the database contains disclosure policies protecting the credentials requested in the counterpart's disclosure policies, which are listed in the message `PolicyExchangeRequest`. If this is the case, they are inserted in the response message `PolicyExchangeResponse`. `CredentialExchange` receives as input the message `CredentialExchangeRequest` and verifies the validity of the counterpart's credential contained in the message. It then selects the next credential to be sent to the negotiation partner. The selected credential is returned in the message `CredentialExchangeResponse`.

The client application is equipped with a GUI, by means of which a user specifies the parameters of the negotiation. The user can also monitor the negotiation process, checking the exchanged disclosure policies and credentials. Figure 6 illustrates the interactions between the client applications run by two negotiating parties and the PP-Trust-$\mathcal{X}$ Web service.

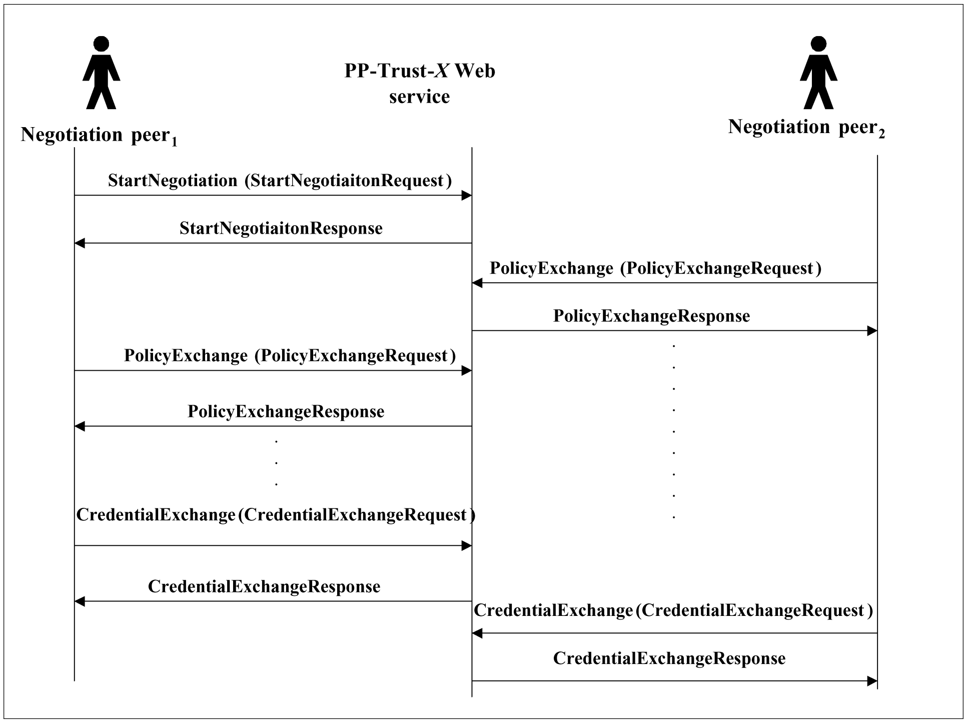
PP-Trust-X Web service
Negotiation peer$_1$
Negotiation peer$_2$
StartNegotiation (StartNegotiaitonRequest)
StartNegotiaitonResponse
PolicyExchange (PolicyExchangeRequest)
PolicyExchangeResponse
PolicyExchange (PolicyExchangeRequest)
PolicyExchangeResponse
CredentialExchange (CredentialExchangeRequest)
CredentialExchangeResponse
CredentialExchange (CredentialExchangeRequest)
CredentialExchangeResponse

Fig. 6. Interactions among PP-Trust-$\mathcal{X}$ components.

10.2 Performance Evaluation

We have carried out an extensive performance evaluation study to estimate the performance and scalability of the negotiation strategies supported by PP-Trust-$\mathcal{X}$. In what follows, we first describe the test cases we have used to analyze the performance of the negotiation strategies. We then present and discuss the most significant results obtained from the experiments.

10.2.1 *Test Cases.* We analyze the scalability of the negotiation strategies supported by the implemented prototype according to two parameters: the complexity of the disclosure policies and the number of negotiation rounds. By complexity of a policy, we mean the number of terms[9] that are required to the counterpart for a single resource. We distinguish between single- and multiple-term policies. Policy complexity influences the number and the length of alternative trust sequences according to which the two parties can complete the negotiation. The number of negotiation rounds is captured by the height of the negotiation tree: the greater the number of levels in the negotiation tree is, the longer the alternative trust sequences are. To evaluate the impact of the complexity of disclosure policies, we have carried out experiments on a balanced tree of forty nodes, arranged in five levels. We developed thirteen test cases in which

[9] We recall that, according to the definition of negotiation tree, a term corresponds to a credential request.

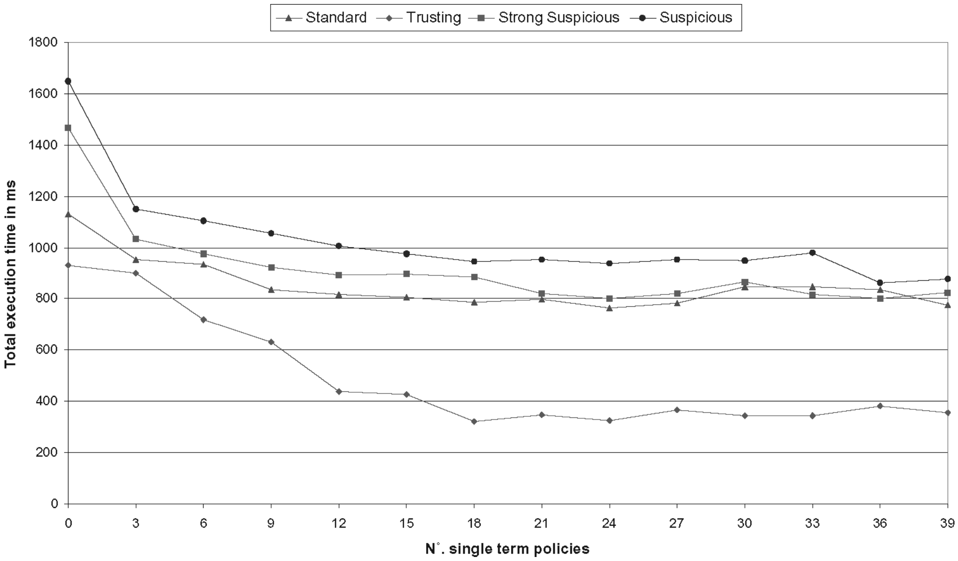

Fig. 7. Time requirements for the test cases: a tree of forty nodes and varying number of simple edges.

we keep the overall number of nodes in the tree invariant and replace three single-term policies with a multiple-term policy composed of three terms on the left side. Therefore, for each strategy, we measure the execution time when varying the number of single-term policies from forty to one scaling of a factor equal to three in each step. Among all possible alternative trust sequences, we always select the shortest one.

Test cases on the negotiation's tree height have been carried out considering negotiation trees of one-hundred and six nodes with four, six, and eight levels. The number of nodes at each level is kept constant. We have considered the cases in which the nodes are all connected by single edges (that corresponds to single term policies) and in which all the sibling nodes are connected by multiple edges (corresponding to multiple term policies).

10.2.2 *Experimental Results.* We have performed our experiments on a Pentium 4 PC with 2.00-GHz processor and with 512 MB of RAM, under Microsoft Windows XP. The same working load of the system was ensured in all the experiments. In addition, for each test case, eleven trials have been executed, and the average of the results obtained from the last ten trials has been computed, excluding the results of the first test. The performance has been measured in terms of CPU time (in milliseconds). In the following, we present and analyze the results of the evaluation of the negotiation strategies varying the complexity of the policies. It is noteworthy that the execution time of the trusting strategy is always lower then the one of the other strategies. This is motivated by the fact that, during the policy evaluation phase, the client receiving a suggestion does not have to check its own policy base, and thus avoids querying the policy data base. Figure 7 illustrates the trends of the strategies varying the number of single-term policies.

