Visualization of Automated Trust Negotiation

Danfeng Yao* Brown University
Michael Shin† Goldman Sachs, Inc.
Roberto Tamassia‡ Brown University
William H. Winsborough§ University of Texas, San Antonio

Abstract

We have designed an interactive visualization framework for the automated trust negotiation (ATN) protocol and we have implemented a prototype of the visualizer in Java. This framework provides capabilities to perform the interactive visualization of an ATN session, display credentials and policies, analyze the relations of negotiated components, and refine access control policies and negotiation strategies. We give examples of the visualization of ATN sessions and demonstrate the interactive features of the visualizer for the incremental construction of a trust target graph (TTG). Our prototype, which implements most components of the visualization framework, has played a key role in a research project that has developed working trust negotiation systems in an industrial environment.

Categories and Subject Descriptors: D.4.6 [**Operating System**]: Security and Protection; I.3.6 [**Computer Graphics**]: Methodology and Techniques–*Interactive Techniques*;

General Terms: Security

Keywords: Automated trust negotiation, interactive protocol visualization

1 Introduction

Interactive communication protocols for computer security can be complex. In automated trust negotiation (ATN) [22, 23, 25] for instance, there are many factors such as role and delegation credentials, privacy constraints and policies involved, whose interplay is not immediately comprehensible. The visualization of interactive protocols is an important role of security. However, there has not been sufficient attention on this topic from the research perspective. A secure protocol or system is only truly secure if people use it properly, and that requires the users to understand the system well and act accordingly.

Previous studies on the usability of security protocols such as PGP [20] show the fallacy of simple assumptions, such as assuming that a system or protocol *will* be used correctly if it *can* be used correctly. Therefore, we focus our attention on the visualization problem of interactive protocols, in particular the automated trust negotiation (ATN) protocol.

A significant amount of work has been done on the topic of *automated trust negotiation* (*ATN*) [15, 22, 23, 25]. ATN addresses the following web-service scenario. Suppose a member of a health insurance plan wants to go to a prescription website and receive the appropriate discount on the purchase of a drug. Currently, the most common procedure involves signing up for an account, generating a password, and most likely divulging a significant amount of private information (e.g., date of birth) that the site generally requires for receiving the discount. Automated trust negotiation aims to solve this privacy problem by designing negotiation protocols that operate on signed credentials that allow one to control the release of sensitive personal information.

Although ATN frameworks have been previously implemented [8, 15], there has not been much effort on the visualization of ATN to assist users in understanding complex ATN protocols. Winsborough and Li [21] illustrate that in current strategies for ATN, there are still potential flaws with different kinds of inferences that can sometimes be made by an adversarial party. Therefore, the ability to visualize ATN protocols is critical in trying to understand where such vulnerabilities might arise. The visualization architecture described in this paper aims to improve ATN's security and usability.

Building an ATN visualization framework where users can interactively participate in the protocol has several advantages:

- It gives teaching and learning support for ATN users to gain familiarity and experience with the protocol.
- It enables users to visually examine the ATN process, modify the policies of releasing private information, and fine-tune the negotiation parameters, policies, strategies, etc.
- The combination of interactive visualization and ATN improves the security of protected resources, as the visualizer can help human users to conduct analysis that detects flaws in policy specifications. This functionality is important for policy makers and administrators managing ATN. We note that this use of the ATN visualizer as a policy debugging tool involves some level of details, and is more suitable for users with a higher degree of familiarity with ATN.

The application in which the ATN visualizer has made its greatest impact to date is in the design of new ATN strategies. The state of a negotiation is a rich structure representing multiple forms of dependency (e.g., dependence of release of one credential upon seeing credentials of the opponent that authorize the disclosure, dependence of verifying that a negotiator has one attribute upon obtaining proof that he has other attributes, etc.). A negotiation can be thought of as a patchwork of many proofs by the negotiators to one another of their authorization for information and resources, including the proofs themselves. In a successful negotiation, these proofs add up to a composite proof that a resource requester is authorized for the desired resource. Different strategies explore different lines of completing such a composite proof. The ATN visualizer is essential to enabling a human to track this exploration. Without it, we found we

*e-mail: dyao@cs.brown.edu
†e-mail: mys@cs.brown.edu. Work by this author was performed while he was at Brown University.
‡e-mail: rt@cs.brown.edu
§e-mail: wwinsborough@acm.org. Work by this author was performed while he was at George Mason University.

had to draw more or less the same pictures for ourselves to successfully track the behavior of a negotiation strategy implementation. In addition to being an valuable aid to understanding the behavior of a correct implementation of a negotiation strategy, the visualizer routinely enabled us to understand easily bugs that would have otherwise consumed days to locate.

1.1 Contributions

We have designed an interactive visualization framework for the automated trust negotiation (ATN) protocol and we have implemented a prototype of the visualizer in Java. This framework allows users to perform the interactive visualization of an ATN session, display credentials and policies, analyze the relations of negotiated components, and refine access control policies and negotiation strategies. We show examples of the visualization of ATN sessions and demonstrate the interactive features of the visualizer for the incremental construction of a *trust target graph* (TTG) [22], which is a directed graph cooperatively constructed by two negotiators of an ATN session.

The data model of our visualization is state-oriented, that is, the negotiation process is captured by a sequence of negotiation states, each associated with a certain stage of the trust target graph. Our visualization framework uses colors and shapes to distinguish four types of nodes and six types of edges as needed by the TTG, where each of them represents a different negotiation target or has different requirements for being justified [22]. Negotiation targets can be informally thought as requests by one negotiator to see the proof that his opponent has some characteristics.

The architecture of the visualization framework includes components such as ATN engine, log parser, visualization unit for nodes and edges. The framework also contains interactive components that listen to the inputs from negotiators. The inputs are fed back into a modification component that updates the condition of the negotiation such as strategies, policies, and the state of the TTG.

