Trust-Based Secure Information Sharing Between Federal Government Agencies

Peng Liu
School of Information Sciences and Technology, The Pennsylvania State University, 003B Thomas Building, University Park, PA 16802. E-mail: pliu@ist.psu.edu

Amit Chetal
Department of Computer Science and Engineering, The Pennsylvania State University, 220 Pond Laboratory, University Park, PA 16802. E-mail: chetal@cse.psu.edu

The September 11 attack and the following investigations show that there is a serious information sharing problem among the relevant federal government agencies, and the problem can cause substantial deficiencies in terrorism attack detection. In this paper we provide a systematic analysis of the causes of this problem; and conclude that existing secure information sharing technologies and protocols cannot provide enough incentives for government agencies to share information with each other without worrying that their own interests can be jeopardized. Although trust-based information access is well studied in the literature, the existing trust models, which are based on certified attributes, cannot support effective information sharing among government agencies, which requires an interest-based trust model. To solve this information sharing problem, we propose an innovative interest-based trust model and a novel information sharing protocol, where a family of information sharing policies are integrated, and information exchange and trust negotiation are interleaved with and interdependent upon each other. In addition, an implementation of this protocol is presented using the emerging technology of XML Web Services. The implementation is totally compatible with the Federal Enterprise Architecture reference models and can be directly integrated into existing E-Government systems.

Introduction

After the terrorist attacks of September 11th on American soil, much of the nation spoke out for a drastic change within the federal government. One of the primary changes focused on addressing information sharing among government agencies. With efficient information sharing, government agents will be able to predict and possibly preempt attacks form occurring. However, prior to September 11th, serious information sharing problems existed within government agencies.

Information Sharing Problems Among Government Agencies

To start, agencies acted as stovepipes, or rigid functionally organized departments (Ulanoff, 2002). For instance, there were multiple watch lists that existed across the federal agencies with regard to suspected malicious terrorists. Each agency had a separate watch list based on separate data organization and schema, but none of the watch lists had details about the other and the watch lists were not unified by any means. Moreover, there was a lack of communication between agencies. In fact, even President Bush conceded, "In terms of whether or not the FBI and the CIA were communicating properly, I think it is clear that they weren't" (Donovan, 2002, p. 1). So, the government agencies could not aggregate their information properly.

Development of E-Government Initiatives

Because of the stovepipes that existed between several government agencies, the president and the rest of the congressional staff developed 24 E-Government initiatives (Federal Enterprise Architecture Program Management Office, 2002e). These initiatives could generate several billion dollars in savings and specify plans so agencies can collaborate and share their information in a more efficient manner, thereby consolidating information.

Establishing the Federal Enterprise Architecture

To address and fulfill the E-Government initiatives, there needs to be a flexible, comprehensible, standardized (reference) model. This model is known as the Federal

Accepted February 26, 2004

 ● Published online 2 December 2004 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/asi.20117

Enterprise Architecture (FEA; Federal CIO Council, 2002). It is vitally important for agencies to communicate in a more citizen-centric, unifying way, where information can be exchanged in a more efficient manner. As a result, this new FEA will leverage technology investments to avoid unnecessary duplication of infrastructure among agencies, link business processes through shared, yet protected information systems, and leverage disparate business processes (April & Fonseca, 2002). It is clear that enabling *effective* information sharing is a major goal of the FEA. (In effective information sharing, the information held by one agency, if useful to another agency, will be timely disclosed to the other agency in a cooperative, proactive manner in most, if not all, cases.) In fact, one of the most intriguing E-Government initiatives is the *E-Authentication initiative*, whose objective is: "Build and enable the mutual trust needed to support widespread use of electronic interactions between the public and government and across governments . . ." (Federal Enterprise Architecture Program Management Office, 2002e, p. 7).

Limitations of Existing Information Sharing Technologies

However, existing information technologies are very limited to achieve the goal of the E-Authentication initiative, i.e., building mutual trust between agencies in such a way that effective information sharing can be enabled, since they cannot build the level of trust that is needed to provide enough incentives for government agencies to share information with each other without worrying that their own interests can be jeopardized. Although the FEA provides a solid, standardized platform for agencies to collaborate with each other, the FEA assumes the use of existing trust models, and agencies are still reluctant to share sensitive information with one another. Several government agencies declared that they were reluctant to share sensitive information with one another and are not comfortable since they don't entirely *trust* the other agencies. This study shows that although trust-based information access is well studied in the literature (Blaze, Feigenbaum, Ioannidis, & Keromytis, 1996, 1999; Chu, Feigenbaum, LaMacchia, Resnick, & Strauss, 1997; Clarke et al., 2001; Grosof, Feigenbaum, & Li, 2003), the existing trust models, which are based on certified attributes, cannot support effective information sharing among government agencies, which requires an *interest-based trust model*. Information sharing among government agencies requires a different, more restrictive trust model primarily due to two reasons: (a) much of the information that each agency carries is highly sensitive, so agencies find it difficult to simply let another agency gain access or observe its sensitive information without first developing enough trust with the other agency; (b) the differences and conflicts-of-interest existing between agencies require a more accountable and fair information sharing procedure, where the (mutual) trust is not only affected by certified attributes and cross-agency authority-mapping, but also affected by whether a win–win information sharing can be achieved without jeopardizing the interests of any agency involved.

Our Contributions

First, we perform a systematic analysis of the causes of the information sharing problem among government agencies, and identify the unique trust management requirements for effective information sharing among government agencies. Second, we propose an innovative interest-based trust model and a novel information sharing protocol, where a family of information sharing policies are integrated, and information exchange and trust negotiation are interleaved with and interdependent upon each other. Third, an implementation of this protocol is presented using the emerging technology of XML Web Services. The implementation is totally compatible with the FEA reference models and can be directly integrated into existing E-Government systems. We believe the proposed trust model and information sharing protocol may dramatically improve the effectiveness of information sharing among government agencies and reduce the deficiencies in countering terrorism attacks.

In the next section, we address the FEA and its limitations. We then explore what kinds of trust models are necessary to enable effective information sharing among government agencies. We proceed to survey the literature of information sharing and show why existing information sharing technologies cannot support effective information sharing among government agencies. In the following section our trust model and our information sharing protocol are presented, after which we present an XML Web Services based implantation of our information sharing protocol. Our conclusions are presented in the final section of the paper.

Federal Enterprise Architecture

In this section, we provide an overview of the FEA so that later in the paper we can show why the FEA is limited in enabling effective information sharing, and proceed to present a FEA-based implementation of our information sharing protocol.

The FEA exists as a function-oriented framework for describing the business operations of the federal government. It is also independent of the agencies that perform those operations (Federal Enterprise Architecture Program Management Office, 2002d).

Principles of Federal Enterprise Architecture

Initially, it is important to understand that there are several principles that form the basis for the federal enterprise architecture (Federal Enterprise Architecture Program Management Office, 2002d). For one, since much of the information stored by each agency requires proprietary software dependencies, there should be an establishment of standards, mainly the establishment of federal interoperability standards. By using the emerging new Web standards such as XML and XML Web services, agencies will be free from software and hardware dependencies. These new standards will allow for increased collaboration and cross-agency

information exchanging. Also, another vital principle involves capitalizing on standardization measures based on common functions between agencies. Federal agencies are attempting to develop reusable functions that exist across agencies and purchase architecture components that will allow for increased collaboration and eliminate redundancy (Federal CIO Council, 2002). Moreover, another principle revolves around coordinating the technology investments among the federal agencies. Since there is constant overlap of functions between the different agencies, vertical and horizontal integration will allow for a reduction in spending, causing a huge increase in federal financial savings.

Finally, a critical principle is to protect federal information against malicious attacks and unauthorized access. Since much of the federal information will be represented in a more collaborative manner through new Web standards such as XML and Web services (Birbeck at al., 2001), new security measures need to be taken. Mainly, sensitive information is no longer designated to one agency but rather is flowing among agencies, so sharing such sensitive information needs to be better protected because the damage caused by an attack can be dramatically magnified across multiple government agencies through the "bridges" built by the FEA.

Models of Federal Enterprise Architecture

The principles that make up the federal enterprise architecture clearly address the existing stovepipes that exist between agencies and address the issues behind the E-Government initiatives. However, to have a deeper understanding of the FEA, we need to discuss the four reference models of the FEA, namely *the business reference model*, the *data reference model*, the *application capability reference model*, and the *technical reference model* (Federal Enterprise Architecture Program Management Office, 2002b). The four models are connected in a hierarchical fashion in the sense that each lower level model is a detailed layout (or exploration) of typically one aspect of the corresponding higher level model. These models depict the process of how the federal enterprise architecture is used for cross-agency information sharing and collaboration.

Business reference model. First, the business reference model describes the lines of business of each agency, and identifies the customers and partners of each agency. The business reference model is a high level view of the federal enterprise architecture based on a hierarchical construct for describing the business operations of the federal government (Federal Enterprise Architecture Program Management Office, 2002b).

The basic procedure (to develop this reference model) involves looking at previous efforts that agencies used to identify their functions, and then once the list is formed in its entirety, categorize the functions into designated mutually exclusive business areas.

Data reference model. The data reference model shows how the components of the business reference model can be used to develop cross-agency information exchanges, and it provides a form of data standardization between the agencies. The data reference model will be used to describe the types of interactions and information exchanges that occur between the federal government and its various customers.