The trends of the four strategies are quite similar: the execution time decreases when increasing the number of single-term policies and it is approximately constant for the test cases corresponding to a number of single-term policies ranging from fifteen to thirty-nine. This is because, in these cases, the shortest trust sequence has always the same length (i.e., five terms). The highest execution times are obtained for test cases without single-term polices, because, in this case, we have a unique trust sequence that contains forty terms (all the negotiation tree's nodes). For the initial cases (corresponding to zero, three, and six single-terms policies), the execution times for all the strategies were very high compared to the subsequent tests. This is because, in these cases, most of the nodes in the negotiation tree are connected by multiple-term policies and, hence, the shortest trust sequence is longer than the one of the remaining cases. For example, in the initial case with zero single-term policies, where all the nodes are connected by multiple-term policies, the length of the trust sequence path is forty, corresponding to the number of nodes in the negotiation tree.

Because of lack of space, we do not report here the results of the experiments on the negotiation tree's height. However, the results of this set of experiments show that, overall, the execution time increases with a linear trend. As remarked above, the trusting strategy is, even in this set of experiments, the most efficient one.

11. RELATED WORK

Trust negotiation for web-based applications has been recognized as an interesting and challenging research area to explore and it has been extensively investigated in recent years. As a result, a variety of techniques and prototypes have been developed [Yu et al. 2003; Herzberg and J. Mihaeli 2000; Winsborough et al. 2000; Winsborough and Li 2002b].

Trust negotiation was introduced by Winsborough et. al. [2000], who presented two negotiation strategies for conducting on-line transactions between strangers. This prior work has been extended by Yu et. al. [2003]. They have developed families of strategies called disclosure tree protocols, which can interoperate in the sense that different parties can use different strategies of the same family. Support for mixed strategy, presented in Section 6.2.1, ensures that all Trust-$\mathcal{X}$ strategies be interoperable according to the above definition. We further extend such concept by introducing strategies that can be dynamically switched by a negotiator during a negotiation. With respect to the strategies proposed in Yu et al. [2003], the strategies we have developed fall in the family of parsimonious strategies, since each Trust-$\mathcal{X}$ strategy always discloses the minimal set of credentials required to satisfy counterpart requirements. However, while maximizing credential protection, our strategies also consider other requirements that negotiating parties may have, related to privacy practices and efficiency. Further, the problem of all the parsimonious strategies of implicitly revealing which credentials a party holds by transmitting policies is minimized in our approach by the use of policy preconditions within the context of a policy and can be further reduced by adopting a strongly suspicious strategy. None of the referenced approaches consider leakage of sensitive information nor protect policies. Such aspects have been explored by Seamons et al. [2001] and Winslett et al. [2003]. Winslett et al. [2003] have designed *Unipro*, that is a unified scheme to model resource protection, including policies. It is a prominent proposal in the negotiation research area and it has influenced our work. However, Unipro does not deal with the support of privacy policies nor does it define an ad hoc policy language.

Seamons et al. [2001] have explored the issue of supporting sensitive policies obtained by the introduction of hierarchies in policy definitions. Furthermore, they have also addressed privacy issues in trust negotiation [Seamons et al. 2002]. Such work provides an overview of some of the privacy problems that may arise during a negotiation. However, it does not provide a comprehensive solution to such problems, in that it only deals with the protection of sensitive policies achieved by the introduction of dynamic policies (policies dynamically modified during a negotiation by the trust negotiation system).

Li et al. [Winsborough and Li 2002b] introduced a role-based trust management language RT_0, which can be used to map entities to roles based on the properties described in their credentials. They have also developed an algorithm to locate and retrieve credentials that are not locally available. This aspect, referred to as *credential chain discovery*, is an important aspect of trust negotiation, since assuming the credentials to be locally stored is an assumption too strong for decentralized collaborative environments. Thus, mechanisms for retrieving remote credentials should be supported by every trust negotiation system. It is straightforward to integrate this feature into the system we have designed. Concerning system architectures for trust negotiation, Hess et al. proposed a trust negotiation in TLS (TNT) handshake protocol by adding trust negotiation features. Winslett et al. [2002] proposed the TrustBuilder architecture for trust negotiation systems. The TrustBuilder architecture includes a credential verification module, a policy compliance checker, and a negotiation strategy module, which is the core of the system. More recently [Lee et al. 2006], the same group of researchers released Traust, which leverages the former TrustBuilder and acts as an agent between a browser and a portal acting as a service provider offering the trust negotiation service.

The Trust Establishment Project at Haifa Research Lab has developed a tool (TE) for enabling trust relationships between strangers based on public key certificates. The TE system includes an intelligent certificate collector that automatically collects missing certificates from certificate repositories, allowing the use of standard browsers that can only pass one certificate to the server. However, it does not provide support for sensitive credentials. One of the TE's basic assumptions is that credentials can be disclosed whenever they are requested. Further, the TE system does not have the notion of sensitive policies.

The idea of selectively disclosing credential attributes presented in Section 5.2 is not new [Brands 2000; Persiano and Visconti 2000; Jarvis 2003]. However, this technique has never been thoroughly explored, especially in trust negotiations. Two significant approaches dealing with this topic are by Bradshaw et al. [2004] and Li et al. [2003]. Bradshaw's work focuses on hidden credential features and on how to improve performance of hidden credentials constructed from identity-based crypto systems, which satisfies credential indistinguishability. The authors propose an improved secret-splitting scheme and show how transactions, which depend on fulfillment of policies described by monotonic boolean formulas, can take place in a single round of messages.

Li et al proposed a scheme called oblivious signature-based envelope [Li et al. 2003] (OSBE). OSBEs are similar to hidden credentials in that the ability to read a message is contingent on having been issued the required secret. Our focus, differently from Li et al. [2003], is to deeply analyze the impact of protected attribute credentials on trust negotiations and devise new strategies to allow interoperability between users adopting various credential formats.

Several privacy-enabled identity management systems have been based on the notion of anonymous credential [Camenisch and Herreweghen 2002; Chaum 1985]. In anonymous credential systems, organizations know the users only by pseudonyms. Different pseudonyms of the same user cannot be linked. Yet, an organization can issue a credential to a pseudonym and the corresponding user can prove possession of this credential to another organization (who knows her by a different pseudonym), without revealing anything more than the fact that she owns such a credential. Idemix [Camenisch and Herreweghen 2002] is the first system implementing anonymous credentials in a federated identity management system. Idemix provides mechanisms for efficient multishow[10] credentials and a flexible scheme for issuing and revoking anonymous credentials. It also provides a mechanism for *all or nothing* sharing and PKI-based nontransferability. Anonymous credentials however may not be adequate for several real world e-commerce applications and web services that require disclosure of various attributes. In our approach, we do not require the user identity to be hidden, even if we protect his/her attributes. More specifically, we control the use of sensitive attributes without necessarily requiring anonymity. By using suspicious and strong suspicious strategies, we can carry on trust negotiations limiting the risk of unnecessary collection of user attributes and credentials.

12. CONCLUSIONS

In this paper, we have presented a system for trust negotiations specifically designed for preserving privacy. The system provides support for P3P policies that can be exchanged at various steps of the negotiation, and different credential formats, providing different degrees of privacy protection. In addition, the system provides a suite of strategies to carry on a negotiation, that exploits the notion of context associated with a policy and allows one to trade-off among efficiency, robustness, and privacy requirements.

