Protecting Privacy During On-line Trust Negotiation

Kent E. Seamons[1], Marianne Winslett[2], Ting Yu[2],
Lina Yu[1], and Ryan Jarvis[1]

[1]Computer Science Department, Brigham Young University
Provo, UT, USA 84602
{seamons, lina, rjarvis}@cs.byu.edu

[2]Department of Computer Science, University of Illinois at Urbana-Champaign
Urbana, IL, USA 61801
{winslett, tingyu}@cs.uiuc.edu

Abstract. The dramatic growth of services and information on the Internet is accompanied by growing concerns over privacy. Trust negotiation is a new approach to establishing trust between strangers on the Internet through the bilateral exchange of digital credentials, the on-line analogue to the paper credentials people carry in their wallets today. When a credential contains sensitive information, its disclosure is governed by an access control policy that specifies credentials that must be received before the sensitive credential is disclosed. This paper identifies the privacy vulnerabilities present in on-line trust negotiation and the approaches that can be taken to eliminate or minimize them. The paper proposes modifications to negotiation strategies to help prevent the inadvertent disclosure of credential information during on-line trust negotiation for those credentials or credential attributes that have been designated as sensitive, private information.

1 Introduction

The dramatic growth of services and information on the Internet is accompanied by growing concerns over privacy from individuals, corporations, and government. Privacy is of grave concern to individuals and organizations operating in open systems like the Internet. According to Forrester Research, only six percent of North Americans have a high level of confidence in the way Internet sites manage personal information [5]. These concerns cause some to operate only under complete anonymity or to avoid going on-line altogether. Although anonymity may be appropriate and desirable for casual web browsing, it is not a satisfactory solution to the privacy issue while conducting sensitive business transactions over the Internet because sensitive transactions usually require the disclosure of sensitive personal information.

When a client and server initiate a transaction on the Internet, often they begin as strangers because they do not share the same security domain and the client lacks a local login. With automated trust establishment, strangers build trust by exchanging digital credentials, the on-line analogues of paper credentials that people carry in their wallets: digitally signed assertions by a credential issuer about the credential owner. A credential is signed using the issuer's private key and can be verified using the issuer's public key. A credential describes one or more attributes of the owner, using attribute name/value pairs to describe properties of the owner asserted by the issuer. Each credential also contains the public key of the credential owner. The owner can use the corresponding private key to answer challenges or otherwise demonstrate ownership of the credential. Digital credentials can be implemented using, for example, X.509 certificates [12].

In our approach to automated trust establishment, trust is established incrementally by exchanging credentials and requests for credentials, an iterative process known as *trust negotiation* [13][3]. A *trust negotiation strategy* controls the exact content of the messages exchanged during trust negotiation, i.e., which credentials to disclose, when to disclose them, and when to terminate a negotiation. A naive strategy for establishing trust is for a client to disclose all its credentials to an unfamiliar server whenever a client makes a request. This simple approach that completely disregards privacy is comparable to a customer entering a store for the first time, plopping down their wallet or purse on the counter, and inviting the merchant to rifle through the contents to determine whether or not to trust the customer. Another approach, one that considers credential sensitivity, is for each party to continually disclose every credential whose access control policy has been satisfied by credentials received from the other party. This eager approach results in needless credential disclosures, even though the other party is authorized to receive them. A third approach is for each party to disclose access control policies that focus the negotiation only on those credentials actually needed to advance the negotiation.

Access control policies (*policies*, for short) for local resources specify credentials that the other negotiation participant must provide in order to obtain access to the resource. A party receives credentials from the other participant and checks to see if the relevant local policies are satisfied. A party may also disclose local policies, which constitute implicit requests for credentials from the other party that will advance the negotiation toward the goal of granting access to the protected resource. The other participant consults a disclosed policy to determine if it can satisfy it and responds accordingly. The purpose of trust negotiation is to find a credential disclosure sequence (C_1, …,C_k, R), where R is the service or other resource to which access was originally requested, such that when credential C_i is disclosed, its access control policy has been satisfied by credentials disclosed by the other party.

As an example of a trust negotiation, suppose Alice is a landscape designer who wishes to order plants from Champaign Prairie Nursery (CPN). She fills out an order form on the web, checking an order for box to indicate that she wishes to be exempt from sales tax. Upon receipt of the order, CPN will want to see a valid credit card or Alice's account credential issued by CPN, and a current reseller's license. Alice has no account with CPN, but she does have a digital credit card. She is willing to show her reseller's license to anyone, but she will only show her credit card to members of the Better Business Bureau. A possible credential exchange sequence for the above example is shown in figure 1.