Our proof-of-concept prototype implements most components of the visualization framework. It has been used in an industrial environment, where it has played a key role in the development of working trust negotiation systems. The visualizer takes in traces generated by a trust negotiation, which provide the visualized contents. Our visualization framework can handle arbitrarily large log data for display. We demonstrate the ability of displaying a complex ATN session in Section 5.3.

1.2 Outline

The paper is organized as follows. In the next section, we discuss related work. In Section 3, we present the architecture and data model for our ATN visualization framework. In Section 5, the trust target graph protocol is further described, and examples are given to illustrate our visualization framework. We give conclusions and outline future work in Section 6.

2 Related Work

The visualization of computations and protocols has been previously studied. There exists a rich body of work on algorithm animation for algorithms and data structures (see, e.g., [9]), programs (see, e.g., [16]) and datasets (see, e.g., [4]).

Visualization systems also exist for network protocols such as TCP [7, 26]. A detailed visualizer for showing network packet paths was built [27], where there is one active party involved. An interactive protocol visualization was presented by Papakosta and Burger [13], which introduces a language and model for the visualization of protocols such as the token-ring protocol and two-way authentication. The paper focuses more on modeling than on visual presentation and implementation.

Graph drawing [5, 17, 18] addresses the problem of constructing geometric representations of graphs, networks, and related combinatorial structures. Our ATN visualization system uses a graph drawing tool [2, 6] to construct drawings of the TTGs used in ATN negotiations sessions.

Previous work on visual specifications of policies and their verification [11] is related to our work of designing ATN visualizer for detecting flaws in negotiation policies and strategies. These two types of work are orthogonal to each other, and the existing tools for policy visualization can be a built-in part of our general ATN visualization framework.

There has been a growing body of research on automated trust negotiation frameworks [15, 23, 25, 22]. Recently, a Trust-Serv, which is a model-driven trust negotiation framework for web-services, is presented by Skogsrud, Benatallah and Casati [15]. The framework supports a visual interface for a trust negotiation modeler, which offers an editor for describing a state machine diagram of a negotiation policy. In comparison, our visualizer focuses on displaying the actual trust negotiation processes.

The concepts of combining visual and automated protocols for better performance in terms of flaw detection can be found in the work by Toeh *et al.* [19]. They described an integration of visual and automated data mining methods for discovering and investigating anomalies in Internet routing. They showed that interactive visualization can allow the user to examine the data and fine-tune the parameters for more accurate anomaly detection. Similar concepts can also be found in the work by Keim *et al.* [10] on the data-mining of large geo-spatial data sets.

3 ATN Visualization Framework

In this section, we describe the challenges associated with building a visualization tool for ATN and we present the architecture and data model of our ATN visualization framework. Our prototype implementation and a specific visualization example will be given in Section 5.

3.1 Requirements for ATN Visualization

Designing a visualization tool for complex interactive protocols such as ATN is challenging. The visualization system needs to be easy for users to understand and interact with. At the same time, it should display in detail the interplay of the credentials, policies, and strategies of the two negotiators. The visualization needs to handle a potentially large number of credentials and complex credential-protecting policies. The challenge is to be able to display the detailed information of an ATN process in a compact, yet simple fashion.

The visualization framework needs to use a data model that is extensible and flexible to support the interactive functionalities. The data model should be able to capture the transitions between one negotiation state to another, along with the associated side effects (exchanged messages,

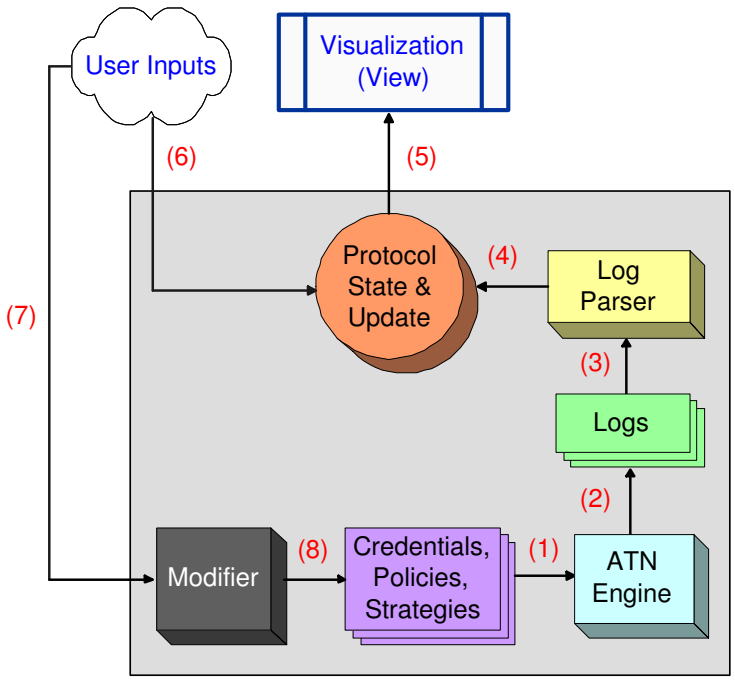

Figure 1: Components of our interactive ATN visualization framework. Numbers represent the data flow in the framework. (1) Information such as credentials, policies, negotiation strategies and parameters is entered into the negotiation engine. (2) Logs are produced by the negotiation engine. (3) The log data is given to the ATN log parser. (4) The log parser processes the ATN logs and creates a sequence of TTG states. (5) The negotiation state is visualized by an ATN visualizer. User's inputs modify the visualization state (6) or the negotiation information (7), which is done through the modifier component (8).

released credential released, etc) of the transitions. Conditions and parameters associated with a negotiation state should be dynamically modifiable by negotiators through an user interface. A negotiator should also be able to visually examine and explore potential negotiation paths.