The work presented in this paper is ongoing and some issues still need to be explored, especially with respect to the prototype. The system currently implemented is to be intended as a core system, which we realized to prove

[10] Credentials can be used multiple times. Possession of a multishow credential can be demonstrated an arbitrary number of times; these demonstrations cannot be linked to each other [Camenisch and Herreweghen 2002].

the correctness and the consistency of our solution, as well as to test its performance. However, it is not to be intended as an industrial product to be released to end users. Trust negotiations in PP-Trust-$\mathcal{X}$ may be controlled by actual individuals, as well as by automatic agents which, based on specific settings set by the user, may run automated trust negotiations. In fact, in the current prototype, the user can select whether to control step by step the execution of the negotiation or let the system managing the negotiation in an automated fashion. However, we need to further extend the algorithms controlling the automatic negotiations, so that, based on the input of the user, the strategies can be dynamically selected as the negotiation proceeds.

As illustrated, the system has been developed to simplify as much as possible the load of a user wishing to interact with unknown parties using PP-Trust-$\mathcal{X}$. However, the amount of information to be submitted by the user may be significant and may often need to be updated. Therefore, we will enhance the current graphical interfaces to ease these tasks and provide new user-friendly tools for policy specification and management. For instance, we plan to extend the prototype by developing mechanisms and modules to semi-automatically design privacy policies to be associated with the policies. With respect to the previous releases of PP-Trust-$\mathcal{X}$, we have improved the efficiency of our system by implementing the functionalities to carry on a trust negotiation as operations offered by a Web service and by reengineering the key algorithms of the system. We have reduced the complexity of the application at the user side, since the main functionalities are remotely executed by the Web service in an optimized manner. We will keep optimizing the PP-Trust-$\mathcal{X}$ prototype to further reduce the computational overhead.

APPENDIX

A. THE ACTION PROCESSING FUNCTION

In this appendix, we present the *Action_processing*() function (Figure A.1) and *Privacy_matching* function Figure A.2. *Action_processing*() (shown in Figure A.1) is in charge of correctly processing a received action during the policy evaluation phase (see Table I). As shown, the function constraints, but does not determine, a single response for each incoming message. It is important to note that the function implements the pure strategies, as they have been presented in Section 6.2. However, it is trivial to modify action responses in order to support the relaxations we have discussed after presenting the strategies.

B. PRIVACY POLICIES

In this section, we report the encoding of two type privacy policies that might be used during trust negotiations. The first type is a coarse-grained privacy policy, and is given in Figure B.1. The second policy we report in Figure B.2 is a fine-grained policy and refers to a specific credential. The policy is attached to disclosure policies, as explained in Section 5.

```
Function Action_processing($m$, $strategy$)
$Input$ :
    $m$:= a valid action;
    $strategy \in \{suspicious, trusting, strongly\ suspicious, mixed\}$;
$Output$ :
    $Reply\_act = \{m_1, .., m_k\}$ a list of replies
$Precondition$:
    Each credential is privacy enhanced

begin
Let $NT = \langle \mathcal{N}, \mathcal{R}, \mathcal{E}, \phi \rangle$ be the negotiation tree for $\mathcal{R}$
Let $PB$ and $\mathcal{X}$-Prof be the policy base and the $\mathcal{X}$-Profile of the
receiving party, respectively
$Reply\_act := \emptyset$;
      switch($m$)
            case: $m = Attr\_req(a)$
               Let $\mathcal{T}$ be the term conveying $a$ and
               let $n$ be the corresponding node
               if $strategy = strongly\ suspicious$ then
                  if $state(n) = DELIV$ then
                  Let $\mathcal{C}$ be the credential in $\mathcal{X}$-Prof proving attribute $a$;
                     $Reply\_act = Reply\_act \cup Attr\_send(C.a)$
                     $state(n) = attr\_discl$;
                     $UpdateState(NT, n)$;
            % function that verifies whether the attr_discl state can be propagated up to the tree
                  else       $Reply\_act = Reply\_act \cup Attr\_refuse(\mathcal{T})$
               else          $Reply\_act = Reply\_act \cup Attr\_refuse(\mathcal{T})$
            case: $m = Cred\_proof\_disclosed\_req(\mathcal{T})$
               if $strategy = suspicious$ then
               Let $n$ be the node conveying $\mathcal{T}$
                Let $\mathcal{C}$ be the credential in $\mathcal{X}$-Prof satisfying term $\mathcal{T}$
                  if $state(n) = DELIV$ then
                     $Reply\_act = Reply\_act \cup Cred\_proof\_disclosed\_send(\mathcal{T})$
                     $state(n) = cred\_proof\_disclosed$;
                     $UpdateState(NT, n)$; % Update the tree state
                  else       $Reply\_act = Reply\_act \cup Cred\_proof\_disclosed\_Ref(\mathcal{T})$
               else          $Reply\_act = Reply\_act \cup Cred\_proof\_disclosed\_Ref(\mathcal{T})$
            case: $m = Send(p_c)$
            $UpdateTree(NT, p_c.rid)$        %Negotiation tree update
               if $p_c.sugg \neq \emptyset$ then
                  Let $p_c.sugg.list$ be $\{< \mathcal{T}_{j1}; \mathcal{T}_{i1}, ..., \mathcal{T}_{in} > ... < \mathcal{T}_{jz}; \mathcal{T}_{i1}, ..., \mathcal{T}_{in} >\}$
                  Let $R \leftarrow \mathcal{T}_1, .., \mathcal{T}_k$ be the rule contained in $p_c$;
                  if $\forall\ \mathcal{T}_i$ in $\{\mathcal{T}_1, .., \mathcal{T}_k\}$ $\exists p \in \mathcal{PB} | p$ contains a rule of the form $\mathcal{T}_i \leftarrow \mathcal{T}_{i1}, ..., \mathcal{T}_{in}$
                          and $< \mathcal{T}_i; \mathcal{T}_{i1}, ..., \mathcal{T}_{in} > \in p_c.sugg.list$ or $\mathcal{T}_i \leftarrow DELIV$ then
                  $Reply\_act$=$Reply\_act \cup Accept(p_c.sugg)$
                     For each $p \in PB | p.rid = \mathcal{T}_i \leftarrow \mathcal{T}_{i1}, ..., \mathcal{T}_{in}$
               and $< \mathcal{T}_i; \mathcal{T}_{i1}, ..., \mathcal{T}_{in} > \in p_c.sugg.list$
                       $UpdateTree(NT, p.rid)$ %update the negotiation tree
                  else
                     if $p_c.op = -$ then
                             $Reply\_act$=$Reply\_act \cup Refuse(p_c.sugg)$
                     else            %suggestion failure, alternative policies are sent
                           for each $p_c \in PB$ such that
                           $p_c.pid$ denotes a rule $\mathcal{T} \leftarrow \mathcal{T}_{j1}, .., \mathcal{T}_{jk}$ and $\mathcal{T} \in \{\mathcal{T}_1, .., \mathcal{T}_k\}$
                          $Reply\_act$=$Reply\_act \cup Send(p_c)$;
            $Reply\_act$=$Reply\_act \cup Privacy\_matching(p_c.priv)$; %matching privacy preferences
Return{$Reply\_act$}
end
```

Fig. A.1. Function *Action_processing()*.

```
Function Privacy_matching($priv$)
$Input$ :
    $priv$: a privacy policy or an empty message;
$Output$ :
    Reply_action assuming either value $Accept(priv)$, or $Refuse(priv)$
$Precondition$:
    $Priv\_nego$ is the remote P3P policy exchanged in the privacy agreement phase
    $Priv\_pref$ are the local privacy preference rules exchanged in the privacy agreement phase
begin
  if $priv \neq \emptyset$      %matching privacy preferences
      Let $p_c$ be the policy to which $priv$ is associated
      Let $rule$ be a local preference rule
      if $\exists p$ for credentials requested in $p_c$ then
      % Match is a function checking compliance between a privacy rule and p3p policy
        if Match($p, priv$)=TRUE then
        % there exists a specific rule for the requested credentials
          Return($Accept(p_c.priv)$)

        else Return($Refuse(p_c.priv)$)
      else
        if Match($priv\_pref, priv$)=TRUE then
          Return ($Accept(p_c.priv)$)
        else Return($Refuse(p_c.priv)$)
  else
  if $p_c.priv = \emptyset$      %matching privacy preferences
      Let $rule$ be a local preference rule
      if $\exists rule$ for credentials requested in $p_c$ then
        if Match($rule, Priv\_nego$)=TRUE then
        % there exists a specific rule for the requested credentials
          Return ($Accept(p_c.priv)$)
        else Return($Refuse(p_c.priv)$)
end
```

Fig. A.2. Function *Privacy_matching*().