When a policy P contains sensitive information, then P itself requires protection in the form of a policy for access to P. For example, a client interacting with an

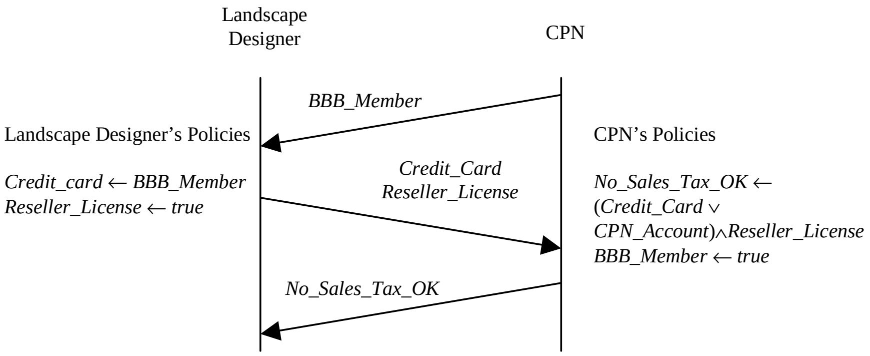

Fig. 1. An example of access control policies and a safe disclosure sequence that establishes trust between a server and a client.

unfamiliar web server may request to see credentials that attest to the server's handling of private information as well as certified security practices during the negotiation prior to disclosing further requests that the client considers sensitive, such as a policy that specifies the combination of credentials that the client requires for access to a business process that the client considers a trade secret. This situation requires that trust be established gradually, so that a policy referencing a sensitive credential or containing a sensitive constraint is not disclosed to a total stranger. Seamons et al. [9] introduced the notion of policy graphs, to represent the layers of access control policies used to guard a resource during gradual trust establishment. A policy graph for a protected resource *R* is a finite directed acyclic graph with a single source node and a single sink *R*. All the nodes except *R* represent policies that specify the properties that a negotiation participant may be required to demonstrate in order to gain access to *R*. Each sensitive credential and service will have its own separate policy graph. Each policy represented as a node in a policy graph *G* implicitly also has its own graph---the maximum subgraph of *G* for which that policy node is the sole sink. Intuitively, the resource represented by node *P* can be disclosed only if there exists a path from the source to *P* such that the disclosed credentials satisfy every policy along the path except possibly the policy at node *P*. Policy graphs allow a policy writer to gradually enforce sensitive constraints on credentials, without showing all the constraints to the other party at the beginning of the negotiation. Figure 2 shows an example of policy graphs and a policy path from source node P_0 to a policy node *P*.

Finally, there will be resources so confidential that the resource owner will not disclose the policy governing the resource under any circumstances. In this situation, only people who know the policy in advance and satisfy the policy proactively without being provided a clue can have access to the resource.

Policy disclosures have the potential for serious breaches in privacy that violate the basic safety guarantees that trust negotiation should provide. In this paper, we identify the ways that trust negotiation strategies fail to adequately protect credential privacy. We then suggest practical remedies that can help to safeguard privacy during trust negotiation and argue that these remedies preserve trust negotiation safety guarantees. This paper also describes the role of trust negotiation in automatically

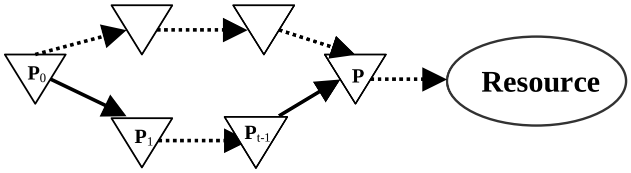

Fig. 2. An example policy graph. A solid line means the two nodes are adjacent to each other while a dashed line means there may be several nodes between the two policy nodes.

determining the privacy handling practices of servers, so that clients can automatically enforce their privacy preferences and rely on third parties for greater assurance of a server's privacy handling practices.

2 Privacy vulnerabilities during trust negotiation

This section presents four ways that privacy can be compromised during on-line trust negotiation. The first two compromises can occur with existing trust negotiation strategies that rely on policy disclosures to guide the negotiation. The third is related to attacks on the policy specification. Finally, the fourth relates to issues regarding the privacy practices of web servers.