XML plays a critical role in developing the E-Government data reference model. XML provides a mechanism for federal lines of business to define and standardize XML schemas so some lines of businesses can interact with other lines of businesses. XML provides a standard way for preserving and communicating information encoding, tagging, and internationalizing information (Wolter, 2001). In XML, tags can be created to represent each function for a particular line of business. Also, these XML tags can be transmitted via HTTP or some other protocol so other agencies can then use these common functions, create new ones, or build upon them.

Application capability reference model. The application capability reference model describes how the application capabilities support the business objectives (Champion, 2002). The application capability reference model is comprised of two submodels: the *conceptual process model* and the *interoperability model*.

The conceptual process model describes the bridge between the functional view of the business reference model and the technology needed to carry out the cross-agency information exchanges. In order to retrieve information from an agency application, six layers of services (i.e., the end user layer, the access portal layer, the crosscutting requirements layer, the Web platform layer, the applications interface layer, and the enterprise data and applications layer) need to be developed so that the information exchanges can take place and each agency will be able to receive his/her desired information. The interoperability model describes the primary application components that support the conceptual process model and how they interoperate within and across the lines of businesses (Federal CIO Council, 2002). The interoperability occurs at the user, data, and application level.

Technical reference model. Finally, the technical reference model shows how technology is being used to deliver application capabilities. The technical reference model defines how the hardware, software, and physical location can support the businesses, data, and functions (Open Resource Group, 1999). The purpose of the technical reference model is for better coordination, development, and support of the E-Government initiatives and cross-agency information sharing.

The technical reference model has three layers, which include the application software layer, the application platform layer, and the external environment layer (Champion, 2002). These three levels generally represent how information can be exchanged in an agency from the end user, to accessing an agency's application or retrieve other relevant

information. The application software components consist of data, documentation, training, and programs. The application platform describes the collection of hardware and software components that provide services or software resources for the application software (Champion, 2002). The external environment consists of the system elements that are external to the application software and the application platform (Champion, 2002). Essentially, the technical reference model shows what measures will be used to implement the federal enterprise architecture.

Need for an Interest-Based Trust Model

In this section, before we show why FEA is limited in enabling effective information sharing among government agencies, we will first identify the agencies' specific requirements on the (mutual) trust model.

Why There Is a Lack of Trust Among Agencies

The primary reason that spurs the hesitation and reluctance for agencies to share their sensitive information with one another is that they simply don't *trust* each other. The lack of trust is due to several reasons. We believe a major reason is that *conflicts of interest* usually exists between agencies and as a result, having agency A share some information with agency B may actually compromise the interests of agency A. In fact, "the conflict between the agencies is deeply rooted. Even an event as horrible as [Sept 11] cannot erase five decades of turf battles" (French, 2003, p. 2). Agencies may experience a great deal of friction in cooperating in a joint effort, and information sharing usually plays an important role in these frictions. As a result, unfair information sharing and misuse of shared information can substantially *discourage* the agencies involved in a joint effort to share information with each other. To illustrate, when two agencies A and B are cooperating in a joint effort, if agency A shares a piece of information (with B) which is critical to a task assigned to B, but B hides a piece of information which is critical to a task assigned to A, then B can perform much better than A, and the performance difference may give B a lot of advantages in competing with A. Hence, to ensure that his/her interests will not be jeopardized, agency A may want to do the same thing as B does, that is, A will stop sharing information with B (proactively). In this way, as more agencies hide their information, there will be less and less mutual trust staying between agencies.

Moreover, the above problem can be further exacerbated by two facts: (a) there are a lot of misunderstandings between government agencies; (b) little *accountability* is provided in existing information sharing processes. Fact (a) indicates that agencies may possess substantial misunderstandings (or doubt) about one another in terms of the intent, objective, and strategies of information sharing due to some inherent differences in their agency structures, cultures, intra-agency policies, beliefs, responsibilities, and missions. For example, "the deeply ingrained differences between the CIA and FBI have long prevented them from working together . . ." (French, 2003, p. 2). The misunderstandings between agencies can make one agency "sense" more risk or less trust when sharing information with another agency, thus they can further discourage agencies to share information.

Fact (b) exacerbates the above problem since the lack of accountability makes it difficult to resolve the conflicts caused by unfair or misused information sharing among agencies. Since the "bad" player cannot be easily identified or published, one agency will probably "sense" more risk or less trust when sharing information with another agency.

Another reason for the lack of trust between agencies is that shared information may be improperly processed by an agency due to the following observations. First, internal corruption might exist between government agencies. In fact, in a recent survey by the Computer Institute and the FBI, it was reported that a great deal of organizations face insider attacks (Jarrell, 2002). Internal corruption within an agency may seriously jeopardize the other agencies' trust in the agency. Second, responsibility and seriousness needs to take charge in order for cross-agency information sharing to take place. In fact, one Colonel Steve York, stated that, "The government has to lead by example. Within the government, agencies don't trust each other. The government has to break down these walls . . ." (York, 1997, p. 2).

Requirements on the Trust Model

Information sharing schemes are trust-based; and whether an information sharing scheme can lead to effective information sharing among government agencies is heavily dependent upon the trust model on top of which the information sharing scheme is constructed. The reasons about why there is a lack of trust among agencies imply that the trust model for effective cross-agency information sharing needs to satisfy the following requirements:

A. *The trust model should be built in such a way that win–win information sharing will always increase the mutual trust.* An information sharing procedure between two agencies is a *win–win* procedure if the *interests* (or payoffs) of both agencies will be increased (to a similar degree) when the procedure ends. If win–win information sharing does not increase the mutual trust or even decreases the mutual trust, then the success of a win–win information sharing may not be able to promote more valuable and successful information sharing, since a higher level of mutual trust is usually needed to make the two agencies willing to share more sensitive or important information. On the other hand, if this requirement is satisfied, a win–win information sharing procedure can increase the mutual trust which, in turn, can encourage more valuable win–win information sharing, and since more sensitive information can be shared, more interests can be obtained by both agencies. This property has the potential to transform the situation where "nobody wants to share information" to the situation where "everybody wants to share information."

B. *The trust model should be built in such a way that win–lose information sharing will always decrease the trust of the loser in the winner.* An information sharing procedure is a *win–lose* procedure if the interests of (at least) one agency will be decreased when the procedure ends. When requirement A is satisfied, this requirement implies that "bad" agencies that hide information to "steal" more interests can actually be punished for the following reason: win–lose information sharing hurts mutual trust; and degraded trust will prevent more information sharing. Hence, good agencies will have less and less win–lose information sharing with bad agencies and more and more win–win information sharing with good agencies. As a result, after a relatively long period of time, all the good agencies will gain a lot of interests but the bad agencies will only gain very little.

C. *Requirements A and B indicate that interests must be part of the trust model.*

D. *Requirement C indicates that information validity and utility must be part of the trust model.* This is because the interests that an agency can gain through an information sharing procedure are ultimately determined by the validity and utility of the information the agency received. A piece of information must be *valid* (i.e., with high integrity) to be useful. And the utility of the piece of information is also determined by such factors as whether it is needed by the agency and whether it is critical to solve a critical problem faced by the agency.

Limitations of Existing Trust Models

Of course, the trust model for information sharing among government agencies is built on existing trust models, which are broken down into three layers. Identification and authentication build the first layer of trust; certified attributes built the second layer of trust; and delegation builds the third layer of trust.

However, as we will shown in the next section, a serious drawback of existing information sharing technologies is that the existing trust models do not satisfy the four requirements we have identified above. Existing trust models can enable agencies to trust "who you are," "the attributes you have," "the authorities you have," but cannot enable agencies to trust "this information sharing procedure will increase my interests" or "this information sharing procedure will be a win–win one." In summary, credentials and authorities are the basis of existing trust models, but they cannot deliver interests' guarantees, however, information sharing among government agencies requires an interest-based trust model.

Limitations of the Federal Enterprise Architecture

Although the federal enterprise architecture provides a solid foundation for cross-agency information exchanging, it assumes the use of existing trust models, which are not interest-based as we will show below. As a result, without a new interest-based trust model, even if the FEA is fully implemented, agencies still will be reluctant to freely share information with one another.

Existing Information Sharing Technologies

In this section, we present a classification of existing information sharing technologies and show why they are limited in enabling effective information sharing among government agencies. Existing information sharing technologies can be classified into two categories: (a) *privacy-preserving information sharing*, where two parties with information x and y, respectively, share their information with each other in such a way that a function of x and y, denoted $f(x, y)$, is computed and learned by the two parties, but the privacy of x and y is preserved during the information sharing process, that is, the two parties learn only $f(x, y)$ and nothing else; and (b) *non-privacy-preserving information sharing*, where two parties with information x and y, respectively, cannot only share a function of x and y with each other but also share (part of) x and/or (part of) y with each other.

Privacy-Preserving Information Sharing

Privacy-preserving information sharing techniques can be broken down into three subcategories: (a1) trust third-party techniques, (a2) secure multi-party computation, and (a3) application specific techniques.

Trust third party. The two parties give their data (i.e., x and y) to a "trusted" third party and have the third party do the computation [i.e., computing the value of $f(x, y)$; Ajmani, Morris, & Liskov, 2001; Jefferies, Mitchell, & Walker, 1995]. However, the third party has to be completely trusted, with respect to intent and competence against security breaches. The level of trust required may be too high for this solution to be practically used by government agencies.

Secure multi-party computation. There are two parties with inputs x and y, respectively, (each one is typically a n-dimension vector) and the goal of secure multi-party computation is to compute a function $f(x, y)$ without involving a third party such that the two parties learn only $f(x, y)$ and nothing else. See Goldreich (2001) and Naor and Nissim (2001) for a discussion of various approaches to this problem.