C. FORMAL PROOFS

Proof of Theorem 6.1. In what follows we prove Theorem 6.1. Formal proof is given considering each possible strategy. We first report the result obtained in our previous paper [Bertino et al. 2004b]. The theorem ensures correctness of the standard strategy and refers to a function, called *SequenceGenerator*, that outputs trust sequences from valid views of Negotiation trees.

THEOREM C.1. *[Bertino et al. 2004b] Let NT be a negotiation tree for a resource $\mathcal{R}$, and let $\mathcal{WT}$ be a valid view on NT. Let $TS' = [R, C_1, .., C_n]$ be the output of Function SequenceGenerator when its input is $\mathcal{WT}$. TS' is a trust sequence for $\mathcal{R}$.*

1. **Suspicious Strategy**. The proof is by induction on the height of the view $\mathcal{WT}$ and exploits the results of Theorem C.1. Here and in the following proofs, we do not consider policy prerequisites. Results for this case can be easily obtained using the same strategies used for the standard strategy.
   a. *Basis*. $\mathcal{WT}$ is a view of height $h = 1$ and has two edges. We do not consider the case when $\mathcal{WT}$ has only one edge, since it is trivial. Thus, $\mathcal{WT}$ consists of three nodes: $\mathcal{R}$, n_1, and n_2. We first consider the case in which the edges are simple. This implies that there exists two simple edges connecting $\mathcal{R}$ with n_1, and n_2, respectively, where $\mathcal{R}$ is the node corresponding to the requested resource R. By hypothesis, $\mathcal{WT}$ is

```
<POLICY xmlns="http://www.w3.org/2000/P3PV1>

......
    <STATEMENT>
        <DATA-GROUP>
          <DATA ref="#dynamic.misc.data" >
              <CATEGORIES>
              <uniqueid/> <state/>
              </CATEGORIES>
          </DATA>
         </DATA-GROUP>
      <ACCESS> <contact_and_other> </ACCESS>
        <!-- Use (purpose)-->
      <PURPOSE resolution-type="independent">
        <current/>
        <admin/>
      </PURPOSE>
      <RECIPIENT> <ours/><same/></RECIPIENT>
      <RETENTION> <stated-purpose/> </RETENTION>
    </STATEMENT>

    <STATEMENT>
          <DATA-GROUP>
              <DATA ref="#dynamic.misc.data" >
              <DATA ref="#user.home-info.online.postal" >
                <CATEGORIES><online/> </CATEGORIES>
          </DATA>
          </DATA-GROUP>
        <ACCESS> <contact_and_other> </ACCESS>
          <PURPOSE>
            <contact-required="opt-in"/>
            <individual-decision="opt-in"/>
        </PURPOSE>
        <RECIPIENT> <ours/></RECIPIENT>
          <RETENTION> <business-practices/> </RETENTION>
    </STATEMENT>
</POLICY>
```

Fig. B.1. Example of coarse-grained P3P policy.

a valid view built using the suspicious strategy. Thus, $\mathit{state}(n_1) = \mathit{cred\_proof\_disclosed}$ and $\mathit{state}(n_2) = \mathit{cred\_proof\_disclosed}$. $\mathit{state}(n_1) = \mathit{cred\_proof\_disclosed}$, and $\mathit{state}(n_2) = \mathit{cred\_proof\_disclosed}$ implies that $X\text{-}\mathit{prof}_{rq}$ contains two credentials C_1 and C_2 satisfying terms $\mathcal{T}(n_1)$ and $\mathcal{T}(n_2)$ in n_1 and n_2, respectively. According to the conditions required for a credential proof disclosure under a suspicious strategy (see function *Action_processing*()), Cp_1 and Cp_2 are disclosed only if $\mathit{state}(n_1) = \mathit{deliv}$ and $\mathit{state}(n_2) = \mathit{deliv}$, where Cp_1 and Cp_2 denote the credential proofs of C_1 and C_2, respectively. Thus, $\mathcal{WT}$ is equivalent to a valid view $\mathcal{WT}'$ with all delivery nodes built according to the standard strategy. By Theorem C.1, $\mathcal{WT}'$ corresponds to a trust sequence TS containing all and only the credentials corresponding to terms in the view, ending with the safe disclosure of R. Thus, $\mathcal{WT}'$ is correct. Since $\mathcal{WT}'$ is equivalent to $\mathcal{WT}$, then also $\mathcal{WT}$ is correct and the theorem holds. The case for multi edges is very similar; thus, we do not report here the proof.

b. *Inductive step*. Consider a valid view $\mathcal{WT}$, with height $h > 1$. Suppose that the thesis holds for views of height $h' < h$ and let us prove the thesis for h. By inductive hypothesis, if we consider a valid view $\mathcal{WT}'$ of height $h' = h - 1$, then an equivalent view $\mathcal{WT}''$ can be found using the standard

```
<POLICY xmlns="http://www.w3.org/2000/P3PV1>
....
    <STATEMENT>
        <DATA-GROUP>
          <DATA ref="http://www.TrustX.repos.credtype#idCard.name">
          <DATA ref="http://www.TrustX.repos.credtype#idCard.lastname">
          <DATA ref="http://www.TrustX.repos.credtype#idCard.birthdate">
          <DATA ref="http://www.TrustX.repos.credtype#idCard.key">
              <CATEGORIES>
              <purchase/>
              </CATEGORIES>
          </DATA>
        </DATA-GROUP>
      <ACCESS> <contact_and_other> </ACCESS>
      <PURPOSE resolution-type="independent">
        <current/>
        <develop />
      </PURPOSE>
      <RECIPIENT> <ours/><same/></RECIPIENT>
      <RETENTION> <stated-purpose/> </RETENTION>
    </STATEMENT>
    <STATEMENT>
          <DATA-GROUP>
              <DATA ref="http://www.TrustX.repos.credtype#idCard.street">
              <DATA ref="http://www.TrustX.repos.credtype#idCard.stateprov">
              <DATA ref="http://www.TrustX.repos.credtype#idCard.postalCode">
              <DATA ref="http://www.TrustX.repos.credtype#idCard.country">
              <DATA ref="http://www.TrustX.repos.credtype#idCard.e-mail">
              <DATA ref="http://www.TrustX.repos.credtype#idCard.phone">
                <CATEGORIES><purchase/> </CATEGORIES>
          </DATA>
          </DATA-GROUP>
        <ACCESS> <contact_and_other> </ACCESS>
          <PURPOSE>
            <contact/>
            <individual-decision/>
        </PURPOSE>
        <RECIPIENT> <ours/></RECIPIENT>
          <RETENTION> <indefinitely/> </RETENTION>
    </STATEMENT>
</POLICY>
```

Fig. B.2. Example of fine-grained P3P privacy policy.