2.1 Possession or non-possession of a sensitive credential

The trust relationships that individuals maintain are often considered sensitive. For instance, employers, health care providers, insurance companies, business suppliers, and financial institutions may maintain potentially sensitive trust relationships with individuals. These trusted institutions are often the issuers of digital credentials. The *type* of the credential can be a reflection of a trust relationship, such as an *IBM employee* credential or a *GM preferred supplier* credential. This can lead to the situation where the possession or non-possession of a certain type of credential is considered sensitive information. Simply revealing the type of the credential may release more information about a trusted relationship than the holder of the credential is comfortable disclosing. We call this kind of credential *possession-sensitive*.

Not all sensitive credentials will be possession-sensitive. For example, a generic *Employee* credential may not be possession-sensitive since the specific employer information is included as an attribute value in the credential, not as part of the credential type. In this case, the credential type conveys less information about the contents of the credential. A negotiation participant might have no qualms about a stranger knowing that they possess an *Employee* credential. However, the attribute values in the credential that identify the employer might be considered sensitive information.

Most requests for a credential will specify the desired type of the credential. As mentioned earlier, when a request for a sensitive credential *C* is received, a negotiation participant who possesses the credential typically responds with a counter-request for the credentials necessary to establish enough trust for *C* to be disclosed. This approach has potential privacy protection loopholes because the issuing of a counter-request can be interpreted as a strong indication of possession of the requested credential, and the absence of a counter-request as a strong indication of non-possession of the credential. Thus, the behavior of a negotiation participant may present clues regarding the possession or non-possession of a sensitive credential, even when the credential itself is never actually disclosed. In order to guard against the release of sensitive information when a possession-sensitive credential is requested during negotiation, its possessor's behavior must not allow the other party to infer whether or not they possess that credential.

2.2 Sensitive credential attributes

Besides credential type, other attributes in a credential can also contain sensitive information. For example, the issuer of the credential can convey a strong indication of the nature of the credential and the trust relationship that it represents. Other examples of sensitive credential attributes include age, social security number, salary, credit rating, etc. We refer to a credential with a sensitive attribute as an *attribute-sensitive credential*.

Trust negotiation may also violate privacy when Alice's credential contains a sensitive attribute value and Bob's credential request (disclosed policy) specifies a constraint on that sensitive attribute value. If Alice issues a counter request in response to Bob's credential request, Bob can infer that Alice owns a credential satisfying the constraint.

Policy disclosures can be used to probe to determine the exact value of a sensitive attribute without the credential ever being disclosed. For example, suppose that Alice considers her age to be sensitive, and the policy *x.type="drivers_license" and x.age>25* is disclosed to her.[1] Normally Alice would respond by disclosing the policy for her driver's license. If Bob is an adversary who is not qualified to gain access to Alice's driver's license, then Bob could engage Alice in a series of trust negotiations involving policy disclosures of the form *x.type="drivers_license" and x.age>24*, *x.type="drivers_license" and x.age>25*, *x.type= "drivers_license" and x.age>26*, etc. Once Alice fails to respond with a policy, Bob can infer that the credential no longer satisfies the constraint. Thus, Bob can determine Alice's age without seeing her driver's license.

A credential may contain multiple sensitive attributes. In some situations, a subset of the sensitive attributes in Alice's credential may be of interest to Bob. One

[1] In practice, the constraint on the credential's age attribute should be expressed as a comparison between the credential's date_of_birth attribute and a specific date. Also, we need to specify that the issuer of the credential is from a qualified government bureau, and that the other party should authenticate to the owner of the credential. We omit such detailed constraints in order to focus on the issue of attribute-sensitive and possession-sensitive credentials. We will make the same simplification in the rest of this paper.

interesting approach to protecting privacy is to selectively disclose attributes within a credential so that only the needed subset is made available to the recipient of the credential [4][7].

2.3 Extraneous information gathering

During trust negotiation, one of the parties may request credentials that are not absolutely necessary or relevant to the trust requirements of the transaction, even though they may be entitled to see them. A policy could require the disclosure of an inordinate amount of information during a trust negotiation, beyond what is truly needed to protect its resource. An unscrupulous party could use seemingly legitimate credential requests to gather extraneous information, violating the privacy of the other party.

Policies should be disclosed over a secure channel. Otherwise, an attacker can modify a policy during transmission to increase the number of required credentials, and force a participant to disclose more information than the requester intends. The attacker could eavesdrop on the subsequent exchange and gather the additional sensitive data, even if the bona fide negotiation participant pays no heed to the unexpected additional data. Even when a negotiation takes place over a secure channel, a similar threat exists for policy data stored on disk. If an attacker gains temporary access to the host managing the policy data, an attacker could modify the policy on disk before it is disclosed over the secure channel, potentially resulting in improper, excessive credential disclosures or in loss of the intended protection for local resources. In this scenario, the attacker would not gain access to the sensitive information in any subsequent secure negotiations, but the attacker effectively disrupts the security by causing improper disclosures that violate the original policy specifications.