Yao (1986) showed that any multi-party computation can be solved by building a combinational circuit, and simulating that circuit. A variant of Yao's protocol is presented in Naor, Pinkas, and Summer (1999), where the number of oblivious transfers is proportional to the number of inputs and not the size of the circuit. However, this approach is too expensive, especially in communication costs. For example, for $n = 1$ million, the communication time for the circuit-based protocol is 144 days (using a T1 line), which is clearly too long for government agencies to share information with each other.

Application specific solutions. Without involving a third party, efficient privacy-preserving information sharing is

possible in some specific information sharing settings, where some specific data structures and some specific forms of secure multi-party computation [i.e., specific forms of $f(x, y)$] can be exploited substantially to reduce the complexity. For example, Agrawal, Evfimievski, and Srikant (2003) propose a set of efficient protocols for information sharing across private databases, where the notion of minimal information sharing across private databases is formalized, and the unique characteristics of the relational data model and the associated query operators are exploited to achieve both privacy and efficiency. Compared with Yao (1986), for $n = 1$ million, the communication time of Agrawal, Evfimievski, and Srikant (2003) is only 0.5 hours instead of 144 days.

Non-Privacy-Preserving Information Sharing

Privacy preserving is actually not a requirement for most, if not all, information sharing activities between government agencies. The techniques for non-privacy-preserving information sharing can be broken down into three subcategories: (b1) privilege-based information sharing, (b3) trust-based information sharing, and (b3) fair data exchange.

Privilege-based information sharing. Access control technologies can be used to support both intraorganization and interorganization information sharing. Within a single organization, information (stored in files or databases) can be shared by assigning proper *authorizations* (or privileges) to the subjects (i.e., users). For example, when Alice and Bob are both authorized to access a piece of information x, x is shared by Alice and Bob. Authorization management is well studied in the field of access control. For example, a popular authorization management scheme is role based access control (Coyne, Feinstein, Sandhu, & Youman, 1996). Interorganization access control focuses on how to allow a user U of organization A to access a piece of information x owned by organization B. To make the access control policy of each organization intact, interorganization access control needs to map (or translate) U's authorizations in organization A to some corresponding authorizations in organization B. Good methods to do such mappings are developed in the literature (Akella et al., 2000; Osborn, 2002). However, interorganization access control is limited in sharing information among government agencies, since interorganization access control assumes that organizations *trust* each other with respect to both the validity of information and the validity of authorizations; however, this level of trust may not be "reachable" between two real world agencies in many information sharing scenarios.

Finally, for a set of organizations that distribute, store, and utilize their information in a decentralized fashion, *digital credentials* (on top of a PKI) can be used to facilitate the enforcement of both intraorganization and interorganization access control. Using digital credentials, each organization can certify the roles, attributes, and authorizations associated with her employees, and the PKI can ensure that the digital credentials issued by each organization are verifiable so that forged credentials will never be used. In addition, to make such enforcements more efficient, privileges can be *delegated* from one party to another party (Ahn, Chu, & Zhang, 2001; Gasser & McDermott, 1990).

Trust-based information sharing. Although privilege-based information sharing may work well under *centralized* trust, privilege-based information sharing cannot handle information sharing under *decentralized* trust. Under centralized trust, there is one authority that everybody trusts, however, such authority does not exist under decentralized trust where one party typically trusts only a couple of other parties. We can use information sharing under centralized trust for *trust-based information sharing*. Trust plays an implicit role in privilege-based information sharing where every credential can be verified in a straightforward way. In contrast, trust plays a much more explicit role in trusted-based information sharing. For example, if organization A does not trust organization B, A will not trust credentials issued by B, and no employee of B can access information owned by A unless he or she holds a credential issued by a party trusted by A.

In trust-based information sharing, everybody determines whether to share a piece of information (with others) based on trust; and the access control is typically enforced based on the *attributes* or *roles* associated with the subjects. To illustrate, assume an online software store O would like to offer (or share) a free software package only to college students, while O does not trust any university directly. Instead, O accepts accrediting credentials issued by the software Accrediting Board for Universities (ABU). As a result, to get the software for free, a student Alice of university StateU needs to show O not only a credential issued by StateU which says that Alice is a student of StateU, but also a credential issued by ABU which says that StateU is an accredited university of ABU. In other words, O establishes its trust in Alice through a *chain* of two credentials: the credential issued by ABU makes O trust StateU and its credentials, and the credential issued by StateU makes O trust Alice.

In summary, trust-based information sharing is composed of two key technologies: (a) *trust management*; and (b) attributed or role-based access control. Trust management enables a party to trust a credential (and its content) issued by a stranger through a chain of other "proving" credentials. Then attribute- (or role-) based access control can be used to disclose a piece of information based on the attributes certified by a trusted credential. Delegation is an inherent part of trust-based information sharing. For example, in the above example, O in fact delegates the authority over the identification of preferred customers to ABU.

Trust management technologies can be broken down into three subcategories: (a) credential-chain-based trust management, (b) trust negotiation, and (c) ad hoc trust management.

- Credential-chain-based trust management: In this category, a chain of credentials issued by a set of parties can be used to make one party trust another party with respect to the other

party's attributes. Technologies in this category include but are not limited to SPKI/SDSI (Clarke, Elien, Ellison, Fredette, Morcos, & Rivest, 2001). KeyNote (Blaze et al., 1999), Policy-Maker (Blaze et al., 1996), REFEREE (Chu et al., 1997), and Delegation Logic (Grosof et al., 2003). Moreover, in Josang (1996, 1999), the concept of partial trust is introduced, where one may consider someone to be completely trusted, completely untrusted, completely unknown, or somewhere in-between.

- Trust negotiation (Hess et al., 2002): In this category, each party can have multiple credentials containing different sets of attributes. To establish trust between two parties, they use access control policies that specify what combinations of digital credentials a stranger must disclose to gain access to a local information resource. However, to preserve the privacy of sensitive credentials, a credential will not be disclosed to party B by party A unless party A has a certain level of trust in B, which is established based on a set of credentials disclosed from B to A. Therefore, A and B need to interactively *negotiate* their trust on each other through typically multiple rounds of trust-based credential exchange. A party can use many different strategies to negotiate trust, however, to preserve parties' autonomy, each party should ideally be able to choose its negotiation strategy independently, while still being guaranteed that negotiations will succeed whenever possible, i.e., that the two parties' strategies will interoperate.
- Ah hoc trust management: In this category, credentials are not necessary to establish trust. For example, in Hailes and Rahman (1997), a simple distributed trust model is proposed where numerical ad hoc trust levels are maintained for each party to measure the degrees to which it trusts the other parties. In Damiani, Paraboschi and Vimercati (2002), a protocol for maintaining and exchanging *reputations* in peer-to-peer networks is proposed. Note that reputations can be directly mapped to trust levels. In Liu and Xiong (2003), a reputation-based trust model for peer-to-peer e-commerce communities is proposed.

Fair data exchange. Experience with e-commerce has shown that an *exchange* of one data item for another between mutually distrusted parties is usually the crux of an electronic transaction. Fair data exchange technologies have been used in many applications such as nonrepudiation of message transmission (Gollmann & Zhou, 1996), certified mail (Deng, Gong, Lazer, & Wang, 1996), contract signing (Ben-Or, Goldreich, Micali, & Rivest, 1990), and e-payment systems (Cox, Sirbu, & Tygar,1995; Tygar, 1998). An exchange is *fair* if at the end of the exchange, either each party receives the item it expects or neither party receives any additional information about the other's item. Fair data exchange protocols in the literature can be broken into two categories: *third-party* protocols which use a *trusted* or *semi-trusted* third party and *gradual exchange* protocols (Ben-Or et al., 1990) where the probability of correctness is gradually increased over several rounds of communications. Third-party protocols can be further classified into two subcategories: *exchanges with online third parties* (Cox et al., 1995); Franklin & Reiter, 1997), where an exchange cannot be completed without using the trusted channel even if both parties play honestly, and *exchange with offline third parties* (Asokan & Shoup, 1998; Bao, Deng, & Mao, 1998), where an exchange can be completed without interferences of the trusted third party if the two parties play honestly and the third party is needed only when a party does not play honestly. A third party is trusted if it will neither misbehave on its own, nor conspire with either of the players. A third party is semi-trusted if the third party may misbehave on its own but will not conspire with either of the two parties (Franklin & Reiter, 1997).

Although information sharing usually involves data exchange, the type of information sharing in fair data exchange and the type of information sharing in E-Government are very different. In particular, information sharing in fair data exchange is fairness-based, but information sharing in E-Government is trust-based. Fair data exchange protocols focus on fairness, non-repudiation, and atomicity, but information sharing in E-Government focuses on trust, access control, and privacy. Trust is easy to establish in fair data exchange, but hard in information sharing in E-Government, where fairness is not necessarily a requirement. Information sharing in E-Government typically requires the two pieces of information to be exchanged between two parties are of similar types, but fair data exchange protocols typically exchange very different types of information.

As a result, fair data exchange technologies are limited in supporting information sharing among government agencies. Relying on a trusted third party may be too strong a requirement for practical information sharing. Using a semi-trusted third party cannot help much, since although the third party will not conspire with either of the two agencies, none of the two agencies may want to reveal the information they want to share with each other to the third party due to the sensitivity of the information. Finally, using a gradual data exchange protocol may be too expensive and cause substantial delays.