From the reusable-code perspective, we want the architecture of the visualization framework to be composed of components that can be easily extendable to meet interactive ATN needs. We divide the framework into several components, each of them handling a task such as ATN engine, log parsing, visualizing ATN states, updating ATN states, etc. This decomposition allows for convenient extensions to support future needs for ATN interactive visualization.

These above design requirements are taken into consideration in our ATN visualization framework, which also provides additional graphical and functional features to assist users to better utilize the tool. We do not argue that the above requirements represent the totality of what a visualization system for ATN should embody. Rather, they are specific to a set of application domains that emphasize on the scalability and flexibility of a design.

3.2 Architecture and Data Model

In this section, we describe the design of an interactive ATN visualization framework. The architecture focuses on providing the information and interactive functionalities for negotiators in order to assist them in the analysis of real-time negotiation sessions or negotiation traces.

The components of the ATN visualization framework and how they interact with each other are shown in Figure 1. The ATN engine is given the negotiation information such as credential lists, access control policies that control the release of sensitive resources and credentials, negotiation strategies and parameters. The language model and protocol specifications are also entered into the negotiation system. The input information may be modified by user's inputs during a negotiation session.

Traces are generated by the negotiation engines and are written to the logs. The traces include the update messages sent between the two negotiators, credential lists, access control policies governing the release of sensitive resources and credentials. We will show in Section 5 how this information is displayed.

The log data is then entered into the ATN log parser, which processes the ATN logs into sequences of trust negotiation states. Each ATN state is represented by a set credentials, a set of credential/resource releasing policies or constraints, and messages exchanged by negotiators. One natural way to realize the ATN display is to use the trust target graph (TTG) protocol presented by Winsborough and Li [22], which is a directed graph and can be represented using nodes and edges. We use the *Grappa* system [2], a Java port of the *GraphViz* graph drawing system [6], to construct drawings of the TTGs used in ATN negotiations sessions. Our visualization prototype and visualization examples are presented in Section 5.

The information in the negotiation states are visualized by a visualizer. One interactive feature (indicated by (6) in Figure 1) that an ATN visualizer needs to support is the selective display of negotiation states such as step, play, pause, fast-forward, etc. This is especially important for the purposes of learning and static traces analysis. Therefore, the visualizer needs to capture the user inputs to roll backward or forward the negotiation states. This functionality is supported by our visualizer and it was proven useful for discov-

ering holes in security policies of ATN systems.

Another important interactive functionality of an ATN visualizer is the capability to allow negotiators to dynamically control and modify negotiation conditions. The combination of human participation and automated negotiation system can produce more accurate and fine-tuned negotiation results. In particular, a user should be able to dynamically specify new policies and constraints into the visualizer based on the current negotiation state, or change the negotiation strategies (for example from eager to parsimonious [23]). This information is processed by the modifier component (as in Figure 1), which then updates the ATN engine.

Here, we describe the data model of our ATN visualization framework. One of the prerequisites in constructing the visualizer is a modular design of data model, and thus state encapsulation is a necessity. This includes all defined policies, released credentials, and negotiation strategies in which the user has adopted. The use of trust target graph conveniently provides us a state-oriented data model for our visualizer. Using this format, a negotiator essentially keeps a log of one's view of the "universe" with respect to ATN objects.

There are three main components of an ATN state in our data model: node, edge, and exchanged message. At each step of the negotiation process, nodes and edges in the current TTG are updated based on the released credentials and policies. Types of nodes and edges are distinguished with different colors and object shapes in our visualizer. As negotiators cooperate through use of the ATN protocol in constructing a shared TTG, a copy of it is maintained by each negotiator. Therefore, another component of an ATN state is the messages that are exchanged between negotiators to synchronize the two TTG copies.

4 Trust Target Graph

In Section 5, we use a specific trust negotiation example to illustrate our visualization framework. The preliminary knowledge about the trust target graph and necessary notations needed in understanding our visualizer are given in this section.

In the trust-target graph protocol [22], a trust negotiation process involves the two negotiators working together to construct a trust-target graph (TTG). A TTG is a directed graph. Each node is either a trust target or a linking goal.

When a requester requests access to a resource, the access mediator and the requester enter into a negotiation process. The access mediator creates a TTG containing one target, which is called the *primary target* [22]. The access mediator then tries to process the primary target, and sends the partially processed TTG to the requester. In each following round, one negotiator receives from the other new information about changes to the TTG, plus credentials that help to justify those changes. It then verifies that the changes are legal, and updates its local copy of the TTG accordingly. The negotiator then tries to process some nodes, making its own changes to the graph, which it then sends to the other party, completing the round.

The negotiation succeeds when the primary target is satisfied; it fails when the primary target is failed, or when a round occurs in which neither negotiator changes the graph. For more details of TTG and issues therein, we refer readers to the paper by Winsborough and Li [22].

Our examples are based on the use of roles as in Role-Based Access Control systems [14], where a role is associated with certain access permissions and an individual user is mapped to a set of roles. An *entity* refers to either an organization or an individual. As in the TTG paper [22], a role r administered by entity A is denoted as $A.r$. Entity A is the administrator of role $A.r$. A role defines a group of entities who are members of this role. A role can be issued to an individual or to another role in a certain organization. For example, organization *ReliefNet* issues the role *ReliefNet.provisioner* to the role *purchasingA* in company *MedixFund* . (In our example, we inherit the names used in the trust negotiation literature [22].) This is written as:

$$ReliefNet.provisioner \leftarrow MedixFund.purchasingA$$

The above reads as a member in role *MedixFund.purchasingA* is also a member is role *ReliefNet.provisioner*. A trust target of TTG is represented using a similar but different notation. For example,

$$MedSup: \ \mathsf{MedSup.discount} \overset{?}{\leftarrow} \text{Alice}$$

This target states *MedSup*'s request for a proof that Alice satisfies the privilege MedSup.discount. This target representation will be used extensively next.