In order to prevent malicious or accidental modification of policies, a policy can be digitally signed to protect its integrity. This will allow Bob to be able to detect any modification of Alice's policy that took place during transmission over an insecure channel or on disk prior to transmission, as long as Alice's private key used to sign the policy has not been compromised.

2.4 Privacy practices

Trust negotiation does not control or safeguard information once it has been disclosed. Thus sensitive information obtained during trust negotiation may be intentionally or unwittingly distributed to an unauthorized third party later on.

Internet users want control over what information they disclose to web sites and they want assurances about what those sites will do with the information once it is disclosed to them. However, control over personal information on line is handled in an ad hoc fashion today. In the United States, business is active through self-regulation to protect privacy. Companies such as TRUSTe [10] and BBBOnline offer privacy seals to a web site that publishes its policy for privacy handling practices and adheres to it. The TRUSTe privacy seal is an indication that the web site has a publicly available privacy policy to which it adheres. Human users can visually recognize the seal and can read the associated privacy policy. Perhaps many users never bother to read and understand the policy. The mere presence of the seal and privacy policy link may alleviate privacy concerns to the point that users are willing to interact with the server. However, the seal itself is susceptible to forgery and the seal's presence says nothing concerning the details of the web site's privacy policy. Interested users must wade through the fine print of legal-sounding documents on the web site to glean the details of the privacy handling policy, an arduous task that most users lack the stamina to complete. Further, many privacy policies include the provision that they are subject to change without notice, rendering their guarantees worthless. For more sensitive interactions, clients will need to *automatically* verify that Internet sites agree to follow certain procedures that meet the client's needs, or that Internet sites are certified to adhere to standardized, well-known privacy practices and procedures.

3 Privacy safeguards for trust negotiation

The previous section introduced privacy loopholes in trust negotiations that disclose access control policies. These shortcomings can be remedied by introducing stronger privacy controls during trust negotiation to prevent the inadvertent disclosure of sensitive credential information during a negotiation, as discussed in this section. Under one approach, policies governing credential disclosure are extended to support the designation of whether possession or non-possession of the credential type or any of the credential attribute values is considered sensitive information. When requests for credentials are received, the security agent must determine the nature of the requests and determine if any credential privacy protections are violated before responding to the request. Another approach is to dynamically modify policies to prevent leakage of sensitive information.

3.1 No response

Trust negotiation strategies traditionally ensure that all disclosures are *safe* [13]. The disclosure of a credential or policy is safe if the credentials disclosed so far by the other party satisfy the access control policy governing access to the credential or policy. When a possession-sensitive credential is requested during trust negotiation, responding to that request with a policy disclosure can amount to an admission of possession without actually disclosing the credential. Failure to respond with a counter request can amount to an admission of non-possession. In order to protect privacy, the definition of a safe disclosure must be extended.

When Alice receives Bob's policy *Q* that refers to a possession-sensitive credential *C*, then if Alice discloses the policy *P* for *C*, Bob may infer that she possesses *C*. If disclosing *C* at this point in the negotiation is not safe, then one possibility is to conclude that it is unsafe to disclose *P*. Then Bob will be unable to determine correctly whether Alice possesses *C*.

To add this feature to current trust negotiation systems, we can extend the model for policy graphs presented in [9] to support possession-sensitive credentials. Previously, the policy graph model assumed that the source policy node of a policy graph can be disclosed, and only internal policy nodes in the graph are sensitive. In the case of a possession-sensitive credential, an indicator can be associated with the source node of a policy graph to indicate whether or not the policy can be disclosed. This can be viewed as a generalization of the requisite/prerequisite rules of [3] to apply to policy graphs. For possession-sensitive credentials, no policies in the graph will be disclosed. Under this approach Alice will only disclose *C* if Bob pushes the necessary credentials to her without being prompted. The down side of this approach is that negotiations can fail to establish trust when it is theoretically possible to do so.

Extending policy graphs to support possession-sensitive credentials indirectly benefits Alice when she wants to disguise the fact that she does not possess certain credentials. Based on Alice's behavior, Bob is unable to distinguish whether Alice's failure to respond with a counter-request during trust negotiation is a sign that Alice does not possess the credential or a sign that she possesses the credential and is unwilling to acknowledge that fact because she considers the credential possession-sensitive.