The Uniqueness of Our Trust-Based Information Sharing Scheme

Although our information sharing scheme is built on top of the set of existing information sharing technologies, it has several unique features. In particular,

- *Our scheme is FEA specific.* Our scheme is motivated by a systematic analysis of the specific security requirements for information sharing in E-Government. To our best knowledge, our scheme is the only information sharing scheme that satisfies the set of security requirements identified in the previous section.
- *Our scheme makes validity and utility of information part of the trust model.* That is, whether Agency A trusts agency B is also affected by the information (and its validity and utility) that A has obtained from B besides credentials. In credential-chain-based trust management and trust negotiation, trust is only affected by credentials. However, experience with real-world cross-agency information sharing shows that even if Agency A believes that an agent of B has a set of eligible

attributes, A may not want to share a piece of sensitive information with the agent due to some conflicts of interests between A and B.

- *Interleaved information sharing and trust negotiation.* In Hoque (2000), each information unit is only associated with a combination of credentials, and the trust level is increased only due to the disclosure of more credentials. By contrast, in our scheme, the trust level may also be increased when the shared information is valid and appreciated. In Hoque (2000), an information unit will be disclosed only when the trust negotiation succeeds. Otherwise, no information will be disclosed. By contrast, in our protocol, information sharing and trust negotiation are interleaved. Partial information sharing is, in fact, used to "negotiate" trust. Even if the two parties cannot finish the *n*-round information sharing process, they still could share some valuable information with each other during the process. In this way, information sharing is more efficient and cost-effective.
- *Web services based implementation.* Such implementation not only enables our scheme to be seamlessly integrated into the FEA reference model, but also enables our scheme to be easily integrated into a variety of existing E-Government applications.

Information Sharing Framework

Our information sharing framework has two parts: (a) an interest-based trust model, and (2) an information sharing protocol built on top of the trust model.

Assumptions

- *Centralized trust.* We assume that all the agencies that want to share information with each other trust a single *certification authority* (CA) and any *credentials* issued by the CA. We assume the CA may delegate the authority to certify an attribute of a government agent to an agency. In this way, one agency may trust the credentials issued by another agency in an indirect fashion through a specific delegation credential issued by the CA. Two agencies need not know each other in advance to be able to share information.
- *Secure communication channels.* We assume that all the agencies are using a secure communication channel to share information with each other since the shared information can be very sensitive. We assume the secure channel can ensure the confidentiality and integrity of the messages being transmitted. Denial-of-service attacks may happen but are out of the scope of this paper.
- *Intra-agency information sharing is in place.* In fact, intra-agency information sharing can be easily achieved using the set of existing information sharing technologies overviewed in the above sectopn, where (a) centralized trust is naturally built; (b) when the agency's information systems are centralized, access control can be directly used to share information among units within the agency where agents playing a more responsible role can access more sensitive information than other agents; (c) when the agency's information systems are distributed and of large scale, digital credentials and delegation can be used, where the amount of information that an agent can access is determined by "which attributes (e.g., a role) are associated with the agent?" Note that intra-agency information sharing typically does not need an interest-based trust model and existing trust models are usually good enough to enable effective intra-agency information sharing.

Design Goals and Requirements

The ultimate goal of our information sharing scheme is to transform the situation where "nobody wants to share information" to the situation where "everybody wants to proactively share information," so that mutual trust can be rebuilt among agencies, differences between agencies can be tolerated, misunderstandings among agencies can be minimized, conflicts can be resolved, and effective information sharing can be achieved.

To achieve this goal, the information sharing scheme needs to satisfy the following requirements:

1. The trust building procedure, part of the information sharing protocol, should provide enough incentives for agencies to do win–win information sharing and should discourage win–lose information sharing. In this way, more agencies will do win–win information sharing and "bad" agencies will be punished by doing win–lose information sharing.
2. To satisfy requirement (1), the information sharing protocol must ensure that no agency can get significant amount of advantages during any steps of the protocol if he/she hides information or does not play honestly, because otherwise win–lose information sharing will be encouraged.
3. Requirement (2) indicates that during any step of the information sharing protocol no significant amount of information can be disclosed, because otherwise the agency who receives the significant amount of information can abort and get significant amount of advantages.
4. Since no significant amount of information can be disclosed in any step of the protocol, *gradual* information disclosing is required.
5. Requirements (2) and (4) indicate the disclosing of information from agency A to agency B and from B to A should be *interleaved* with each other, because otherwise the agency who discloses the information first will be the loser even if he/she discloses the information gradually.
6. Requirements (1) and (5) indicate that trust building and information exchange should be interleaved with each other, since validity and utility of information are part of trust, and trust is a precondition for information disclosure.
7. Fairness: The protocol should ensure that every information sharing procedure is *fair*, that is, no matter where the procedure stops (note that an agency may abort at any point of the procedure), the information or the interests gained by the two agencies (involved in the procedure) should be almost the same. From the perspective of trust building, fairness can be interpreted as either the information is exchanged and the trust levels are increased, or the information is not exchanged at all and the trust levels are dropped, but nothing in between could happen. This requirement can be viewed as being deduced from requirement (1).

Information Sharing Policies

We use four types of information sharing policies in the information sharing protocol. They are *information release* (IR) policies, *credential release* (CR) policies, *trust level adjustment* (TLA) policies, and *access control* (AC) policies.

To illustrate the four types of policies, we need to first introduce a set of notations. In our protocol, we assume there are two agencies: A and B. We assume $C_{a1}, \ldots, C_{am}$ are the *m* credentials held by agency A; and $C_{b1}, \ldots, C_{bn}$ are the *n* credentials held by agency B. We assume $U_{a1}, \ldots, U_{au}$ are the *u* units of information owned by A; and $U_{b1}, \ldots, U_{bv}$ are the *v* units of information owned by B. We assume TL(A) is the *trust level* of A in the eyes of agency B, which indicates the degree to which B trusts A; and TL(B) is the trust level of B maintained by A. We assume the trust levels are quantified using an integer from 1 to 10, the simplest way. Although trust levels can be quantified in a more complicated way, numerical trust levels are enough to show the idea of our information sharing protocol. Finally, we assume *utility*(U_{ai}) measures the utility level of information unit U_{ai} to agency A, which indicates the degree to which U_{ai} is useful to A. It is clear that utility( ) is a crucial function since it determines the importance of an information unit. However, measuring or quantifying the utility level of an information unit is a complicated issue, since the utility level of an information unit is usually a relative dynamic measurement. That is, (a) an information unit which is not useful to agency A can be very valuable to agency B, and (b) an information unit can be much more valuable to an agency if it was released to the agency 5 minutes earlier. Although a detailed study of this issue is out of the scope of this paper, we believe artificial intelligence and natural language processing would play an important role in this research. Finally, to make our protocol complete and tangible, we assume utilities are simply quantified using an integer from 1 to 7, and utilities are quantified by human beings (e.g., an FBI agent). (Here, "7" is determined in an ad hoc manner.)

Now we define the four types of information sharing policies one by one following an order that can best illustrate the relationships among these policies.

An information release policy is used to help agency A to determine when a specific information unit can be disclosed to agency B. An IR policy is simply a set of *component policies*. An example IR component policy is shown in Figure 1, where we can see that an IR component policy is composed of two parts: the *condition* part is a conjunction or disjunction of a set of predicates, and the *action* part specifies the information unit that can be released when the condition is satisfied. The condition of an IR component policy can consist of 0 to 4 predicates. When there are 0 predicates, there is no restriction to disclose the information unit. When there are 4 predicates, as Figure 1 shows, when agency A wants to determine whether to disclose an information unit to agency B, the first predicate is to check if B is "authorized" to access the information unit. How such checking should be done is specified by an access control policy that we will define shortly.

The second predicate is to verify the validity of the information units that agency A has already received from B through the (same) information sharing procedure so far. The third predicate is to check the utilities already earned by A through the protocol run. We assume the validity is checked and the utilities are measured by a human being. How to have a software agent check the validation and measure the utilities is one of our future research topics. The fourth predicate is to check if A has built a certain level of trust in B.

In summary, the first predicate builds the lower level trust needed for cross-agency information sharing. The second and third predicates are to satisfy design requirements (1), (2), (3), (4), (5), and (7). The fourth predicate is to satisfy design requirement (6).

In our scheme, trust levels are dynamically maintained based on a specific trust level adjustment policy. An example TLA component policy is shown in Figure 2, where the first and second predicates build the lower level trust on top of which interest-based trust can be built. The *attributes-carried()* predicate will help agency A build a level of certified-attribute-based trust in B. Note that this level of trust is exactly the type of trust built by existing trust management techniques (see the above section 4). Here, a credential will typically say that an entity has a certain *attribute*. For example, a security clearance level is an attribute of a government agent; and "being an employee of Dept of Justice" is another attribute of the agent. Sometimes, a credential can say a delegation. For example, a credential issued by the Department of Commerce can say that if an agent has the attribute "being an employee of Dept of Justice," then the agent will be given the attribute "a collaborator of Dept of Commerce." The third and fourth predicates build an additional level of interest-based trust. Finally, it should be notd that Figures 1 and 2 show clearly how trust building and information exchange are interleaved with each other.

In our scheme, the disclosure of credentials is also controlled by a credential release policy, since credentials may contain very sensitive attributes (e.g., the responsibility of a

Condition: access-control-passed($\{C_{b1}, ..., C_{bp}\}$, U_{aj}) AND received-and-valid(U_{b1}, ..., U_{bq}) AND utility(U_{b1}, ..., U_{bq}) >3 AND TL(B) >4

Action: release information unit Uaj to B

FIG. 1. An example IR component policy.