5 ATN Visualization Prototype

We demonstrate our trust negotiation visualizer in a typical negotiation scenario. Our visualizer can be run to display the trust target graph for one negotiator or for both negotiators. Screenshots of a two-sided visualizer are given in Figure 2 and 3.

Figure 2 visualizes the trust target graphs at two different stages of the negotiation between Alice and *MedSup*. The top screenshot shows the TTGs in the beginning of the negotiation. The **Negotiation Progress** bar indicates 16 percent of the negotiation process has been finished. The bottom screenshots shows the TTGs in the middle of the negotiation. The progress percentage is possible because the visualization is based on previous negotiation traces. The **Negotiation Progress** bar is located at the lower right of each window. Figure 3 shows the trust target graphs at two other stages of the negotiation. The top screenshot shows the TTGs when 77 percent of the negotiation process has been finished. The bottom screenshots shows the TTGs close to the end of the negotiation.

Within the visualizer, the upper boxes show the TTGs of the negotiators. For example, Alice is on the left and *MedSup* is on the right. Lower windows display the credentials, access control policies of both parties. Selecting **Local Credentials** tab gives the credentials held by the negotiator of that side. **Remote Credentials** tab gives the negotiation opponent's credentials or the opponent's policies governing its sensitive resources that have been disclosed so far. The **Policy** tab displays the access control policies governing the negotiator's sensitive credentials. The buttons at the bottom of the visualizer give the options of playing, pausing, fast-forwarding, and rewinding the negotiation process. The graphical properties such as the colors and shapes of nodes and edges can be customized.

5.1 Overview of Trust Negotiation Syntax

We illustrate the syntax used in the Trust Target Graph (TTG) visualizations based on a specific trust negotiation example between Alice and a fictitious company *MedSup*. A

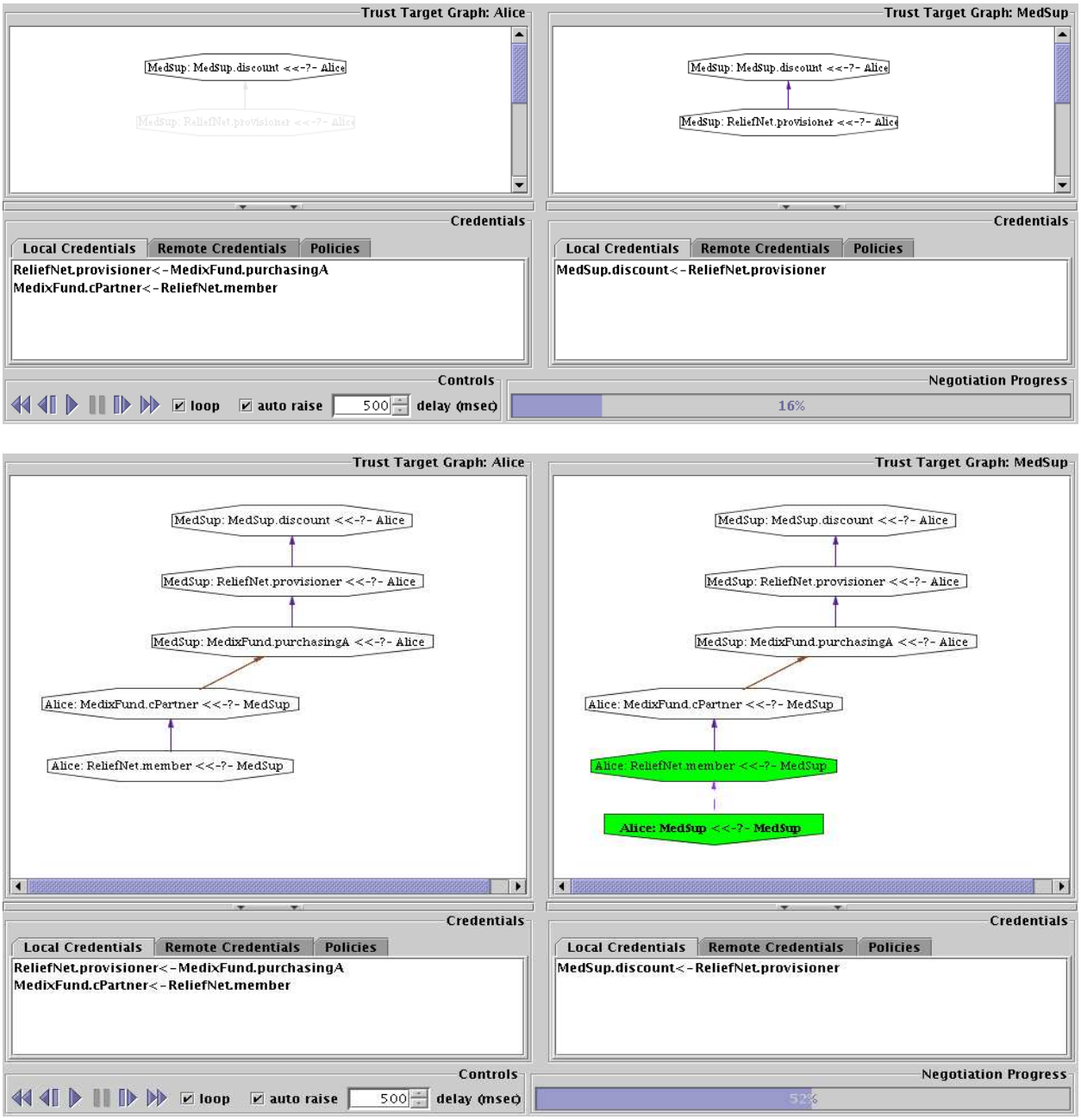

Figure 2: Visualizations of the trust target graphs at two different stages of the negotiation between Alice and $MedSup$. The top screenshot shows the TTGs in the beginning of the negotiation. The **Negotiation Progress** bar indicates 16 percent of the negotiation process has been finished. The bottom screenshots shows the TTGs in the middle of the negotiation. The progress percentage is possible because the visualization is based on previous negotiation traces. The **Negotiation Progress** bar is located at the lower right of each window. See Section 5.2 for more descriptions on the meanings of TTGs.