3.2 Pretend to possess a credential

If Bob asks for credential *C* and Alice doesn't have *C*, some previously proposed trust negotiation strategies may require Alice to tell Bob that she does not possess *C*. This is a problem if Alice is sensitive about the fact that she does not have *C*. If we assume that *C* is sensitive and those that possess *C* are not opposed to having others believe that they possess it, then suppose Alice can create a policy governing access to *C*, even if she does not possess *C*. Whenever Alice receives a request for *C*, she discloses the same policy for *C* as those who possess *C*. Once a stranger demonstrates that they can be trusted with information regarding the credential *C*, Alice can send an admission that she does not possess *C*.

To support the ability to mislead strangers about whether one possesses a sensitive credential, we can extend the policy graph model to include an internal policy node in a policy graph with the policy *false*. For example, suppose that policy *P* governs access to *C*. Those that do not possess *C* can use a policy graph for *C* that is obtained as follows. Let *G* be the policy graph for *C* when a party possesses *C*. We can replace the sink node of *G* by *false*, add a new sink node representing *C*, add an edge from *false* to the new sink node, and let the resulting graph be *G'*. Then the graph for *C* when a party does not possess *C* should be *G'*. Whenever *C* is requested, the source node of *G'* is disclosed. Under this approach, Bob will have to pass the same set of trustworthiness tests, whether or not Alice possesses *C*, before he can determine whether Alice possesses *C*.

This approach safeguards non-possession of a sensitive credential from those who are not authorized to know that information. Once this approach is widely adopted, it effectively safeguards possession of a sensitive credential, since an attacker cannot be certain whether or not the other party in the negotiation possesses the credential, or is simply acting as though they possess it. This approach may only be practical for safeguarding possession-sensitive credentials when there are standard, reusable, widely-adopted policies for certain types of credentials that come as the standard equipment in trust negotiation packages.

3.3 Dynamic policy graphs

As described in the section on sensitive credential attributes, attribute-sensitive credentials are vulnerable to a probing attack that can expose the exact value of a sensitive attribute. One way to overcome this vulnerability with trust negotiation is by dynamically modifying policies before they are evaluated.

Because of the hierarchical nature of policy graphs and sharing of variables between policies, there may be several ways to specify the same set of constraints on credentials in a resource's policy graph. Some of these specifications may cause the other party to leak information about the credentials it possesses, even if the credentials are never disclosed. For example, suppose a customer places an order for a rental car at an on-line car rental service. The policy for such a service *R* states that the customer must show his/her driver's license to prove that his/her age is 25 or over. Figure 3 shows two ways to express this policy. The first version has a single policy node in *R*'s policy graph, which states $x.type{=}drivers\_license \wedge x.age \geq 25$. When the customer's security agent receives such a policy, it checks its driver's license credential. If the age attribute is over 25, it sends back the access control policy for the customer's driver's license as the next message. Otherwise, the negotiation fails. The problem is that if the on-line store receives the policy of the customer's driver's license credential, it can guess that the customer is over 25, even though the driver's license has not been disclosed. When the customer is sensitive about the age attribute, such information leakage is undesirable.

In the second policy graph in Figure 3, the source of *R*'s policy graph only contains the constraint on the credential's type attribute, namely, $x.type{=}drivers\_license$. The source has only one child P_2, which is $x.age \geq 25$, and P_2's child is the protected resource *R*. By adopting such a policy graph, when the client requests access to *R*, the source node of *R*'s policy graph is first sent back, which contains no constraints on sensitive attributes. Therefore, when the client returns its driver's license's policy in the next message, the server cannot infer any information about that credential's age attribute. The negotiation continues and only after the client actually discloses its driver's license can the server know the credential's age attribute and check whether the policy in the next level is satisfied. Thus no information about the driver's license's sensitive attributes is leaked.