Condition: received-and-verified(C_{b1}, ..., C_{bp}) AND attributes-carried($\{C_{b1}, ..., C_{bp}\}$, $\{a_1, ..., a_r\}$) AND received-and-valid(U_{b1}, ..., U_{bq}) AND utility(U_{b1}, ..., U_{bq}) >4

Action: increase the value of TL(B) by 1

FIG. 2. An example TLA component policy.

Condition: received-and-valid(U_{b1}, ..., U_{bq}) AND utility(U_{b1}, ..., U_{bq}) >3 AND TL(B) >5 AND received-and-verified(C_{b1}, ..., C_{bp}) AND attributes-carried({C_{b1}, ..., C_{bp}}, {a_1, ..., a_r})

Action: release credential C_{aj} to B

FIG. 3. An example CR component policy.

special FBI agent). An example CR component policy is shown in Figure 3, where we can see that whether to release a credential to B is not only dependent on whether B has released some sensitive credentials to A, but also dependent upon the level of interest-based trust A has in B, since releasing sensitive credentials may also hurt the interests of an agency. It should be noticed that our CR policies are different from existing trust negotiation policies (Hess et al., 2002).

Finally, an access control policy determines whether an agency is eligible or authorized to access an information unit owned by another agency. An example AC component policy is shown in Figure 4, where we can see that our access control policies are almost the same as existing trust-based information access policies (Grosof et al., 2003) where authorities are associated with attributes; and an entity with attribute *x* will have the authorities associate with *x*. Nevertheless, it should be noted that Figures 3 and 4 show that information exchange and credential negotiation are actually interleaved with each other. Figures 1, 2, and 3 show that trust building and credential negotiation are actually also interleaved with each other.

More formally, the syntax of these four information sharing policies in BNF is shown in Figure 5.

Information Sharing Protocol

The four types of information sharing policies clearly indicate how an information sharing protocol should run. In this section, we first use an example to illustrate how the protocol works, then specify the protocol.

The example is shown in Figure 6, where the protocol run involves seven messages and six major procedures, namely P1, . . . , P6, after some of the messages. We assume agency A initiates the information sharing procedure by asking B to disclose information unit U_{b3}. We assume A also discloses a credential, namely C_{a1}, together with the request. B processes this request via P1 as follows: (a) P1 will first use C_{a1} to adjust the value of TL(A) according to a TLA component policy, we assume as a result, TL(A) is increased from 1 to 2. Next, P1 will check the set of IR component policies regarding U_{b3}. We assume {C_{a1}, C_{a2}, C_{a3}} are needed to access U_{b3}. So each

Condition: received-and-verified(C_{b1}, ..., C_{bp}) AND attributes-carried({C_{b1}, ..., C_{bp}}, {a_1, ..., a_r})

Action: B can access information unit U_{aj}

FIG. 4. An example AC component policy.

⟨list of X⟩ ::= ⟨X⟩ | ⟨X⟩ “,” ⟨list of X⟩
⟨conj list of X⟩ ::= ⟨X⟩ | ⟨X⟩ “AND” ⟨conj list of X⟩
⟨OP⟩ ::= > | ≥
⟨PM⟩ ::= + | -
⟨TL⟩ ::= ⟨func⟩
⟨utility⟩ ::= ⟨func⟩
⟨access-control-passed⟩ ::= ⟨pred⟩
⟨received-and-valid⟩ ::= ⟨pred⟩
⟨attributes-carried⟩ ::= ⟨pred⟩
⟨received-and-verified⟩ ::= ⟨pred⟩
⟨access-control-passed-pred⟩ ::= ⟨access-control-passed⟩ “(” “{” ⟨list of credential⟩ “}” “,” ⟨info-unit⟩
⟨received-and-valid-pred⟩ ::= ⟨received-and-valid⟩ “(” ⟨list of info-unit⟩ “)” ⟨OP⟩ ⟨integer⟩
⟨TL-pred⟩ ::= ⟨TL⟩ “(” ⟨prin⟩ “)” ⟨OP⟩ ⟨integer⟩
⟨utility-pred⟩ ::= ⟨utility⟩ “(” ⟨list of info-unit⟩ “)” ⟨OP⟩ ⟨integer⟩
⟨attributes-carried-pred⟩ ::= ⟨attributes-carried⟩ “(” “{“ ⟨list of credential⟩ “}” “,” “{“ ⟨list of attributes⟩ “}” “)”
⟨received-and-verified-pred⟩ ::= ⟨received-and-verified⟩ “(” ⟨list of credential⟩ “)”
⟨IR-condition⟩ ::= ⟨conj list of access-control-passed-pred⟩ “AND” ⟨conj list of received-and-valid-pred⟩ “AND” ⟨conj list of utility-pred⟩ “AND” ⟨conj list of TL-pred⟩
⟨IR-action⟩ ::= “release” ⟨info-unit⟩ “to” ⟨prin⟩
⟨An IR component policy⟩ ::= “Condition:” ⟨IR-condition⟩ “Action:” ⟨IR-action⟩
⟨TLA-condition⟩ ::= ⟨conj list of received-and-verified-pred⟩ “AND” ⟨conj list of attributes-carried-pred⟩ “AND” ⟨conj list of received-and-valid-pred⟩ “AND” ⟨conj list of utility-pred⟩
⟨TLA-action⟩ ::= ⟨TL⟩ “(” ⟨prin⟩ “)” “=” ⟨TL⟩ “(” ⟨prin⟩ “)” ⟨PM⟩ ⟨integer⟩
⟨A TLA component policy⟩ ::= “Condition:” ⟨TLA-condition⟩ “Action:” ⟨TLA-action⟩
⟨CR-condition⟩ ::= ⟨conj list of received-and-valid-pred⟩ “AND” ⟨conj list of utility-pred⟩ “AND” ⟨conj list of TL-pred⟩ “AND” ⟨conj list of received-and-verified-pred⟩ “AND” ⟨conj list of attributes-carried-pred⟩
⟨CR-action⟩ ::= “release” ⟨credential⟩ “to” ⟨prin⟩
⟨A CR component policy⟩ ::= “Condition:” ⟨CR-condition⟩ “Action:” ⟨CR-action⟩
⟨AC-condition⟩ ::= ⟨conj list of received-and-verified-pred⟩ “AND” ⟨conj list of attributes-carried-pred⟩
⟨AC-action⟩ ::= “release” ⟨info-unit⟩ “to” ⟨prin⟩
⟨An AC component policy⟩ ::= “Condition:” ⟨AC-condition⟩ “Action:” ⟨AC-action⟩

FIG. 5. Syntax of information sharing policies in BNF, in which ⟨pred⟩, ⟨func⟩, ⟨prin⟩, and ⟨integer⟩ represent a predicate symbol, a function symbol, a principal, and an integer, respectively. The first definition, ⟨list of X⟩, is a macro.

access-control-passed() predicate in these component policies will be evaluated as FLASE. Next, to enable agency A to disclose C_{a2} and C_{a3} to B (so that B may disclose U_{b3} to A), B needs also to disclose some information units and credentials to A so that the set of security requirements listed in the section, “Design Goals and Requirements” can be satisfied. Hence, B will disclose U_{b1} and C_{b1} in message 2. Here U_{b1} can be disclosed because we assume there is an IR component policy regarding U_{b1} allows B to do so based on C_{a1} and TL(A). Moreover, B asks A to disclose U_{a3}.

When A receives message 2, P2 will first use U_{b1} and C_{b1} to adjust the value of TL(B). Next, P2 will check his/her IR and

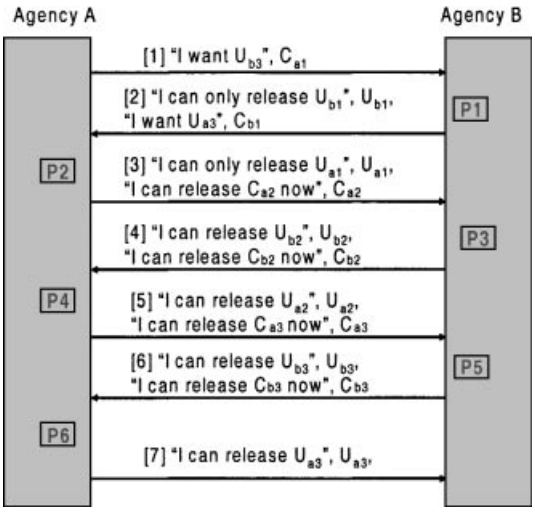

FIG. 6. An example run of the information sharing protocol.

CR component policies to figure out which credentials and/or information units can be disclosed. We assume $\{C_{b1}, C_{b2}, C_{b3}\}$ are needed to disclose U_{a3}, and U_{a1} and C_{a2} can be disclosed due to U_{b1}, C_{b1}, and the fact that TL(B) is increased.

When B receives message 3, P3 will do almost the same type of things as P2 does and as a result, U_{b2} and C_{b2} are disclosed. When A receives message 4, P4 will do almost the same type of things as P2 does and as a result, U_{a2} and C_{a3} are disclosed. When B receives message 5, P5 will do almost the same type of things as P3 does and as a result, U_{b3} and C_{b3} are disclosed. Finally, when A receives message 6, A will disclose U_{a3} in message 7.

Discussion

First, some people may wonder "How can agency A know after receiving message 1 that U_{a1} and C_{a2} will be needed to enable B to finally disclose U_{b3}?" The answer has two parts: (a) agency B can tell A via message 2 which kind of credentials is needed for B to disclose U_{b3}. B can figure out this information based on his/her IR and CR component policies using the techniques developed in Hess et al. (2002). (b) U_{a1} is chosen primarily because of the relation between U_{a1} and U_{a3}. Since B wants U_{a3}, so information related to U_{a3} should be able to increase the interests of B and encourage B to increase his/her trust in A. For example, U_{a1} can be an approximation (or estimation) of U_{a3}.