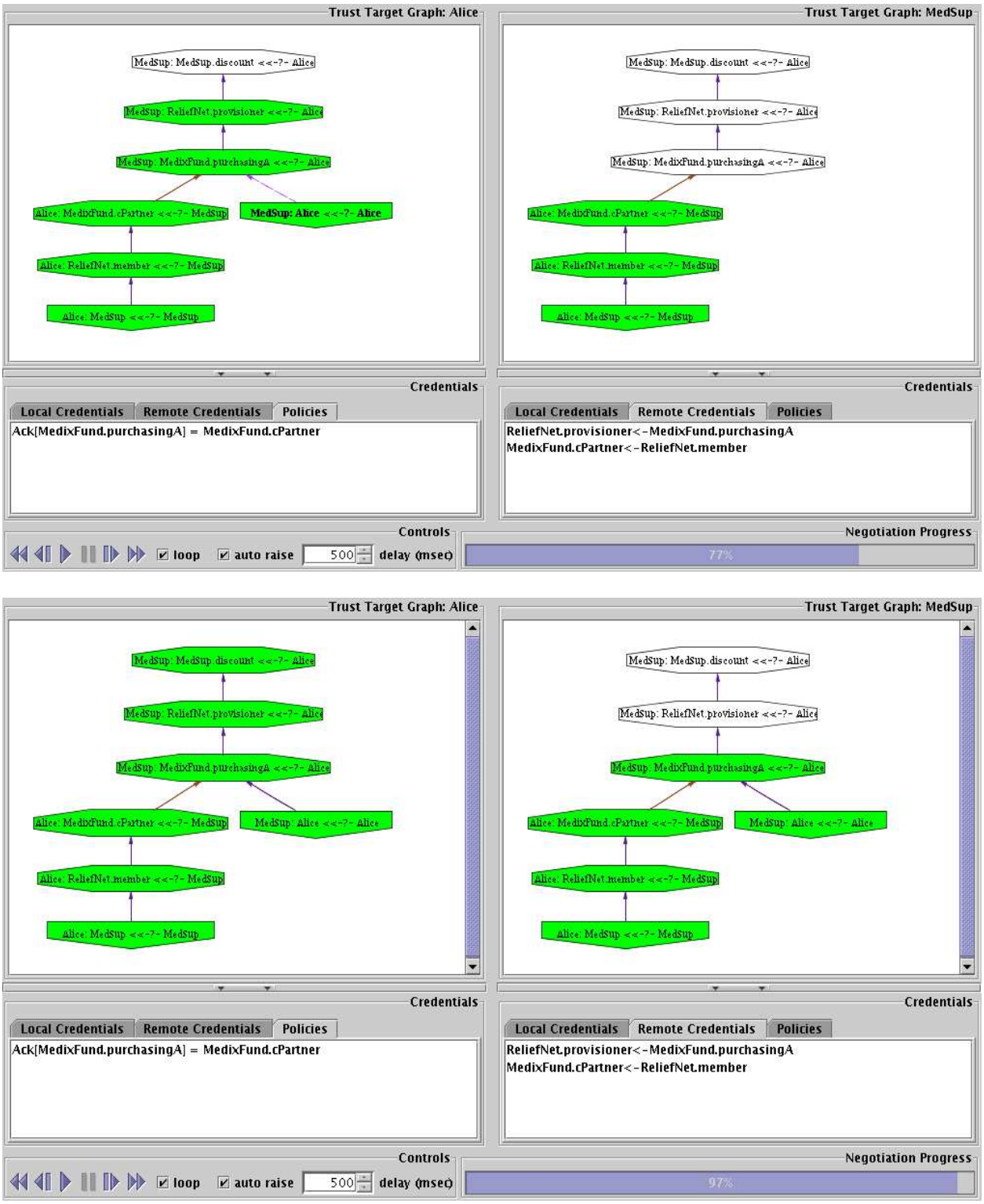

Figure 3: Visualizations of the trust target graphs at two different stages of the negotiation between Alice and $MedSup$. The top screenshot shows the TTGs when 77 percent of the negotiation process has been finished. The bottom screenshots shows the TTGs close to the end of the negotiation. The **Negotiation Progress** bar is located at the lower right of each window. See Section 5.2 for more descriptions on the meanings of TTGs.

Node name	Shape	Meaning
Trivial target	Inverted house	A target that is always satisfied; usually represents the authentication of an entity (Alice is indeed Alice).
Standard target	Octagon	Represents that a negotiator wants to see the proof of a certain subject having a certain role.
Linking goal	Rectangle	Represents that a verifier wants to see the proof of a subject having an attribute, where the authority of attribute is undetermined. For example, any valid student ID is acceptable.
Linked role target	Trapezium	Represents that a verifier wants to see the proof of a subject having a certain role of any organization that is certified by a certain authority (See more explanation in Section 5.3).
Intersection target	House	Represents that a negotiator wants to see the proof of a certain subject having an intersection of multiple roles.