One straightforward approach to preventing leakage is for Alice to act conservatively as defined in section 3.1 on no response, i.e., if Alice receives a policy *P* that involves constraints on sensitive attributes of a credential, and it is the first time that that credential variable appears in the path from the source to *P*, Alice simply assumes that the policy will never be satisfied. Thus no policy for that credential will be sent back. The obvious drawback is that a possible successful negotiation may fail because a party does not cooperate enough. As another approach, we may require that when a policy *P* is disclosed that contains a credential variable that does not occur in any predecessor of *P* along a path from the source to *P*, then *P* cannot contain

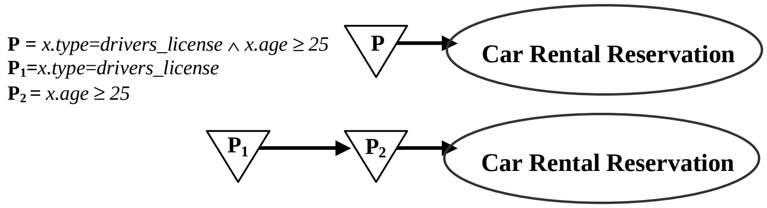
P = x.type=drivers_license ∧ x.age ≥ 25
P1=x.type=drivers_license
P2 = x.age ≥ 25
P
Car Rental Reservation
P1
P2
Car Rental Reservation

Fig. 3. Two different policy graphs that express the requirements for an on-line car rental reservation service**.**

constraints on sensitive attributes. If all the policy graphs satisfy this condition, as we show in the above example, attribute-sensitive credentials are protected. However, in practice, imposing such a requirement severely limits a policy writer's flexibility. For example, negotiation failure may seem mysterious because policy nodes that contain no new credential variables may never be disclosed. What's more, in most cases, a policy writer will not know which attributes of the other party's credentials are sensitive, making policy graph design extremely hard.

In fact, what makes the above two approaches impractical is the gap between the expectations of Bob, who designed the local policy graphs, and Alice, who tries to satisfy those policies. On one hand, Bob wants a policy to contain complete constraints on credentials so that the whole policy graph is simple and manageable. On the other hand, Alice wants to separate constraints on sensitive credentials from constraints on insensitive ones, which results in better protection in information of credentials. To fill this gap, we can dynamically modify the policy before Alice's strategy engine starts to try to satisfy it.

Dynamic policy modifications are performed by a policy transformation agent (PTA) inside each party's security agent (please refer to figure 4). When Alice sends a resource's access control policy *P* to Bob, before the policy goes to Bob's strategy engine, Bob's PTA first checks whether *P* involves any sensitive attributes of credentials. If so, the PTA transforms *P* into a hierarchical policy. That is, the PTA breaks *P* into a set of subpolicies whose conjunction entails *P*, and arranges the subpolicies into a sequence such that the first subpolicy in the sequence does not involve any sensitive attributes. After such a sequence is formed, only the first policy in the sequence is passed to Bob's strategy engine, which checks Bob's credentials and sends back policies for the relevant credentials. Because of the policy transformation done by the PTA, the strategy engine does not see the rest of the original policy that involves constraints on sensitive attributes. Therefore, its response does not imply that Bob's credentials can satisfy those sensitive constraints.

Let's look at our previous example. Suppose Alice sends *R*'s policy $x.type=drivers\_license \wedge x.age \geq 25$ to Bob. Upon receiving it, Bob's PTA checks the driver's license credential and finds the age attribute is sensitive. Then it decomposes the original policy into two subpolicies: $P_1 = x.type = drivers\_license$ and $P_2 = x.age \geq 25$. P_1 and P_2 form a sequence and P_1 is passed to the local strategy engine. In this case, even if the age given on Bob's driver's license is less than 25, the strategy engine will still send its policy in the next message. If Alice knows that Bob has a PTA, then Alice cannot infer anything about Bob's age from Bob's message.

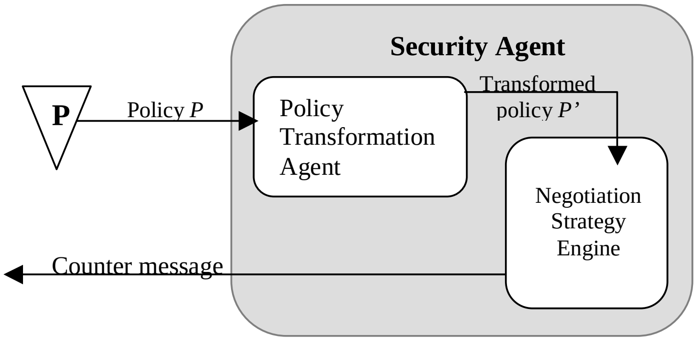

Fig. 4. Upon receiving a policy *P*, the security agent first lets the PTA check whether *P* involves sensitive attributes of credentials. If so, the PTA will transform *P* to *P,* and pass *P'* to the strategy engine. Based on *P'* instead of *P*, the strategy engine suggests the response which is sent back to the other negotiation party. Note that this figure only shows how the PTA works. We omit other necessary modules of a security agent, such as the credential verification module and policy compliance-checking module.