More formally, the information sharing protocol can be specified as shown in Figure 7.

Second, our information sharing scheme has three key features. (a) It uses an interest-based trust model where utilities of shared information units are part of the mutual trust. (b) It interleaves trust negotiation and information exchange, where not only trust negotiation is gradually done but also information sharing is gradually done. (c) It satisfies the set of security requirements identified in the "Design Goals and Requirements" section. In particular, our protocol does gradual, interleaved information sharing where no agency can get significant amount of advantages during any step of the protocol, and the fairness is ensured.

(a) The protocol has a single type of messages. Each *message* may have up to four parts: Part 1 - requesting an information unit; Part 2 – requesting a credential; Part 3 – (releasing) a set of credentials; Part 4 – (releasing) a set of information units.

(b) The protocol is started by a message that must have Part 1.

(c) The protocol is between two *parties*. Each party can *exit* the protocol at any point of time.

(d) In each run of the protocol, at least one party wants to get a specific information unit from the other party.

(e) Each party maintains a stack of requests. The request on the top of the stack is the *current* request.

(f) When a party receives a message,

(f.1) Part 1 or Part 2 of the message, if not empty, will be pushed into the request stack.

(f.2) She will use Part 3 and Part 4 (if any) of the message to adjust the trust level of the other party based on her TLA policies.

(f.3) If the current request is for an information unit (credential), she will check her IR (CR) component policies regarding the request, denoted *x (c)*. As a result, if *x (c)* can be released based on Part 3 and Part 4 of the message and the credentials and information units she has received previously, she will compose a message to release *x (c)*. When *x (c)* is released, the request will be removed from the stack. Otherwise, she will identify the set of credentials and/or information units that the other party needs to provide first, then she will compose a message to request the other party to release these items. Moreover, she will check her IR (CR) component policies to see which info units or credentials that are relevant to *x* or *c* can be released now. These items will be then included in the message back to the other party.

(f.4) Periodically when a party finds that she cannot release an item, she will check if there is any *deadlock* generated. For example, a deadlock will happen when one party needs item X to release Y, but the other party needs Y to release X. If such a deadlock exist, the protocol will abort.

(g) The protocol will terminate when the request stack of each party becomes empty.

FIG. 7. The information sharing protocol.

Third, it should be noted that the reason that a deadlock may exist is because the information sharing policies of the two parties involved are *inconsistent* with each other. Consistent information sharing policies can be developed to avoid such deadlocks, although how to develop such policies is out of the scope of this paper.

Prototype Implementation

To be completely compatible with the FEA, we implement our information sharing scheme using XML Web services. Web services are an emerging, state-of-the-art

technology that provides the perfect technology for integrating and sharing information among agencies. One of its primary features is interoperability of two systems. Web services are a critical part of both the FEA conceptual process reference model and the interoperability reference model, since Web services can enable government agencies with different types of information systems (e.g., different data management schemes, different operating systems, different network protocols) to communicate and interact with each other in a platform transparent fashion.

Components of Web Services

The components of XML Web services include applications, UDDI, WSDL, and SOAP (Champion, 2002). Applications may be clients of XML Web services provided by another application, providers of XML Web services to other applications, or both (Federal CIO Council, 2002). UDDI, Universal Description Discovery and Integration, provides a *registry* of available XML Web services. The Web Service Description Language (WSDL) provides an XML-based description of XML Web services and how to interact with them (Cameron, 1998). The WSDL is the interface definition language of XML Web services. Finally, SOAP, or Simple Object Access Protocol, is a lightweight remote-procedure-call protocol that uses XML to format messages and HTTP to transmit messages (Ajmani et al., 2001). An example SOAP message is shown in Figure 8.

The general procedure of using a Web service starts when a client searches for the Web service from an UDDI registry. The UDDI registry in some sense is equivalent to the yellow pages in a phone book. However, the UDDI registry is not a completely secure registry and in fact is a public registry. Many government officials are very reluctant to populate a public registry of Web services using the UDDI standard (Ewald, 2002). As a result, government agencies that wish to share their services with each other will share the service description information through a secure communication channel. In either way, the client will be able to obtain a description of the Web service, which is written in WSDL, through a private or public registry. Then, the client can retrieve the description (written in WSDL), parse the XML data contained in the description, and learn the URL of the Web service and the right ways to call the Web service using SOAP. At the Web server end, each SOAP request will be first received by the SOAP listener then forwarded to a specific request processor, called the WSDL server, which will parse the XML data contained in the SOAP request. A set of service requests, which can be understood by the *server application* will be generated after the XML data are parsed, and the set of service requests will be sent to the server application for processing.

```
[SoapHeader("Credentials",Required=true)]
[WebMethod]
public return_type function_name()
{
 //Code used to implement web method
}
```

FIG. 8. SOAP header format in .NET.

Microsoft .NET XML Web Services

We implement our information sharing scheme using Microsoft .NET, which provides a very powerful, mobile, and developer friendly infrastructure for creating a Web service. Many real-world government agencies have already deployed some .NET Web services.

Some important components of XML Web services in .NET include the .ASMX file extension, the .ASPX file extension, and the WSDL. Initially, the Web service itself is implemented using the .ASMX file extension. Essentially, .ASMX files are based on ASP .NET. In these .ASMX files, besides the code that implements each Web method of the Web service, the set of Web methods that compose the Web service are also defined using a set of <WebMethod> tags.

After the (Web service) provider creates all of its functions or Web methods in the .ASMX files, it creates a WSDL description of the Web service, where the client can find the exposed methods of the Web service, the parameters and return types that these methods expose, and any other exposed information. The WSDL description is generated based on the set of <WebMethod> tags. Finally, the client uses an .ASPX file to call the Web service. The .ASPX file is generated based on the WSDL description of the Web service that the client can get from a registry.

How to Use XML Web Services to Implement Our Information Sharing Protocol

In general, we can use XML Web services to implement our information sharing protocol as follows: (a) each agency X provides a set of *information sharing Web services* to the agencies who may want to share information with agency X. The WSDL descriptions of these Web services should be composed and disclosed to an agency that wants to share information with X in a *secure* way, since such descriptions may themselves contain sensitive information and should only be readable to certain authorized agencies. (b) In addition, each agency X provides a specific *information sharing-support* Web service to her own agents. Since this service (i.e., the URL) is pre-known to every agent of X, no WSDL description or registration is needed. (c) When an agent of agency X wants to get a specific information unit U from agency Y, the agent will first authenticate himself/herself to the information sharing-support service (“support service” for short) provided by X. Next, the agent will use a standardized interface (provided by the support service) to ask the support service to *initiate* an information sharing procedure with agency Y. However, to have an error-free information sharing procedure with Y, the support service must first understand which kinds of information sharing

services are provided by Y and how these services should be called (or used) in terms of the procedures and the message format that need to be followed. Hence, before the support service starts the information sharing procedure, it will first search for the WSDL descriptions of the services provided by Y. Next, the support service will compose a request message according to the service descriptions and send it to an information sharing Web service provided by Y (note that the request message is similar to message (1) in Fig. 6). (d) When the information sharing service of Y receives the request message, it will process the request according to the information sharing protocol we illustrated in "Information Sharing Protocol" section. Note that if an agent of Y (i.e., a human being) needs to be involved in the information sharing procedure, the service of Y will notify the agent and prepare an interface for the agent to jump in. (e) In the following, the agent of Y and the agent of X can follow the information sharing protocol (e.g., the procedure shown in Fig. 6) in a naïve manner, except that in the implementation the support service functions as the "proxy" for the agent of X, and the information sharing service functions as the "proxy" for the agent of Y.

1. FBI, the web service provider agency, develops its information sharing policies and establishes the corresponding web methods
2. The FBI passes the WSDL description of its information sharing web service to a secure channel
3. The CIA agent, i.e., the web service client, retrieves the WSDL description
4. The client parses the WSDL description, calls the corresponding web methods of the information sharing web service to initiate and interact with the FBI's service in such a way that the information sharing protocol can be successful executed

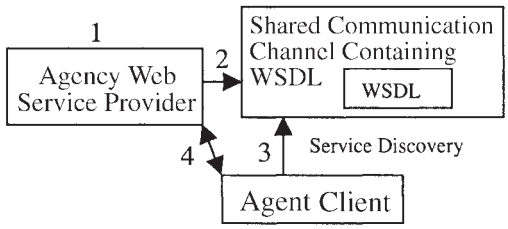

FIG. 9. Steps of information sharing between two agencies using Web services.

An Example Implementation

To better understand how exactly Web services can be used for two agencies to share information (with each other) within the federal enterprise architecture framework, we implemented a simple information sharing system prototype using XML Web services for a real-world information sharing application between two agencies. In particular, we use this prototype to support FBI and CIA to share anti-terrorism information with each other.[1] However, for simplicity, we implement the FBI's information sharing Web service, but do not implement the support service of the CIA; instead, we make a CIA agent directly call the FBI's service. (Note that this implementation can be easily extended to include all the components we mentioned in the above section.)