Table 1: Shapes and meanings of nodes used in TTG.

similar example is used in [22]. In this example, *MedSup* authorizes a discount privilege MedSup.discount to some entities, and Alice wants to request for this discount. The goal is to find out whether Alice is qualified for MedSup.discount or not, based on the credentials and policies held by Alice and *MedSup*. Alice is a member of role *MedixFund.purchasingA*, and has two local credentials:

1. Organization *ReliefNet* issues the role *ReliefNet.provisioner* to the role *MedixFund.purchasingA*:

$$ReliefNet.provisioner \leftarrow MedixFund.purchasingA$$

2. The role *MedixFund.cPartner* is issued to the role *ReliefNet.member*.

$$MedixFund.cPartner \leftarrow ReliefNet.member$$

Notice that having a local credential does not necessarily mean that the owner is the issuer or the subject. Alice considers her role *MedixFund.purchasingA* sensitive. She has an acknowledgement policy, which states that the role *MedixFund.purchasingA* can only be released to entities with role *MedixFund.cPartner*. *Acknowledgement policy*, which was first defined in [22], can be viewed as a policy controlling the release of a sensitive credential. In addition, it provides extra privacy protection for the credential holder against malicious queries. In general, *MedSup* would also consider some attributes and credentials to be sensitive. However, here we assume that it does not.

MedSup authorizes the discount privilege MedSup.discount to those with the role *ReliefNet.provisioner*. Notice that *MedSup* is the issuer of this credential, and this credential is stored locally at *MedSup*. This credential can also be thought as an access control policy governing the discount privilege. *MedSup* is a valid member of *ReliefNet*, therefore has the role *ReliefNet.member*.

When Alice requests a discounted sale from *MedSup*, *MedSup* responds with the access control policy for MedSup.discount. This response is a trust target [22]:

$$MedSup: \text{ MedSup discount} \overset{?}{\leftarrow} \text{Alice}$$

In this example, the target states *MedSup*'s request for a proof that Alice satisfies MedSup.discount. This is the primary target, because satisfying it is the central goal of negotiation.

5.2 Negotiation Visualizer

In this section, the negotiation process between Alice and *MedSup* is described, and how the trust target graphs are constructed is shown in details. Screenshots of TTGs for both negotiators at different negotiation stages are shown in Figure 2 and 3. Table 1 and 2 show the shapes and colors of different types of nodes used in the visualizer. For example, a satisfied standard target is a green octagon. Some of the nodes are not used in the Figure 2 and 3, but are used later in Figure 4. The visualizer also uses different colors and styles for various types of edges in TTG, the details of which are omitted in this paper.

Node type	Color
Default	Purple
Satisfied node	Green
Failed node	Red
Unknown node	White

Table 2: Colors of nodes used in TTG.

1. Alice, who is a member of role *MedixFund.purchasingA*, requests for discount MedSup.discount from *MedSup*. The primary target is entered to the TTGs of both Alice and *MedSup*, and is the top node in Figure 3. The shape of standard targets of TTG is octagon in our visualizer, as shown in Figure 2.

$$MedSup: \text{ MedSup discount} \overset{?}{\leftarrow} \text{Alice}$$

2. The credential of *MedSup* associated with MedSup.discount is consulted. It has a local credential that gives the privilege MedSup.discount to the role *ReliefNet.provisioner*. This is shown in the **Local Credential** tab in the upper right window of Figure 3. A standard target node reflecting this credential is added to the TTG of *MedSup*, with a purple edge which is of type standard implication edge [22].

$$MedSup: ReliefNet.provisioner \overset{?}{\leftarrow} \text{Alice}$$

MedSup informs Alice that the role *ReliefNet.provisioner* is qualified for the privilege MedSup.discount. In other words, Alice needs to prove that she is a valid member of *ReliefNet.provisioner* to obtain the discount. Alice's TTG is updated with this information, i.e. the target node above is inserted, as shown in the bottom screenshot of Figure 2.