3.4 Privacy practices

Trust negotiation presents the opportunity to develop a more systematic approach to handling privacy practices on the web. Using trust negotiation, certified privacy practices can be represented in the form of digital credentials that can be disclosed in response to user policies that require certain privacy practice guarantees. In contrast to the web site privacy logos used today, the security agent on the user side can make sure that privacy practice credentials are not forged, and verify the ownership of such credentials using the public key encryption algorithm inherent in digital credentials. This can be done automatically without involving the user.

Representing privacy practices in credentials rather than in plaintext web documents satisfies several purposes. First, standardized privacy practice credential types will more easily support automated verification of privacy policies in software. Second, trust in the issuer that signs the credentials is a way to create stronger trust in the privacy practices of the organization. A third party that audits the sites' practices can issue credentials describing privacy practices.

When dealing with privacy issues, web sites typically offer some form of an opt-in or opt-out process. The client's security agent can record what has been opted in and opted out and request and receive a signed commitment from the server specifying the agreement on collected private data and on how the server will handle the client's private information. We call the signed commitment from the web site an *electronic receipt* that functions as a proof in case of dispute. [11]. The electronic receipt can also be signed by the client to protect the website from the client's repudiation.

Such a commitment can be referenced later should the client seek redress of some kind after it is detected that the server has broken its earlier commitment. This approach does not technically prevent the disclosure, but only provides assurance to the client with evidence to present during the recourse process. Technical solutions that we develop will require associated legal and social processes beyond the scope of this paper to be effective in mitigating privacy risks currently preventing certain on-line transactions

4 Related work

Recent work on trust negotiation strategies (e.g. [13]) includes proposals for policy disclosures. No previous work considers privacy vulnerabilities during trust negotiation. Previously proposed strategies can be strengthened with the recommendations from this paper in order to provide greater privacy protection guarantees.

Bonatti et al. [3] introduce a uniform framework and model to regulate service access and information release over the Internet. Their framework is composed of a language with formal semantics and a policy filtering mechanism that provides service renaming in order to protect privacy. Their work adopts a model very similar to the trust negotiation model presented in this paper.

X-Sec [1] is an XML-based language for specifying credentials and access control policies. X-Sec was designed to protect Web documents. Because credentials and policies can themselves be expressed as XML documents, they can be protected using the same mechanisms developed for the protection of Web documents. If X-Sec were used in trust negotiation, the privacy protection approaches presented in this paper would help to safeguard the contents of X-Sec credentials and policies.

Persiano et al. [7] introduce the SPSL protocol to extend TLS so that a portion of a certificate that Alice discloses to Bob can remain secret from Bob. This is desirable when a credential contains sensitive attributes that need not be disclosed in order to establish trust.

Brands [4] introduced techniques for designing Private Credentials and protocols for issuing and disclosing Private Credentials. One important property of Brands' design is its support for selective disclosure of a Private Credential's attributes, which will provide more privacy protection in trust negotiation. A party may only disclose those attributes of a credential that are actually relevant to a policy. Further, by adopting Private Credentials in trust negotiation, the granularity of credential disclosure access control can be extended to the attribute level, which will give users much more flexibility in expressing their security requirements.

Biskup describes two methods of preserving secrecy in a database system [2], namely *refusal* and *lying*. When an unauthorized user requests access to sensitive information, the refusal approach is to return a special value like *mum*, while the lying approach is to return the negation of the real answer. In Biskup’s work, the assumption is that the user knows that the requested information exists on the system yet does not know if that information is considered secret and protected by the system. *Secrecies known* is the situation when a user knows information on the system that is considered a secret; otherwise it is *secrecies unknown*. Refusal works for both the secrecies known and secrecies unknown cases, while lying only works under secrecies unknown. The assumption of knowledge of the existence of secret data on the system is similar to the situation with attribute-sensitive credentials. The dynamic policies approach is a type of refusal, at least for the constraint portion on the sensitive attribute. The failure to respond with a counter request for a possession-sensitive credential is also a form of refusal. Pretending to own a possession-sensitive credential that one does not possess is akin to the lying approach, at least temporarily.