In this prototype, we assume both the FBI and the CIA contain pertinent information with regard to anti-terrorism. For example, the FBI may maintain such anti-terrorism information as criminal lists and the CIA may maintain such anti-terrorism information as a set of intelligence gatherings.[1] The steps that the two agencies will take in sharing information with each other are depicted in Figure 9. In particular, we assume a CIA agent wants to get an information unit from the FBI. So initially, the FBI will create an information sharing Web service and specify the Web methods for the service. Then, the FBI will send the WSDL description of the service (or the set of Web methods) to a secure Web site (or a secure channel) for the CIA agent to retrieve. A sample of the WSDL service description is shown in Figure 10. Moreover, one way to create the secure channel in the federal enterprise architecture is to use a trusted *broker*. In particular, when one agency X needs to share information with another agency Y, it can simply send its WSDL service description along with the agency it wants to share information with, to a broker. The broker will allow an agency to access the service description only if the agency is authenticated as Y.

Once the client (on behalf of the CIA agent) retrieves the WSDL description, using an XML parser, it will analyze the service description and determine the input arguments for the Web methods that it needs to invoke during the protocol run via SOAP messaging.

Key Features

The prototype implementation has several interesting features, which are as follows. These features indicate the benefits of using XML Web services.

- It supports two-way information sharing. When the CIA client calls a FBI Web method, the client can disclose some credentials or information to the FBI. On the other hand, the execution of a FBI Web method can disclose some credentials or information to the CIA agent in several ways. For example, the FBI Web method can show an information unit or a credential directly on the interface (i.e., a Web browser) used by the CIA agent. For another instance, the FBI Web method can guide the CIA agent to download an information unit or a credential via the Web browser.
- It provides multiple Web methods for building trust.
- It exploits SOAP header authentication to counter session hijack attacks. The idea of SOAP header authentication is shown in Figure 8. That is, a Web method can be executed only if the credentials (e.g., a password) provided by the client can authenticate the client. Using this technique, the FBI Web service can enforce *dynamic* authentication for

[1]The FBI and CIA used in this implementation are used only as example agencies. The real government agencies may contain different names, perform different responsibilities, and contain different types of information.

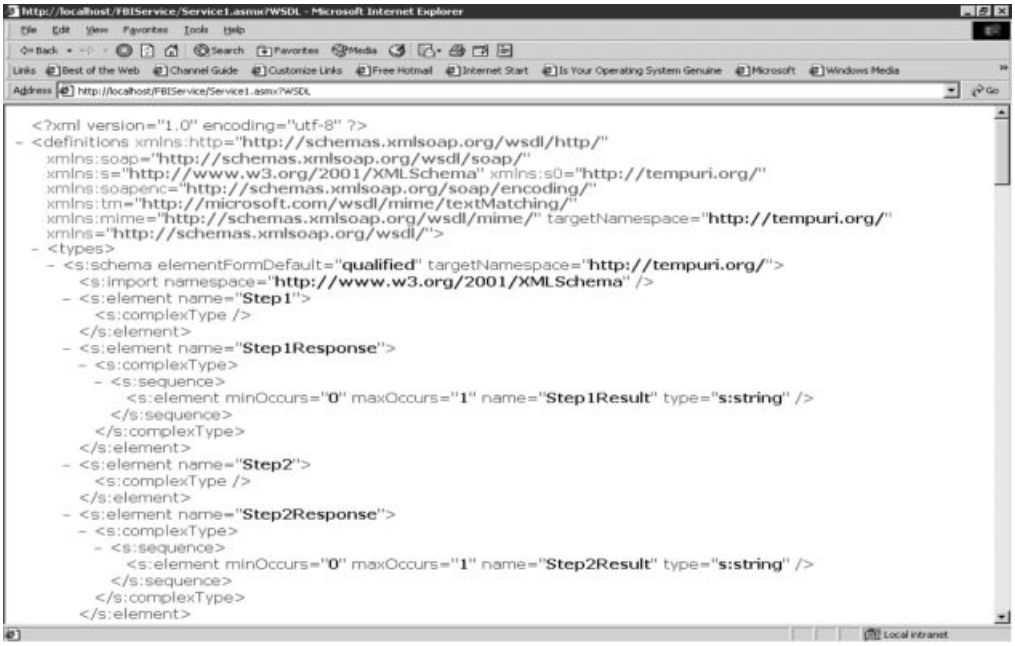

FIG. 10. WSDL description of an information sharing Web service.

each Web method in such a way that the attacker will not succeed in replaying Web methods or hijacking an ongoing information sharing session.
- It provides privacy for using Web methods. That is, clients not qualified to run a Web method (due to trust and authorizations) would never be able to see what the Web method is about or how the Web method should be used.
- It provides a design for peer-to-peer cross-agency information sharing.
- It provides a powerful, easy-to-use interface with .NET technology.
- It integrates information sharing and data management. Both the CIA client and the FBI Web services manage the information units they may share in a MS ACCESS database.

Limitations

Nevertheless, at this stage, the prototype is still preliminary and it has several limitations, which are as follows. The last three limitations can be overcome by better engineering, while the first limitation needs future research, which we will address in below.

1. It requires human interaction.
2. It is only an emulation of the real-world information sharing activities between FBI and CIA. It uses emulated data instead of real data.
3. It uses a small, centralized database.
4. Using a broker to create a secure channel for WSDL service descriptions distribution introduces some extra overhead compared with using UDDI.

Conclusion and Future Research

Although trust-based information access is well studied in the literature, the existing trust models, which are based on certified attributes, cannot support effective information sharing among government agencies, which requires an interest-based trust model. To solve this information sharing problem, we propose an innovative interest-based trust model and a novel information sharing protocol, where a family of information sharing policies are integrated, and information exchange and trust negotiation are interleaved with and interdependent upon each other. In addition, an implementation of this protocol is presented using the emerging technology of XML Web Services. The implementation is totally compatible with the Federal Enterprise Architecture reference models and can be directly integrated into existing E-Government systems. We believe our cross-agency information sharing scheme can transform the situation where "nobody wants to share information" to the situation where "everybody wants to proactively share information," so that the mutual trust can be rebuilt among agencies, differences between agencies can be tolerated, misunderstandings among agencies can be minimized, conflicts can be resolved, and effective information sharing can be achieved. With secure, effective information sharing, the agencies' ability to predict attacks and preempt them can be significantly enhanced.

In addition, the following future research issues are identified and successful resolution of these research issues can further improve the agencies' ability to effectively and efficiently share information with each other.

Develop Privacy Preserving UDDI Registry

One way to improve upon the existing implementation is to develop a *privacy preserving* UDDI registry. With such a UDDI, an agency A will be able to register his/her information sharing services without losing privacy, in the sense that his/her services will only be visible to the agencies that B

would like to share information with. In this way, agencies can enjoy the simplicity and efficiency of UDDI without losing confidentiality; and no dedicated secure service notice facilities are needed.

Develop Automated Software Agents for Trust Negotiation

Another way to improve upon the existing implementation is to develop automated or semi-automated software agents who negotiate trust and share sensitive information with each other on behalf of the corresponding agencies. In this way, a human being can be completely or partially relieved from the labors (or efforts) needed to run the information sharing protocol; and more efficient and timely information sharing may be achieved. More timely information sharing can mean more timely detection of terrorism attacks and less damage caused by such attacks. A key research issue in using software agents is how to enable a software agent to intelligently evaluate the validity and utility of a piece of information.

Cookie Management for Incremental Trust Negotiation

Another interesting future research area would be to develop cookie management between two agencies. Cookies are information stored on your own computer for future use after some WWW sessions. With the development of cookie management, two agencies that have already built a certain level of trust with one another don't have to redevelop the mutual trust. Instead, they could use cookies to remember the current trust state for future reuse. In this way, information sharing can be made quicker and more efficient. Nevertheless, a drawback of using cookies is that the security risk is increased.

Direct Querying According to Previously Stored Information

In relation to this research, another efficient way for each agency to share information with one another is for them to directly "query" the other agencies' databases according to the database scheme information they were already given. For instance, if two agencies have already shared information with one another, then each agency would have already obtained substantial scheme information about the other agency's databases. Such scheme information may enable both agencies to compose accurate, execu0 queries. By sending a query directly from one agency to another, an agency will be able to pinpoint the information he/she wants to get in a finer granularity.

Acknowledgments

This work is supported by NSF CCR-TC-0233324 and Department of Energy Early Career PI Award.

References

Agrawal, R., Evfimievski, A., & Srikant, R. (2003). Information sharing across private databases. In A. Y. Halevy, Z. G. Ives, & A. Doan (Eds.), Proceedings of the ACM SIGMOD Data 2003 (pp. 86–97). New York: ACM.

Ahn, G., Chu, B., & Zhang, L. (2001). A role-based framework for healthcare information systems. In R. Sandhu & T. Jaeger (Eds.), Proceedings of the ACM Symposium on Access Control Models and Technologies (pp. 153–162). New York: ACM.

Ajmani, S., Morris R., & Liskov, B. (2001). A trusted third-party computation service (Technical report MIT-LCS-TR-847). Cambridge, MA: MIT.

Akella, Y., Bilello, M., Dawson, S., De Capitani di Vimercati, S., Samarati, P., Lincoln, P., et al. (2000, January). Secure access wrapper: Mediating security between heterogeneous databases. Paper presented at the DARPA Information Survivability Conference and Exposition, Hilton Head, SC.

Asokan, N., & Shoup, V. (1998). Asynchronous protocols for optimistic fair exchange. In Proceedings of the IEEE Symposium on Research in Security and Privacy (pp. 86–99). Piscataway, NJ: IEEE.