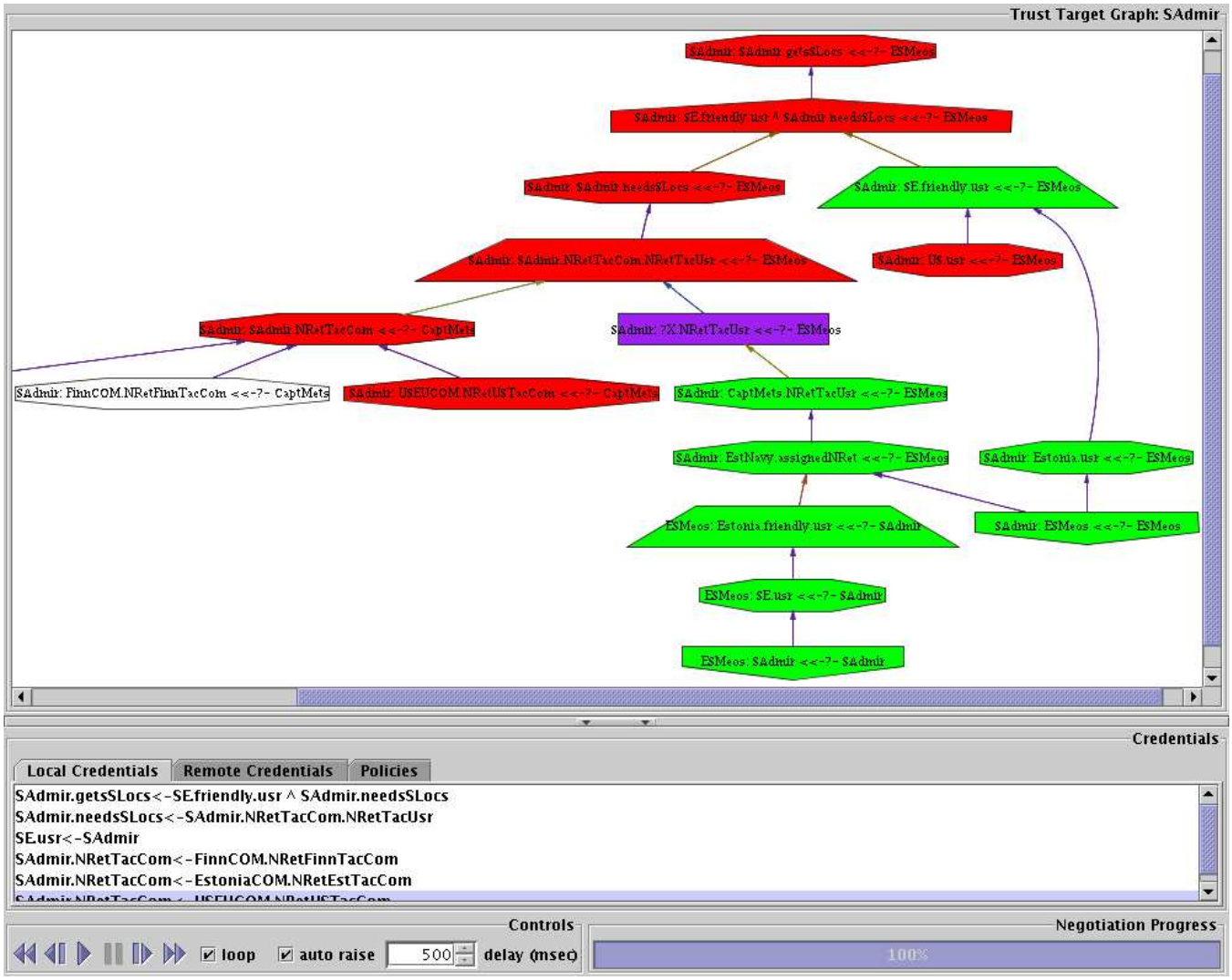

Figure 4: An example of TTG visualization for a complex ATN session. The figure shows the negotiation from the perspective of *SAdmir*, which is the resource owner. The resource requester is *ESMeos*.

3. Alice is a member of *MedixFund.purchasingA* and together with her credential (1) (shown in the **Local Credentials** tab), she can prove that she is a member of role *ReliefNet.provisioner*. Recall that in credential (1) the role *ReliefNet.provisioner* is delegated to members of role *MedixFund.purchasingA*.

4. However, Alice considers her role *MedixFund.purchasingA* sensitive and cannot just release it to *MedSup*. Alice's acknowledgement policy is consulted, which indicates that releasing the role *MedixFund.purchasingA* requires the proof of *MedixFund.cPartner* membership. This acknowledgement policy is shown in the **Policy** tab in the lower left window of Figure 3, where **Ack** is short for acknowledgement policy:

   Ack[MedixFund.purchasingA] = MedixFund.cPartner

   A target node representing this requirement is added to Alice's TTG with a brown edge, which is of the type linking solution edge [22].

   $$Alice : MedixFund.cPartner \overset{?}{\leftarrow} MedSup$$

5. Alice's credential (2) (shown in the **Local Credentials** tab) further indicates that *ReliefNet.member* is a valid member of *MedixFund.cPartner*. Therefore, equivalent to showing a *MedixFund.cPartner* credential, *MedSup* can prove its membership of *ReliefNet.member* in order to satisfy Alice's acknowledgement policy. A standard target node is added to Alice's TTG.

   $$Alice : ReliefNet.member \overset{?}{\leftarrow} MedSup$$

6. The request for *ReliefNet.member* or *MedixFund.cPartner* is sent to *MedSup*. The TTG of *MedSup* is updated accordingly with new target nodes. *MedSup* has the role *ReliefNet.member*, and therefore can satisfy Alice's request. This is represented in TTG

as inserting an edge and a trivial target [22] with shape of an inverted house, which is the bottom node of *MedSup*'s window in Figure 2.

$$Alice : MedSup \overset{?}{\twoheadleftarrow} MedSup$$

The color of satisfied targets is changed from white to green in our visualizer.

7. Upon receiving the proof of *ReliefNet.member* from *MedSup*, Alice's TTG is updated to be in sync with the TTG of *MedSup*. Her role *MedixFund.purchasingA* is released. Similarly, the TTG is updated as inserting an edge and a trivial target for Alice (of a shape of an inverted house), which connects to the target representing *MedixFund.purchasingA* (see Figure 3). Satisfied targets are changed from color white to color green.

   Because Alice proves that she has role *ReliefNet.provisioner* and is qualified for MedSup.discount, the second-to-top target node of her window in Figure 3 is colored green. These updates are sent over to *MedSup* so that its TTG can be updated and all the target nodes are satisfied. The negotiation succeeds.

The visualizer gives users the ability to flexibly control the display of a negotiation process. Its easy-to-understand user interface design makes it possible for all levels of users to utilize the tool. The capability of providing dynamic analysis and modifications of negotiation conditions such as strategies and policies based on user inputs from the visualizer is currently not supported by our prototype implementation, and is being investigated.

5.3 Complex Trust Target Graph

Our implementation of the visualization framework played a key role in a research project that developed working trust negotiation systems in an industrial environment. It supports the display of trust target graphs from complex trust negotiation scenarios and where the number of credentials and policies involved is large.

Figure 4 shows the visualization of an unsuccessful ATN session with a relatively large number of nodes and edges. It displays the TTG from the perspective of the resource owner *SAdmir*. The other negotiation party is called *ESMeos*, who is the resource requester. Figure 4 has several types of nodes that are not used in our previous ATN example. Nodes of shape trapezium represent *linked role* target and the rectangular shape node represents a linking goal [22].

A linking goal is like a target in which the attribute authority is undetermined. Linked roles are essential when there are many potential issuers of acceptable credentials. For instance, if university students are to receive a subscription discount, the policy should accept a student ID from any university. How are universities identified in the policy? It is unreasonable for the policy author to enumerate legitimate universities; this is not his expertise. Instead, an accrediting credential from a known accrediting board can be used in conjunction with the student's ID to satisfy a policy such as any student coming from an accredited university is allowed to access a database.

We refer readers to the TTG literature [22] for a complete description of the definitions of different TTG nodes and edges.