strategy. Consider a leaf node n in $\mathcal{WT}'$. Let us add a set of edges originating at n, such that the length of the view becomes h. Let us first add a simple edge $e = (n, n_1)$. According to the suspicious strategy, if this edge is added to the view, there must exist a policy containing a rule r_c relating $\mathcal{T}(n)$ with $\mathcal{T}(n_1)$. Moreover, since by hypothesis the view is valid, $\mathit{state}(n_1)$ must be *cred_proof_disclosed*. Suppose, without loss of generality, that the policy belongs to $\mathcal{RQ}$. If $\mathit{state}(n_1) = \mathit{cred\_proof\_disclosed}$, then $X\text{-}\mathit{prof}_{cn}$ must contain a credential C_1 satisfying term $\mathcal{T}(n_1)$. Furthermore, C_1 is a privacy enhanced credential, and the credential proof Cp_1 for C_1 can be disclosed. According to the conditions required for a credential proof disclosure under the suspicious strategy (see function *Action_processing*()), Cp_1 is disclosed only if $\mathit{state}(n_1) = \mathit{deliv}$. Thus, by concatenating node n_1 with the view $\mathcal{WT}''$ which exists by inductive hypothesis, we obtain a valid view of height h built according to the standard strategy. Thus, by Theorem C.1, this view corresponds to a trust sequence *TS* containing all and only the credentials corresponding to terms in the view, ending with the safe disclosure of R. Thus, also the corresponding view built using the suspicious strategy is correct and the theorem holds.

We are left to consider the case when we add a set of $j > 1$, edges rooted at n. We omit the proof for this case since it can be done using the same reasonings we have applied for the basis step.

2. **Trusting Strategy**. The proof is by induction on the height of the view $\mathcal{WT}$.

a. *Basis*. $\mathcal{WT}$ is a view of height $h = 2$. We do not consider the case when $\mathcal{WT}$ has height 1, since it is trivial. In addition, we assume that in building the tree the *sugg* component of a policy is always taken into account. We first consider the case in which the edges in the view are simple. This implies that there exist two simple edges: one connecting $\mathcal{R}$ with n_1 and the other connecting n_1 with n_2, where $\mathcal{R}$ is the node in the tree corresponding to the requested resource R. By hypothesis, $\mathcal{WT}$ is a valid view built using the trusting strategy, then the policy base of $\mathcal{RQ}$ must contain a disclosure policy p'_c containing a rule r'_c of the form: $\mathcal{T}(n_1) \leftarrow \mathcal{T}(n_2)$. Furthermore, the policy base of $\mathcal{CN}$ must contain a disclosure policy p_c of the form: $(r_c; \ldots < \mathcal{T}(n_1); \mathcal{T}(n_2) >)$,[11] such that: (i) r_c is a rule of the form: $R \leftarrow \mathcal{T}(n_1)$; and (ii) $< \mathcal{T}(n_1); \mathcal{T}(n_2) >$ is the suggestion stored in the *sugg* field of the policy context. Finally, since $\mathcal{WT}$ is a valid view it must be composed by all delivery nodes. Thus, $\mathcal{PB}_{cn}$ must contain a rule r''_c of the form: $\mathcal{T}(n_2) \leftarrow deliv$. Thus, by simply appending on the tree, the nodes corresponding to rules r_c, r'_c, and r''_c, following the standard strategy, a valid view $\mathcal{WT}'$, for the standard strategy, rooted at $\mathcal{R}$ and composed by edges $e' = (\mathcal{R}, n_1)$ and $e'_2 = (n_1, n_2)$ is determined. By Theorem C.1, $\mathcal{WT}'$ corresponds to a trust sequence *TS* containing all and only the credentials corresponding to terms in the view, ending with the safe disclosure of R is equivalent to $\mathcal{WT}'$ it is correct and the theorem holds.

We are left to consider the case for multi edges. Thus, we suppose that there exists a multi edge $e = \{(\mathcal{R}, n_1)(\mathcal{R}, n_3)\}$ connecting $\mathcal{R}$ with n_1 and n_3. Since the height of the view is $h = 2$, n_1 and n_3 are nonleaf nodes. We assume, without loss of generality, that n_1 and n_3 are connected by simple edges to two further nodes n_2 and n_4. By hypothesis, $\mathcal{WT}$ is a valid view built using the trusting strategy, then the policy base of $\mathcal{RQ}$ must contain two disclosure policies containing two rules, r'_c and r^v_c, respectively, of the form: $\mathcal{T}(n_1) \leftarrow \mathcal{T}(n_2)$ and $\mathcal{T}(n_3) \leftarrow \mathcal{T}(n_4)$. Furthermore, since the tree is built adopting a trusting strategy, the Policy Base of $\mathcal{RQ}$ must contain a disclosure policy p_c of the form: $(\mathcal{R} \leftarrow \mathcal{T}(n_1), \mathcal{T}(n_3); \ldots < \mathcal{T}(n_1); \mathcal{T}(n_2) >< \mathcal{T}(n_3); \mathcal{T}(n_4) >)$, such that: (i) $\mathcal{R} \leftarrow \mathcal{T}(n_1)\mathcal{T}(n_3)$ is a rule; and (ii) $< \mathcal{T}(n_1); \mathcal{T}(n_2) >< \mathcal{T}(n_3); \mathcal{T}(n_4) >$ is the suggestion stored in the policy context. Finally, since the view is composed of all delivery nodes, two rules of the form: $\mathcal{T}(n_2) \leftarrow deliv$ and $\mathcal{T}(n_4) \leftarrow deliv$ must belong to $\mathcal{PB}_{cn}$. We denote these two rules by r''_c and r'''_c, respectively. Thus, by simply appending to the tree the nodes

[11] For simplicity here and in the following we consider only the sugg component in the policy context, since the remaining components are not relevant for the proof.

corresponding to the rules above, following the standard strategy, a valid view $\mathcal{WT}'$ for the standard strategy, rooted at $\mathcal{R}$ and composed by edges $e' = \{(\mathcal{R}, n_1)(\mathcal{R}, n_2)\}$ and $e'_2 = (n_1, n_2)$ $e'_3 = (n_3, n_4)$ is determined. By Theorem C.1, $\mathcal{WT}'$ corresponds to a trust sequence TS containing all and only the credentials corresponding to terms in the view, ending with the safe disclosure of R. Thus, also $\mathcal{WT}$ is correct, since it is equivalent to $\mathcal{WT}'$, and the theorem holds.

b. *Inductive step*. Consider a valid view $\mathcal{WT}$, with height $h > 2$. Suppose that the thesis holds for views of height $h' < h$ and let us prove the thesis for h. By inductive hypothesis, if we consider a valid view $\mathcal{WT}'$ of height $h' = h-2$, then a corresponding valid view $\mathcal{WT}''$ can be found using the standard strategy. Consider a leaf node n in $\mathcal{WT}'$ and suppose, without loss of generality, that $\mathit{party}(n) = \mathcal{CN}$. Let us add two edges to this view (the case when we add only one edge is trivial). Thus, suppose we add two edges $e = (n, n_1)$ and $e_2 = (n_1, n_2)$, such that the length of the view becomes $h' + 2 = h$. Since, by hypothesis, the resulting view is valid, then the policy base of $\mathcal{RQ}$ must contain a disclosure policy p'_c containing a rule r'_c of the form: $\mathcal{T}(n_1) \leftarrow \mathcal{T}(n_2)$. Furthermore, since the tree is built adopting a trusting strategy, the policy base of $\mathcal{CN}$ must contain a disclosure policy p_c of the form: $(\mathcal{T}(n) \leftarrow \mathcal{T}(n_1);\ \ldots < \mathcal{T}(n_1); \mathcal{T}(n_2) >)$, such that: (i) $\mathcal{T}(n) \leftarrow \mathcal{T}(n_1)$ is a rule; and (ii) $< \mathcal{T}(n_1); \mathcal{T}(n_2) >$ is the suggestion stored in the policy context. Finally, since the view is valid, it is composed of all delivery nodes. Thus, a rule r''_c of the form: $\mathcal{T}(n_2) \leftarrow \mathit{deliv}$ must belong to the policy base of CN. By appending the nodes corresponding to rules r_c, r'_c, and r''_c, to the existing view $\mathcal{WT}''$, built according to the standard strategy, which exists by hypothesis, a valid view $\mathcal{WT}'$ of height h can be found which is equivalent to the view $\mathcal{WT}$ of the same height built using the trusting strategy. Based on the same reasonings we have done for the basis step, we can deduce that $\mathcal{WT}$ is correct, and the theorem holds.