April, C.A., & Fonseca, B. (2002). Monumental mission. InfoWorld. Retrieved July 27, 2003, from http://www.infoworld.com/article/02/09/06/020909ctgovt_1.html

Bao, F., Deng, R.H., & Mao, W. (1998). Efficient and practical fair exchange protocols with off-line TTP. In Proceedings of the IEEE Symposium on Research in Security and Privacy (pp. 86–99). Piscataway, NJ: IEEE.

Ben-Or, M., Goldreich, O., Micali, S., & Rivest, R.L. (1990). A fair protocol for signing contracts. IEEE Transactions on Information Theory, 36(1), 40–46.

Birbeck, M., Diamond, J., Duckett, J., Gudmundsson, O.G., Kobak, P., Lenz, E., et al. (2001). Professional XML (2nd ed.). Birmingham, UK: Wrox Press.

Blaze, M., Feigenbaum, J., Ioannidis, J., & Keromytis, A.D. (1996). Decentralized trust management. In the Proceedings of the 1996 IEEE Symposium on Security and Privacy (pp.164–173). Piscataway, NJ: IEEE.

Blaze, M., Feigenbaum, J., Ioannidis, J., & Keromytis, A.D. (1999). The keynote trust-management system, version 2, Internet Engineering Task Force (IETF) Request for Comments (RFC) 2704. Retreived from http://www.faqs.org/rfcs/rfc2704.html

Cameron, D. (1998). E-commerce security strategies: Protecting the enterprise. Charleston, SC: Computer Technology Research Group.

Champion, M.E. (2002). Web services architecture: W3C Working Draft, W3C. Retrieved July 27, 2003, from http://www.w3.org/TR/2002/WD-ws-arch-20021114

Chu, Y., Feigenbaum, J., LaMacchia, B., Resnick, P., & Strauss, M. (1997). REFEREE: Trust management for web applications. World Wide Web Journal, 2, 706–734.

Clarke, D., Elien, J., Ellison, C., Fredette, M., Morcos, A., & Rivest, R.L. (2001). Certificate chain discovery in SPKI/SDSI. Journal of Computer Security, 9(4), 285–322.

Cox, B., Sirbu, M., & Tygar, D. (1995, July). NetBill security and transaction protocol. Paper presented at the First USENIX Workshop on Electronic Commerce, New York.

Coyne, E.J., Feinstein, H.L., Sandhu, R., & Youman, C.E. (1996). Role-based access control models. IEEE Computer, 29(2), 38–47.

Damiani, E., Paraboschi, S., & Vimercati, S. (2002). A reputation-based approach for choosing reliable resources in peer-to-peer networks. In the Proceedings of the ACM Conference on Computer and Communications Security (pp. 207–216). New York: ACM.

Deng, R.H., Gong, L., Lazer, A.A., & Wang, W. (1996). Practical protocols for certified electronic mail. Journal of Network and System Management, 4(3), 279–297.

Donovan, J. (2002). U.S.: Congress probes FBI And CIA intelligence failures. RadioFreeEurope. Retrieved July 27, 2003, from http://ww.rferl.org/nca/features/ 2002/06/05062002152908.asp

Ewald, T. (2002). The web services idea. Microsoft Corporation. Retrieved July 27, 2003, from http://msdn.microsoft.com/webservices/understanding/readme/default.aspx

Federal CIO Council. (2002). E-gov enterprise architecture guidance (common reference model). FEA Working Group. Retrieved July 17, 2003, from http://www.cio.gov/documents/E-Gov_Guidance_July_25_Final_Draft_2_0a.pdf

Federal Enterprise Architecture Program Management Office. (2002a). The business reference model version 1.0: A foundation for government wide improvement. Federal Architecture Program Management Office. Retrieved from http://www.feapmo.gov/resources/fea_brm_release_document_rev_1.pdf

Federal Enterprise Architecture Program Management Office. (2002b). The business reference model version 1.0: Agency Briefing. Federal Architecture Program Management Office. Retrieved July 27, 2003, from http://www. feapmo.gov/fea_downloads.htm

Federal Enterprise Architecture Program Management Office. (2002c). Draft business reference model common response document. Federal Architecture Program Management Office. Retrieved July 27, 2003, from http://www. feapmo.gov/fea_downloads.htm

Federal Enterprise Architecture Program Management Office. (2002d). Solution architects working group: Overview of missions and objectives. Federal Architecture Program Management Office. Retrieved July 27, 2003, from http://xml.gov/presentations/sawg/overview.ppt

Federal Enterprise Architecture Program Management Office. (2002e). Twenty four presidential priority e-govt initiatives. Federal Architecture Program Management Office. Retrieved July 27, 2003, from http://www.feapmo.gov/fea.htm

Franklin, M.K., & Reiter, M.K. (1997). Fair exchange with a semi-trusted third party. In R. Graveman, P. Janson, C. Neumann, & L. Gong (Eds.), the Proceedings of the 4th ACM Conference on Computer and Communications Security (pp. 1–7). New York: ACM.

French, M. (2003). CIA, FBI wrangle over threat center. Federal Computer Week. Retrieved July 27, 2003, from http://www.fcw.com/fcw/articles/2003/0421/ web-ciafbi-04-23-03.asp

Gasser, M., & McDermott, E. (1990). An architecture for practical delegation in a distributed system. In the Proceedings of the IEEE Symposium on Research in Security and Privacy. Piscataway, NJ: IEEE.

Goldreich, O. (2001). Secure multi-party computation. Working draft, version 1.3. Retrieved from http://www.wisdom.weizmann.ac.il/~Oded/

Gollmann, D., & Zhou, J. (1996). A Fair Non-Repudiation Protocol. In the Proceedings of the IEEE Symposium on Research in Security and Privacy (pp. 55–61). Piscataway, NJ: IEEE.

Grosof, B.N., Feigenbaum, J., & Li, N. (2003). Delegation logic: A logic-based approach to distributed authorization. Proceedings of ACM Transactions on Information and Systems Security, 6(1), 128–171.

Hailes, S., & Rahman, A. (1997). A Distributed Trust Model. In T. Haigh & B. Blakely (Eds.), the Proceedings of New Security Paradigm Workshop, ACM. New York: ACM.

Hess, A., Jacobson, J., Jarvis, R., Seamons, K.E., Smith, B., Yu, L. Yu, T., & Winslett, M. (2002). Negotiating trust on the web. IEEE Internet Computing 6(6), 30–37.

Hoque, F. (2000). e-Enterprise: Business models, architecture, and components. New York: Cambridge University Press.

Jarrell, D. (2002) The value of information sharing. Federal Computer Incident Response Center. Retrieved July 27, 2003, from http://www.fedcirc.gov

Jefferies, N., Mitchell, C., & Walker, M. (1995). A proposed architecture for trusted third party services. In E. Dawson & J. O. Golie (Eds.), Cryptography Policy and Algorithms Conference. Springer LNCS (Vol. 1029, pp. 98–104). Heidelberg, Germany: Springer.

Josang, A. (1996). The right type of trust for distributed systems. In T. Haigh, B. Blakey, M. E. Zurbo, & C. Meadows (Eds.), the Proceedings of New Security Paradigm Workshop, ACM. New York: ACM.

Josang, A. (1999). An algebra for assessing trust in certification chains. In S. Kent & G. Tsudik (Eds.), The Proceedings of Network and Distributed Systems Security Symposium. Reston, VA: The Internet Society.

Liu, L., & Xiong, L. (2003). A reputation-based trust model for peer-to-peer e-commerce communities. In the Proceedings of 4th ACM Conference on Electronic Commerce (pp. 228–229). New York: ACM.

Naor, M., & Nissim, K. (2001). Communication preserving protocols for secure function evaluation. In The Proceedings of the ACM Symposium on Theory of Computing (pp. 590–599). New York: ACM.

Naor, M., Pinkas, B., & Summer, R. (1999). Privacy preserving auctions and mechanism design. In The Proceedings of the 1st ACM Conference on Electronic Commerce (pp. 129–139). New York: ACM.

Open Resource Group. (1999). Technical reference model in detail. The Open Group. Retrieved July 27, 2003, from http://www.opengroup.org/architecture/ togaf7/presents/togaf_ovu.pdf

Osborn, S.L. (2002). Integrating role graphs: A tool for security integration. Data and Knowledge Engineering, 43(3), 317–333.

Seamons, K.E., Winslett, M., & Yu, T. (2001, February). Limiting the disclosure of access control policies during automated trust negotiation. Paper presented at the Network and Distributed System Security Symposium, San Diego, CA.

Skonnard, A. (2002). Understanding XML namespaces. DevelopMentor. Retrieved July 27, 2003, from http://msdn.microsoft.com/msdnmag/issues/01/07/xml/ default.aspx

Tygar, J.D. (1998, August). Atomicity versus anonymity: Distributed transactions for electronic commerce. Paper presented at the 24th VLDB Conference, New York.

Ulanoff, L. (2002). An insider's look at homeland security and technology. PC Magazine. Retrieved July 27, 2003, from http://www.pcmag.com/article2/ 0,4149,652467,00.asp

Wolter, R. (2001). XML web services basics. Microsoft Corporation. Retrieved July 27, 2003, from http://msdn.microsoft.com/library/default.asp?url=/library/ enus/dnwebsrv/html/webservbasics.asp

Yao, A.C. (1986, October). How to generate and exchange secrets. Paper presented at the 27th Annual Symposium on Foundations of Computer Science, Toronto, Canada.

York, S. (1997). Battling the cyberthreat. Wabash Magazine. Retrieved July 27, 2003, from http://www.wabash.edu/magazine/1997/fall/features/face_of_conflict/cyber_threat.htm