TRUST MEDIATION FOR DISTRIBUTED INFORMATION SYSTEMS

Brian Toone, Michael Gertz, Premkumar Devanbu
Department of Computer Science, University of California, Davis
{toone, gertz, devanbu}@cs.ucdavis.edu

Abstract: Distributed information systems are increasing in prevalence and complexity as we see an increase in the number of both information consumers and information providers. Applications often need to integrate information from several different information providers. Current approaches for securing this process of integration do not scale well to handle complex trust relationships between consumer applications and providers. *Trust mediation* is a technique we introduce to address this problem by incorporating a model for representing trust into a framework for retrieving information in a distributed system. Our model for representing trust uses a type system by which data from a source is labeled with a trust type based on qualities of the data itself or the information source(s) providing the data. With this model we develop algorithms to perform static analysis of data queries to infer how the result of the data query can be trusted. We describe an enhanced mediation framework using this inference technique that enables the mediator to govern the flow of information to match intended trust policies in large distributed information systems, even when information may originate from many heterogeneous sources.

Key words: Trust mediation, trust model, scalable information security, data integration

1. INTRODUCTION

This paper is concerned with *trusted, distributed information systems*, where clients pose questions that can only be answered by combining information from several different sources. If clients plan to use this information in ways critical to their business or property, how can they gain confidence in the information to be retrieved and used? Information relevant to a client

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35691-4_52
D. Gritzalis et al. (eds.), *Security and Privacy in the Age of Uncertainty*

query may occur in multiple sources of variable quality and/or access costs. The following settings illustrate the dimensions of this problem.

- Consider an investor attempting to decide whether to invest money in a business. She may need information about the market the business sells into: revenue/sales information about the company; stock price history; qualifications of the management team; and (given current events) possible criminal history and/or financial misdeeds of the other major stakeholders.
- A doctor with a large, changing, and ethnically diverse set of patients, confronted with a client presenting unusual symptoms, may wish to know: the past history of the patient; types of congenital problems members of the client's ethnic group are known to have; and drugs that have undesirable interactions with the medication the patient currently uses.
- A user desiring to install a new piece of software may want to know: if any software currently installed on his machine will induce a library version conflict [1]; the set of known defects and vulnerabilities in the new software; what other libraries and packages he needs to download to use the new software.

In the above settings, we see a large, potentially diverse set of users, interacting with many different information sources. In addition, there is a strong need for *trust* in the retrieved information; incorrect information may lead to service disruptions, or loss of life and/or property. A trusted distributed information system, then, provides clients with not just answers, but answers from trustworthy sources. There are several challenges to the proper design of a trusted distributed information system:

Scaling Trust Relationships Trust, as a social construct is at its core a pairwise relationship between a *trustor* and a *trustee*. The natural extension of this to information processing, where each client attempts to develop a trust relationship with an information source, leads to a quadratic scaling problem with m clients and n sources. We desire a solution that manages this scaling problem.

Diversity of Trust Clients may have different criteria by which they choose to assign trust to information sources. Brand names, ratings by bureaus, perceived commitment to information acquisition, investment in processing and networking capability, may all be factors. We desire an approach that does not inhibit diverse bases for trust relationships.

Trust Composition As illustrated above, a typical client query may require a diversity of information, which in turn may imply a diversity of sources that may not be uniformly trusted. In the presence of differing trust levels, the architecture should be able to compose information at different trust levels, and still give the client an indication of the trustworthiness of the computed answer. If it were not possible to develop uniform rules to combine information at different trust levels, then the client would have to manage this in an ad-hoc and inherently non-scaleable fashion.

Our paper makes several contributions to these problems. First, we introduce the *trust mediation* architecture, which introduces the notion of trust into traditional mediated architectures. This architecture separates the concern of providing trust ratings into separate elements, thus providing a means of scaling the management of trust relationships. Second, the architecture uses a flexible, open notion of *trust types* which allows "trust" to be defined in different ways in different contexts. Finally, for specific, and (we believe) intuitively appealing and important contexts, we provide a formal *static trust typing* system which allows types for queries to be inferred, even before the queries are evaluated; query types are inferred based on trust ratings for primitive relations provided by elements of the trust mediation architecture.

The paper is organized as follows: Section 2 discusses related work. Section 3 gives an overview of our conceptual framework for implementing the trust management component and outlines our formal model for representing the trust that is being managed by our framework. Then in Section 4 we describe our algorithms for processing trust, which forms the core of the trust-enhanced mediation component. Finally, in Section 5 we summarize our approach and describe future work.

2. RELATED WORK

Trust mediation has been studied extensively under the name *secure mediation*, addressing the problem of managing client credentials to ensure that a mediator does not violate information source security policies when integrating data to satisfy client queries [2-6]. We address an orthogonal problem of ensuring that data integrated from multiple sources meets client requirements in terms of trustworthiness of the sources used to provide information. In [7], a model is described for secure mediation that somewhat resembles our approach to trust mediation. The model includes the notion of "characterizing properties" assigned by trusted authorities to client and source entities. We take a similar approach except we establish "trust types" and "trust requirements" akin to their notion of "characterizing properties".