We omit the proof for multi edges since it is very similar to the one for simple edges.

3. **Strongly Suspicious Strategy**. The proof is by induction on the height of the view $\mathcal{WT}$.

a. *Basis*. $\mathcal{WT}$ is a view of height $h = 1$ and has two edges. We do not consider the case when $\mathcal{WT}$ has only one edge, since it is trivial. Thus, $\mathcal{WT}$ consists of three nodes: $\mathcal{R}$, n_1, and n_2. We first consider the case in which the edges are simple. This implies that there exist two simple edges connecting $\mathcal{R}$ with n_1, and n_2, respectively. By hypothesis, $\mathcal{WT}$ is a valid view built using the suspicious strategy. Thus, $\mathit{state}(n_1) = \mathit{I\_discl}$ and $I(n_2) = \mathit{attr\_discl}$. $\mathit{state}(n_1) = \mathit{attr\_discl}$, and $\mathit{state}(n_2) = \mathit{attr\_discl}$ imply that $X\text{-}\mathit{prof}_{RQ}$ contains two credentials C_1 and C_2 satisfying terms $\mathcal{T}(n_1)$ and $\mathcal{T}(n_2)$ in n_1 and n_2, respectively. Furthermore, C_1 and C_2 contain attributes with names a_1 and a_2, respectively, specified in the corresponding terms. According to the conditions required for an attribute disclosure under a suspicious strategy (see function *Action_processing*()), credential attributes C_1a_1 and C_1a_2 are disclosed only if $\mathit{state}(n_1) = \mathit{deliv}$ and $\mathit{state}(n_2) = \mathit{deliv}$. Thus,

$\mathcal{WT}$ is equivalent to a valid view $\mathcal{WT}'$ with all delivery nodes built according to the standard strategy. By Theorem C.1, $\mathcal{WT}'$ corresponds to a trust sequence *TS* containing all and only the credentials corresponding to terms in the view, ending with safe disclosure of $\mathcal{R}$. Thus, $\mathcal{WT}'$ is correct. Since $\mathcal{WT}'$ is equivalent to $\mathcal{WT}$ then also $\mathcal{WT}$ is correct and the theorem holds. The case for multi edges is similar, thus we do not report here the proof.

b. *Inductive step*. Consider a valid view $\mathcal{WT}$, with height $h > 1$. Suppose that the thesis holds for views of height $h' < h$ and let us prove the thesis for h. By inductive hypothesis, if we consider a valid view $\mathcal{WT}'$ of height $h' = h - 1$, then an equivalent view $\mathcal{WT}''$ can be found using the standard strategy. Consider a leaf node n in $\mathcal{WT}'$. Let us add a set of edges originating at n, such that the length of the view becomes h. Let us first add a simple edge $e = (n, n_1)$. According to the suspicious strategy, if this edge is added to the view, there must exist a policy having a rule r_c relating $\mathcal{T}(n)$ with $\mathcal{T}(n_1)$. Moreover, since by hypothesis the view is valid, $state(n_1)$ must be $attr\_discl$. Suppose, without loss of generality, that the policy belongs to $\mathcal{RQ}$. If $state(n_1) = attr\_discl$, then $X\text{-}prof_{cn}$ must contain a credential C_1 satisfying term $\mathcal{T}(n_1)$. Furthermore, C_1 contains attribute name a_1, specified in the corresponding term. According to the conditions required for a credential proof disclosure under the suspicious strategy (see function *Action_processing*()), $C_1 a_1$ is disclosed only if $state(n_1) = deliv$. Thus, by concatenating node n_1 with the view $\mathcal{WT}''$, which exists by inductive hypothesis, we obtain a valid view of height h built according to the standard strategy. Thus, by Theorem C.1, this view corresponds to a trust sequence *TS* containing all and only the credentials corresponding to terms in the view, ending with the safe disclosure of R. Thus, also the corresponding view built using the suspicious strategy is correct, and the theorem holds.

Thus, we are left to consider the case when we add a set of $j > 1$, edges rooted at n. We omit the proof for this case, since it can be done using the same reasonings we have applied for the basis step. □

Proof of Theorem 6.2. The proof is by induction on the height of the view *WT*.

a. *Basis*. Let us first suppose that $\mathcal{WT}$ is a view of height $h = 1$. Since $\mathcal{WT}$ is built according to a mixed strategy, a single step of the tree building can be performed using different strategies. If $\mathcal{WT}$ is built following a standard strategy, then, $\mathcal{WT}$ correctness is ensured by Theorem C.1. Similarly, if $\mathcal{WT}$ is built according to a suspicious strategy then, by the results proved by Theorem 6.1, case 1, $\mathcal{WT}$ is correct. If the adopted strategy is a strong suspicious strategy, correctness is ensured by Theorem 6.1, case 2. of the proof.

Let us now consider the case when the view is built according to a trusting strategy. Then, according to a trusting strategy, $\mathcal{WT}$ grows of two levels in one single step (view height is $h = 2$) and the thesis holds by results of case 3 of the proof of Theorem 6.1.

b. *Inductive step*. Consider a valid view $\mathcal{WT}$, with height $h > 1$. Suppose that the thesis holds for views of height $h' < h$ and let us prove the thesis for h. By inductive hypothesis, a correct view $\mathcal{WT}'$ of height $h' = h - 1$ can be built using a mixed strategy. Consider a leaf node n in $\mathcal{WT}'$. Let us add new nodes in the view $\mathcal{WT}'$ so that $\mathcal{WT}'$ becomes of height h. Suppose, without loss of generality, that n belongs to $\mathcal{RQ}$. If $\mathcal{CN}$ follows a standard strategy in adding the new nodes, then the correctness of the resulting view $\mathcal{WT}$ follows from Theorem C.1. Similarly, if nodes are added following a suspicious strategy, correctness of the resulting view is proved by case 1 of the proof. If the adopted strategy is a strong suspicious strategy, correctness is ensured by case 3 of the proof. Finally, if a trusting strategy is adopted $\mathcal{WT}$ grows of two levels in one single step (height is $h' = h + 2$) and the thesis holds by results obtained in case 3 of proof of Theorem 6.1.

Proof of Theorem 6.3. In what follows we prove Theorem 6.3. Formal proof is given considering each possible strategy.