6 Conclusions and Future Work

We have described the architecture and data model of an interactive visualization framework for the automated trust negotiation protocol (ATN). A prototype of our ATN visualization framework has been implemented in Java and presented using an ATN example. Our ATN visualizer is not only useful in learning environments, but can also be used for conducting security analysis of policy and ATN protocol specifications. We also demonstrated that our ATN visualizer is capable of handling complex trust negotiation scenarios.

This work is the first step towards an integrated visualization framework for interactive protocols such as ATN. Future work will be focusing on bringing more interactive components into the implementation, such as visualizations for potential negotiation paths and strategies, and for connections between policies and released credentials.

We would also like to investigate the visualization for WSPL (Web Services Policy Language) [1, 24], which is a protocol for service policies that are implemented with eXtensible Access Control Markup Language [12]. WSPL attempts to provide an automated generation of service parameters for a given web service. Preliminary studies [3] on the WSPL visualization is limited to the visualization of one-round combinations of service parameters. It would be interesting to introduce multi-round negotiations to the WSPL model and support the visualization of complex web-service negotiation scenarios.

Acknowledgements

We would like to thank Seth Proctor for his comments on the design of interactive protocol visualization. We are grateful to Sean Cannella for sharing his insights on the model of a general-purpose protocol visualizer.

This work was supported in part by NSF grants CCF–0311510, IIS–0324846, CNS–0303577 and CNS-0325951.

References

[1] A. H. Anderson. An introduction to the Web Services Policy Language (WSPL). In *POLICY '04*, pages 189–192, 2004.

[2] N. S. Barghouti, J. Mocenigo, and W. Lee. Grappa: A Java graph package. In *Fifth International Symposium on Graph Drawing*, pages 336–343. Springer-Verlag, 1997.

[3] S. A. Cannella. Interactive protocol visualization (and a wspl case study). `www.cs.brown.edu/people/scannell/wsplv/ipvis.pdf`.

[4] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in Information Visualization: Using Vision To Think*. Morgan Kaufmann, San Francisco, California, 1999.

[5] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.

[6] J. Ellson, E. R. Gansner, L. Koutsofios, S. C. North, and G. Woodhull. Graphviz and dynagraph - static and dynamic graph drawing tools. *Graph Drawing Software*, 2003.

[7] J. Hall, A. Moore, I. Pratt, and I. Leslie. Multi-protocol visualization: a tool demonstration. In *Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research*, pages 13–22. ACM Press, 2003.

[8] A. Hess, J. Jacobson, H. Mills, R. Wamsley, K. E. Seamons, and B. Smith. Advanced client/server authentication in TLS. In *NDSS*, 2002.

[9] C. Hundhausen, D. Sarah, and J. Stasko. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing*, 13(3):259–290, 2002.

[10] D. Keim, C. Panse, M. Sips, and S. North. Pixel based visual mining of geo-spatial data. *Computers and Graphics*, 28(3):327–344, 2004.

[11] M. Koch and F. Parisi-Presicce. Visual specifications of policies and their verification. In *FASE 2003*, pages 278–293, 2003.

[12] M. Lorch, S. Proctor, R. Lepro, D. Kafura, and S. Shahi. First experiences using XACML for access control in distributed systems. In *Proceedings of the ACM Workshop on XML Security 2003*, pages 25–37, October 2003.

[13] S. Papakosta and C. Burger. Generating interactive protocol simulations and visualizations for learning environments. In *4th Plenary Workshop for Cabernet Members*, 2001.

[14] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29, Number 2:38–47, 1996.

[15] H. Skogsrud, B. Benatallah, and F. Casati. Trust-serv: model-driven lifecycle management of trust negotiation policies for web services. In *Proceedings of the 13th international conference on World Wide Web (WWW '04)*, pages 53–62, 2004.

[16] J. Stasko, J. Domingue, M. H. Brown, and B. A. Price, editors. *Software Visualization: Programming as a Multimedia Experience*. The MIT Press, Cambridge, Massachusetts, 1998.

[17] R. Tamassia, editor. *Handbook of Graph Drawing and Visualization*. CRC Press, 2006. To appear.

[18] R. Tamassia and G. Liotta. Graph drawing. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, second edition, 2004.

[19] S. T. Teoh, K. Zhang, S.-M. Tseng, K.-L. Ma, and S. F. Wu. Combining visual and automated data mining for near-real-time anomaly detection and analysis in bgp. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC-2004)*, pages 35 – 44, 2004.

[20] A. Whitten and J. Tygar. Why Johnny can't encrypt: a usability evaluation of PGP 5.0. In *8th Usenix security symposium*, pages 169–184, 1999.

[21] W. H. Winsborough and N. Li. Protecting sensitive attributes in automated trust negotiation. In *Proceeding of the ACM workshop on Privacy in the Electronic Society*, pages 41–51. ACM Press, 2002.

[22] W. H. Winsborough and N. Li. Towards practical automated trust negotiation. In *Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks (POLICY 2002)*, pages 92–103. IEEE Computer Society Press, June 2002.

[23] W. H. Winsborough, K. Seamons, and V. Jones. Negotiating disclosure of sensitive credentials. In *Second conference on security in communication networks*, September 1999.

[24] Web Services Policy Language (WSPL) implementation. `http://sourceforge.net/projects/openwspl/`.

[25] T. Yu, X. Ma, and M. Winslett. PRUNES: an efficient and complete strategy for automated trust negotiation over the internet. In *ACM Conference on Computer and Communications Security*, pages 210–219, 2000.

[26] C. Zhao and J. Mayo. A TCP/UDP protocol visualization tool: Visual TCP/UDP. In *International Conference on Engineering Education (ICEE '02)*, 2002.

[27] J. A. Zinky and F. M. White. Visualizing packet traces. In *Conference proceedings on Communications architectures & protocols*, pages 293–304. ACM Press, 1992.