In addition, we establish mediation algorithms that take into account the interaction of properties when considering the trustworthiness of the result of data integrated from multiple, differently trusted sources. Secure mediation research does not typically address such interaction because security in this context is related to *independent* access by clients to multiple sources. Because the trustworthiness of sources is not considered, one only has to address the properties of a single client when determining whether access to sources should be allowed. This is not the case for trust mediation, where we

consider different levels of trustworthiness when integrating data from multiple sources simultaneously.

To a certain extent our proposed framework for trust mediation resembles the concept of utilizing data quality information to improve data integration [8]. In our approach, however, we present a decoupled architecture for assigning and managing trust information. Our proposed architecture provides better scalability to account for trust relationships between clients and large numbers of heterogeneous sources.

Trust management is a major component of our framework. PGP [9] is a way of managing trust via certificate chains. Policymaker [10] is an example of a general purpose framework for trust management. When applied to our environment, it is subject to both trust scalability and trust diversity problems.

3. TRUST MEDIATION

This section describes our framework for trust mediation. Trust mediation can be divided into two functional components: trust management and trust-enhanced mediation. *Trust management* describes the process of maintaining some quantitative or qualitative measure of trust relationships among trustors and trustees. The *trust-enhanced mediation* component uses trust metadata to govern information flow to clients from sources. We address each of these components, in turn, in Section 3.1 and 3.2.

3.1 Conceptual Architecture

We build our trust mediation framework by extending the infrastructure typical of a mediated query system (MQS) (see, e.g., [11] for an overview). Multiple heterogeneous sources supply information to multiple clients. A mediator or collection of mediators connects clients to sources by integrating information from sources to satisfy client queries. Figure 1a shows the query/response interaction among components in a typical MQS. Lighter arrows indicate queries. Darker arrows indicate responses.

Figure 1b highlights the many trust relationships in a typical MQS. The dashed arrows in Figure 1b point from trustor to trustee. For example, clients trust the mediator to decompose submitted queries correctly (t_{decomp}) and to integrate information returned from sources correctly (t_{comp}). Clients also trust sources to provide correct, reliable information. Conversely, sources trust clients to use data in a non-malicious way ($t_{gooduse}$). In a competitive market model where multiple mediators may exist, mediators trust sources to

provide correct ($t_{correct}$), reliable ($t_{reliable}$) information because incorrect information reflects poorly on the mediator.

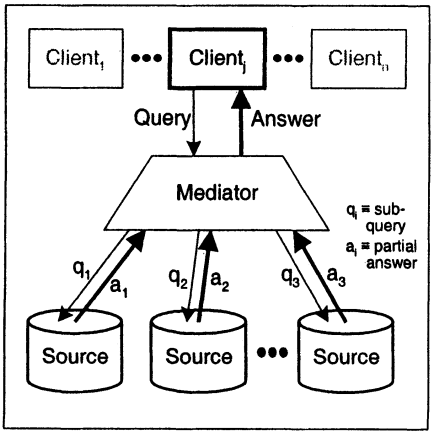

Figure 1a. Mediated Query System (MSQ) component interactions.

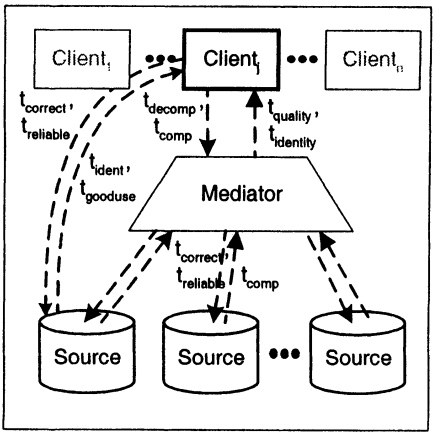

Figure 1b. Implicit trust relationships in a MQS. Arrows point from trustor to trustee.

Our framework for trust mediation provides a mechanism to specify and take these trust relationships into account in the design of a MQS. Depending on the type of information sources, however, specifying and accounting for every trust relationship shown in Figure 1b may be unnecessary to achieve the desired characteristics of a trustworthy distributed information system. For example, a system where information providers exist in the public domain obviates the need for analysis of the trustworthiness of clients. Clients cannot misuse the system to obtain access to restricted information because no such provider exists. Similarly, the trustworthiness of mediators can be assumed when the mediator exists within the same administrative and security domains as clients accessing the system. For simplicity, we will assume that the trustworthiness of mediators and clients is either irrelevant or handled outside of our trust mediation framework – e.g. use existing secure mediation techniques such as those found in [7].

With many clients and numerous information sources, mediators now are in the untenable position of tracking a quadratically growing number of trust relationships in addition to their normal data integration tasks. Therefore, we propose to separate the concern of "trust management". It is not feasible to program the mediator with a different global schema for each client to satisfy individual trust requirements. Figure 2 shows a high level view of our approach. *Trust authorities* analyze sources and assign trust ratings based on precise trust definitions. A trust authority may be an actual external entity such as the Better Business Bureau or a conceptual component consisting of

a network of clients willing to share their expertise and experience interacting with a source in order to establish a trust rating for a source. Whatever the implementation, ratings assigned by trust authorities are stored in a *trust broker*, which the mediator accesses during processing of client queries.