1. **Suspicious Strategy**. The proof is by induction on the number K of messages exchanged during the policy evaluation phase of a negotiation. Here and in the following, we do not consider policy prerequisites. Results for this case can be easily obtained using the same strategies used for the standard strategy (see Bertino et al. [2004b]). For simplicity, we limit our analysis to policies requiring a single credential at a time. The proof for policies requiring more credentials is an immediate consequence of the results of the proved case.
   a. *Basis*. We start with $k = 2$, that is, one message for each negotiating party is exchanged. By hypothesis, $\mathcal{RQ}$ and $\mathcal{CN}$ trust requirements can be satisfied. Thus, by definition 6.1, the adopted strategy needs to be compliant with the local settings of $\mathcal{RQ}$ and $\mathcal{CN}$. Let $\mathcal{R}$ be the resource requested by $\mathcal{RQ}$ to $\mathcal{CN}$. Assume $\mathcal{CN}$ trust requirements for R are defined in terms of a policy *pol* requiring a credential *cred*. Then, when $\mathcal{CN}$ executes the *Action_processing*() function, the message sent by $\mathcal{CN}$ takes the form of a *Cred_proof_req*(*cred*) action. d reply actions for a *Cred_proof_req*(*cred*) from CN side are either *Cred_proof_disclosed_send*(*cred*), *Send*(*pol*) or *Cred_proof_disclosed_ref*(*cred*). If a *Cred_proof_send*(*cred*) is sent, then the negotiation successfully succeeds and the thesis holds. A *Send*(*pol*) message cannot be sent, since, by hypothesis, the number of messages to exchange is two. Similarly, a *Cred_proof_disclosed_ref*() action cannot be sent since it causes negotiation failure, but this is in contradiction with parties trust requirements satisfiable hypothesis.
   b. *Inductive step*. Suppose we have proved the thesis for negotiations requiring $k' < k$ messages and let us prove the thesis for k. Let m be the $k - 1$ message shipped. Assume without loss of generality, that m is sent by $\mathcal{CN}$ to $\mathcal{RQ}$. Two valid actions are possible. If the sent message was a valid reply action for a previous request of a credential proof *cred* and no further policies are adopted for *cred* in PB_{rq}, then a sequence of all delivery credentials can found and then negotiation succeeds. By contrast, if m = *Cred_proof_req*(*cred*), then $\mathcal{RQ}$ possible replies are the following. If

there exists a policy in PB_{rq} for *cred*, requiring a credential $cred_1$, then according to the *Action_Processing*() function the valid message to be sent is *Cred_proof_req*($cred_1$). $\mathcal{CN}$, in turn, will either send a counter policy for $cred_1$ or send a *Cred_proof_send*($cred_1$) message. However, a *Send*(*pol*) message cannot be sent since by hypothesis the number of messages to be exchanged is set to k. In the other case a trust sequence is found and the thesis holds. The only other potential reply is *Cred_proof_refuse*(*cred*) message, which causes process failure, leading to an absurd for our assumptions.

2. **Strongly Suspicious Strategy**. The proof is by induction on the number of messages exchanged during the policy evaluation phase of a negotiation.
   a. *Basis*. We start with $k = 2$, that is, one message for each negotiating party is exchanged. Let $\mathcal{R}$ be the requested resource from $\mathcal{RQ}$ to $\mathcal{CN}$. By hypothesis, $\mathcal{RQ}$ and $\mathcal{CN}$ trust requirements can be satisfied. Thus, by definition 6.1, the adopted strategy needs to be compliant with the counterpart local settings of $\mathcal{RQ}$ and $\mathcal{CN}$. Then, when $\mathcal{CN}$ executes the *Action_processing*() function, the message sent by CN takes the form of a *Attr_req*(a) action.

   Valid reply actions for a $Attr_{req}(cred)$ from $\mathcal{CN}$ side, are either $Attr_{send}(C.a)$, where C denotes the credential conveying the requested attribute, *Send*(*pol*), or *Attr_refuse*($C.a$). If a *Attr_send*($C.a$) is sent, then the negotiation succeeds and the thesis hold. A *Send*(*pol*) message cannot be sent since by hypothesis the number of messages to exchange is set to 2. Similarly, a *Attr_refuse*(a) action cannot be sent, since it causes negotiation failure, but this is in contradiction with parties trust requirements satisfiable hypothesis.

   b. *Inductive step*. Suppose we have proved the thesis for negotiations requiring $k' < k$ messages and let us prove the thesis for k. Let m be the $k - 1$ message shipped. Assume, without loss of generality, that m is sent by $\mathcal{CN}$ to $\mathcal{RQ}$. Two valid actions are possible. If the sent message was a valid reply action for a previous request of an attribute *attr* and no further policies are adopted for *attr* in PB_{rq}, then a sequence of all delivery credentials can found and then negotiation succeeds. By contrast, if $m = Attr\_req(attr)$, then $\mathcal{RQ}$ possible replies are the following. If there exists a policy in PB_{rq} for attributes a, requiring one or more remote attributes $attr_1, \ldots, attr_n$, then, according to the *Action_Processing*() function, the valid messages that can be sent are *Attr_req*($attr_1$), ... ,*Attr_req*($attr_n$). $\mathcal{CN}$, in turn, will either send a counter policy for the requested attributes or directly send such attributes. However, a $I(pol)$ message cannot be sent since, by hypothesis, the number of messages to be exchanged is set to k. In the latter case, all the requested attributes are disclosed and the thesis holds. The only other potential reply is an *Attr_refuse*(*attr*) message, which causes a failure, leading to an absurd since it contradicts the assumptions.

3. **Trusting Strategy**.
   a. *Basis*. We start with $k = 3$, that is, one message for each negotiating party is exchanged. Let $\mathcal{R}$ be the resource requested from $\mathcal{RQ}$ to $\mathcal{CN}$.

By hypothesis, $\mathcal{RQ}$ and $\mathcal{CN}$ trust requirements can be satisfied. Thus, by definition 6.1, the adopted strategy needs to be compliant with the local settings of $\mathcal{RQ}$ and $\mathcal{CN}$. Assume $\mathcal{CN}$ trust requirements for R are defined in terms of a policy *pol* requiring a term T_1 to be satisfied. Thus, the action executed by $\mathcal{CN}$ is *Send*(*pol*). Further, since a trusting strategy is used, the *sugg* component conveys a suggestion for credential T_1, expressed in term T_2. We first assume that the value of the *op* component of *sugg* is "−." By hypothesis, parties are compatible, so there must exist in $\mathcal{RQ}$ profile a credential *cred* satisfying term T_1. Further, there must exist in $\mathcal{RQ}$ policy base a policy having a rule of the form $cred \leftarrow T_2$. Thus, the only possible reply action is *Accept*($p.sugg$). Any different action is in contradiction with our hypothesis. Then, $\mathcal{CN}$ checks whether it has further requirements on T_2. If not, a sequence of credentials can actually be found, composed by the credential satisfying T_2, *cred*, and the resource originally requested. Thus, the negotiation successful ends and the thesis holds. Let us now consider the case when $sugg.op =$ "+." According to our hypothesis, two different cases may hold. $\mathcal{RQ}$ may or may not have a rule of the form $cred \leftarrow T_2$. If $\mathcal{RQ}$ has such rule, then the proof is exactly the same as the one for sugg.op = "−." By contrast, if $\mathcal{RQ}$ does not have the suggested rule he/she will reply with a *Refuse*($p.sugg$) message. According to our hypothesis, negotiation has to end after three messages have been exchanged. Then, *cred* can be either delivery or it may not belong to $\mathcal{RQ}$. However, the latter case is in contradiction with our compatibility hypothesis. As such, the only possibility is that of $\mathcal{RQ}$ having *cred* regulated by a delivery policy, which leads to a successful end of the negotiation.

b. *Inductive step*. Suppose we have proved the thesis for negotiations requiring $k' < k$ messages, and let us prove the thesis for a negotiation with k messages. Let m be the $k - 1$ message shipped. Assume, without loss of generality, that m is sent by $\mathcal{CN}$ to $\mathcal{RQ}$. Upon the exchange of the kth message, two valid actions are possible. If the sent message was a valid reply action for a previous policy suggestion (that is, the message has the form *Accept*(*sugg*)), no further policies are adopted for the corresponding credential *cred* in PB_{rq} policy base, then all the trust requirements have been satisfied and the negotiation succeeds. By contrast, if $m =$ *Refuse*(*sugg*), two different cases are possible. If the sign associated with the suggestion component is "+," then $\mathcal{RQ}$ can ship a new policy (by the message *Send*(p)) and proceed with the negotiation process. Instead, if $sugg.op =$ "−," there are two further possibilities. Either *cred* is not belonging to $\mathcal{RQ}$ or *cred* is possessed by $\mathcal{RQ}$, but $\mathcal{RQ}$ adopts another policy different from the suggested one. In the former case, the negotiation fails, which contradicts our hypothesis and gives rise to an absurd. In the remaining cases, if the adopted policy is a delivery policy, then the negotiation succeeds, and the thesis holds. By contrast, if the policy governing *cred* is of a different form, then, according to Table I, it will not be accepted by $\mathcal{CN}$, resulting again in negotiation failure, which is in contradiction with our hypothesis.