The P3P standard [8] defined by W3C focuses on negotiating the disclosure of a user's sensitive private information based on the privacy practices of the server. Trust negotiation generalizes P3P to base disclosure on any server property of interest to the client that can be represented in a credential. The work on trust negotiation focuses on certified properties of the credential holder while P3P is based on data submitted by the client that are claims the client makes about itself. Support for both kinds of information in trust negotiation is warranted. Bonatti et al. also provide for both types of data in client and server portfolios in their system [3].

5 Conclusion and future work

This paper identifies the privacy vulnerabilities present in on-line trust negotiation, especially when access control policies are being disclosed. Although policy disclosure helps to focus the negotiation on those credentials that can lead to a successful negotiation, the approach is subject to serious lapses in privacy protection, and can inadvertently disclose evidence of possession or non-possession of a sensitive credential or inadvertently disclose the value of a sensitive credential attribute, without the credential ever being disclosed. Another privacy risk during trust negotiation is excessive gathering of information that is not germane to the transaction, due to unscrupulous negotiation participants or attacks on policy integrity. And finally, there is cause for concern regarding the handling of private information once it has been disclosed.

This paper identifies two kinds of sensitive credentials, a possession-sensitive credential and an attribute-sensitive credential, and identifies several potential means of preventing information leakage for these kinds of credentials. The paper proposes modifications to negotiation strategies to help prevent the inadvertent disclosure of credential information during on-line trust negotiation for those credentials or credential attributes that have been designated as sensitive, private information. The paper also describes how trust negotiation can be used to enforce the privacy preferences of clients by verifying a server's privacy handling policies and certifications automatically, potentially strengthening the current ad hoc process that services such as TRUSTe and BBBOnline employ for managing privacy practices on the Internet.

After a formal analysis of the privacy guarantees provided by these approaches, we will implement one or more of the approaches in TrustBuilder, our prototype environment for developing reusable trust negotiation components that run in a variety of contexts [6]. We have been experimenting with trust negotiation strategies that exchange policies in TrustBuilder. Since privacy vulnerabilities exist when policy disclosures take place, this will provide us with an excellent environment to experiment with our proposed solutions to the privacy problem.

Acknowledgements

This research was supported by DARPA through AFRL contract number F33615-01-C-0336 and through Space and Naval Warfare Systems Center San Diego (SSCSD) grant number N66001-01-18908.

References

1. Bertino, E., Castano, S., Ferrari, E.: On Specifying Security Policies for Web Documents with an XML-based Language, Proceedings of Sixth ACM Symposium on Access Control Models and Technologies, Chantilly, Virginia (2001).
2. Biskup, J.: For Unknown Secrecies Refusal is Better than Lying, Data & Knowledge Engineering 33 (2000), Elsevier Science, Amsterdam (2000).
3. Bonatti, P., Samarati, P.: Regulating Service Access and Information Release on the Web, Proceedings of the 7th Conference on Computer and Communications Security, Athens, Greece (2000).
4. Brands, S. A.: Rethinking Public Key Infrastructures and Digital Certificates, MIT Press, Cambridge, Massachusetts (2000).
5. Companies Must Adopt A Whole-View Approach To Privacy, According to Forrester Research, http://www.forrester.com/ER/Press/Release/0,1769,514,00.html (2001).
6. Hess, A., Jacobson, J., Mills, H., Wamsley, R., Seamons, K. E., Smith, B.: Advanced Client/Server Authentication in TLS, Network and Distributed System Security Symposium, San Diego, CA, (2002).
7. Persiano, P., Visconti, I.: User Privacy Issues Regarding Certificates and the TLS Protocol, in Proceedings of the 7th ACM Conference on Computer and Communications Security, Athens, Greece (2000).
8. Platform for Privacy Preferences (P3P) Specification, W3C Working Draft 26 August (1999), http://www.w3.org/TR/WD-P3P/Overview.html.
9. Seamons, K. E., Winslett, M., Yu, T.: Limiting the Disclosure of Access Control Policies During Automated Trust Negotiation, Symposium on Network and Distributed System Security, San Diego (2001).
10. TRUSTe, http://www.truste.org.
11. Tygar, J. D.: Atomicity versus Anonymity: Distributed Transactions for Electronic Commerce, Proceedings of 24th International Conference on Very Large Data Bases, New York City, New York (1998).
12. International Telecommunication Union, Recommendation X.509 - Information Technology - Open Systems Interconnection - The Directory: Authentication Framework (1997).
13. Yu, T., Winslett, M., Seamons, K. E.: Interoperable Strategies in Automated Trust Negotiation, Proceedings of the 8th ACM Conference on Computer and Communications Security, Philadelphia, Pennsylvania (2001).