Let us now assume that $m = Send(pol)$. Since a trusting strategy is used, the *sugg* component of *pol* conveys a suggestion. Using the same reasonings of the previous case, we can easily derive the potential valid counterpart actions and prove that for all of them the thesis holds.

4. **Mixed Strategy**
   a. *Basis*. Let us first suppose $k = 2$. Since the negotiation is carried out according to a mixed strategy, a single round can be performed using different strategies and thus several types of messages can be shipped. If the message is that of a standard strategy, then correctness is ensured by Theorem C.1. Similarly, if the message is that of a suspicious strategy then, by the results proved by case 1 of the proof, the thesis hold. If the adopted strategy is a strong suspicious strategy, correctness is ensured by case 2 of the proof.

   Let us now consider the case when the negotiation is carried on according to a trusting strategy. Then, the minimum number of messages needed to end the process are three and the thesis holds for the results of case 3 of the proof.

   b. *Inductive step*. Suppose we have proved the thesis for negotiations requiring $k' < k$ messages and let us prove the thesis for k. Let m be the $k - 1$ message shipped. Suppose, without loss of generality, that the message m is shipped from $\mathcal{RQ}$. If $\mathcal{CN}$ follows a standard strategy, then the correctness and completeness of the thesis follows from Theorem C.1. Similarly, if nodes are added following a suspicious strategy, completeness is proved by case 1 of the proof. If the adopted strategy is a strong suspicious strategy, correctness and completeness is ensured by case 3 of the proof. Finally, if a trusting strategy is adopted a further round can be required (length of the negotiation is $k' = k + 2$) and the thesis holds by the results obtained in case 3 of the proof.

REFERENCES

Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y. 2003. Implementing P3P using database technology. *19th International Conference on Data Engineering*. Bangalore, India.

Bertino, E., Ferrari, E., and Squicciarini, A. 2003. X-TNL—an XML based language for trust negotiations. *Fourth IEEE International Workshop on Policies for Distributed Systems and Networks*. Como, Italy.

Bertino, E., Ferrari, E., and Squicciarini, A. 2004a. Privacy preserving trust negotiations. *4th International Workshop on Privacy Enhancing Technologies*. Toronto, Canada.

Bertino, E., Ferrari, E., and Squicciarini, A. 2004b. Trust-*X*—a Peer to Peer Framework for Trust Establishment. *IEEE Trans. Knowl. Data Eng. 16*, 7, 827–842.

Bonatti, P. and Samarati, P. 2000. Regulating access services and information release on the Web. *7th ACM Conference on Computer and Communications Security*. Athens, Greece.

Bradshaw, R., Holt, J. E., and Seamons, K. E. 2004. Concealing complex policies with hidden credentials. In *CCS '04: Proceedings of the 11th ACM Conference on Computer and Communications Security*. ACM Press, New York. 146–157.

Brands, S. 2000. *Rethinking Public Key Infrastructure and Digital Credentials*. MIT Press, Cambridge, MA.

Camenisch, J. and Herreweghen, E. V. 2002. Design and implementation of the idemix anonymous credential system. In *CCS '02: Proceedings of the 9th ACM Conference on Computer and Communications Security*. ACM Press, New York. 21–30.

Chaum, D. 1985. Security without identification: transaction systems to make big brother obsolete. *Commununications of ACM 28*, 10, 1030–1044.

Clark, J. 1999. XSL transformations (XSLT). version 1.0 W3C recommendation. Available at: `http://www.w3.org/TR/xslt`.

Cranor, L., Langherinrigh, M., and Marchiori, M. 2002. A P3P preference exchange language 1.0 (APPEL1.0). W3C Working Draft.

Cranor, L., Langherinrigh, M., Marchiori, M., Presler-Marsall, M., and Reagle, J. 2003. P3P-the platform for privacy preferences, version 1.1. Available at: `http://www.w3.org/P3P/1.1/`.

Herzberg, A. and J. Mihaeli, E. A. 2000. Access control meets public key infrastructure, or: Assigning Roles to Strangers. *IEEE Symposium on Security and Privacy*. Oakland, CA.

Housley, R., Polk, W., Ford, W., and So, D. 2002. Internet X.509 public key infrastructure certificate and certificate revocation List (crl) profile. RFC 3280.

IBM. IBM Tivoli privacy wizard. Available at: `www.tivoli.resource_center/maximize/privacy/wizard_code.html`.

Jarvis, R. 2003. Selective disclosure of credential content during trust negotiation. Master of Science Thesis, Brigham Young University, Provo, UT.

JRC. 2002. JRC P3P resource centre. Available at: `http://p3p.jrc.it`.

Lee, A. J., Winslett, M., Basney, J., and Welch, V. 2006. Traust: A trust negotiation-based authorization service for open systems. In *SACMAT '06: Proceedings of the 11th ACM Symposium on Access Control Models and Technologies*. ACM Press, New York. 39–48.

Li, N., Du, W., and Boneh, D. 2003. Oblivious signature-based envelope.

Microsoft. 2004. Infocard project. Available at `http://msdn.microsoft.com/winfx/reference/infocard/default.aspx`.

Naor, M. 1990. Bit commitment using pseudorandomness. *Advances in Cryptology- 89*. Lecture Notes in Computer Science, vol. 435, New York.

Persiano, P. and Visconti, I. 2000. User privacy issues regarding certificates and the TLS protocol. *Proceedings of the ACM Conference on Computer and Communication Security*, Athens, Greece.

Seamons, K. E., Winslett, M., and Yu, T. 2001. Limiting the disclosure of Access Control Policies during automated trust negotiation. *Network and Distributed System Security Simposium*. San Diego, CA.

Seamons, K. E., Winslett, M., and Yu, T. 2002. Protecting privacy during on line trust negotiation. *2nd Workshop on Privacy Enhancing Technologies*. San Francisco, CA.

Westin, A. F. 1967. Privacy and freedom. Atheneum, New York.

Winsborough, W. and Li, N. 2002a. Towards practical automated trust negotiation. *IEEE 3rd Intl. Workshop on Policies for Distributed Systems and Networks*. Monterey, CA.

Winsborough, W. H. and Li, N. 2002b. Protecting sensitive attributes in automated trust negotiation. *ACM Workshop on Privacy in the Electronic Society*.

Winsborough, W. H., Seamons, K. E., and Jones, V. 2000. Automated trust negotiation. *DARPA Information Survivability Conference and Exposition*, Vol. I, 88–102.

Winslett, M., Yu, T., Seamons, K. E., Hess, A., Jarvis, J., Smith, B., and Yu, L. 2002. Negotiating trust on the Web. *IEEE Internet Computing, 6*, 6, 30–37.

World Wide Web Consortium. References for P3P implementation. Available at: `http://www.w3org/P3P/implementations`.

World Wide Web Consortium. Uniform resource identifiers, naming and addressing: URIs, URLs, ... Available at `http://www.w3.org/addressing`.

Yu, T. and Winslett, M. 2003. A unified scheme for resource protection in automated trust negotiation. *IEEE Symposium on Security and Privacy*, 110. Oakland, CA.

Yu, T., Winslett, M., and Seamons, K. E. 2003. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. *ACM Transactions on Information and System Security 6*, 1 (Feb.).

Received June 2004; revised October 2005, June 2006, and February 2007; accepted February 2007