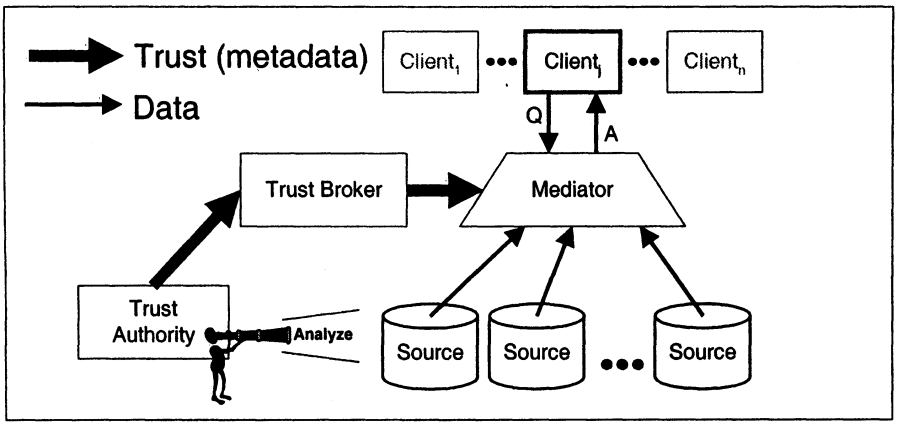

Figure 2. Conceptual architecture for trust mediation.

The operation of our architecture begins when clients submit queries to a mediator. Clients may also attach trust requirements to the submitted queries. The mediator determines multiple query plans for the client query based on the global (mediated) schema. It is important to note here that multiple query plans exist because data may be duplicated in several sources. Recall that the trustworthiness of these sources may be different and thus the result of query plan execution may not be identical as is typical in a mediated query system. This is accounted for by the client through the specification of trust requirements. To select a query plan for execution, the mediator processes the trust ratings stored in the trust broker for the sources specified in each query plan. The result of this processing step is trust ratings for the integrated data that would be returned to the client for each executed query plan. The trust processing algorithm we present for performing this static analysis of query plans to determine trustworthiness of the integrated result is detailed in Section 4. Operation of the framework continues as the mediator executes a query plan whose trust rating satisfies the client trust requirements. The retrieved data is then sent to the client. The mediator notifies the client if no query plan satisfies the client trust requirements.

3.2 Trust Model for Trust Mediation

Before describing the details of assigning trust ratings, formulating trust requirements, and processing trust, we present in this section our model for

representing trust. We describe more clearly the notion of trust we use for trust mediation by defining the relationship that exists when a trustor trusts a trustee. Our notion of trust is based on expectations similar to those expressed in [12], where trust is defined as "a belief in the system characteristics, specifically belief in the competence, dependability and security of the system, under conditions of risk." We adapt this definition for our setting:

Definition 1. *Trust* in the context of distributed information systems is the belief held by a trustor that expectations will be met regarding specific observable attributes of a trustee and/or the information the trustee provides.

In the framework that we have presented, trust authorities act on behalf of clients (trustors) to evaluate and establish trust ratings for information sources (trustees) according to this definition of trust. To accommodate a real distributed information system where clients may trust information in multiple ways depending on its intended use, we add granularity to our definition by introducing the notion of a *trust type*.

Definition 2. A *trust type* specifies the attributes and expected values of those attributes, which are measured to decide whether a trustee is trusted.

Definition 2 gives us the ability to define precisely what it means for a trustor to trust a trustee. The attributes specified in a trust type could be related to the information source or, alternatively, to the information content – even at the granularity of actual data values. The four example trust types given in Figure 3 take this latter approach.

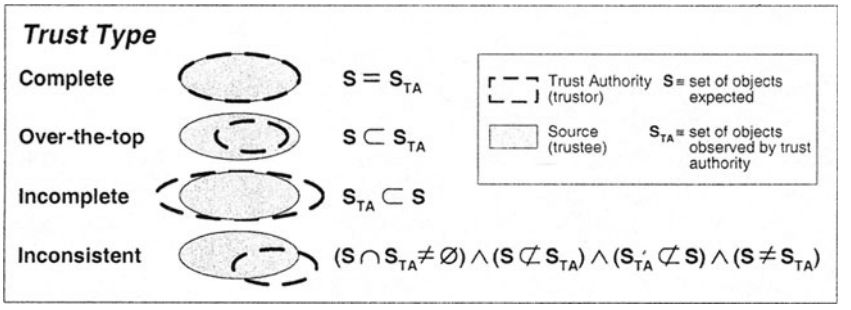

Figure 3. Four example trust types with definitions using set notation.

The trust type "complete" indicates that the trustor (e.g., trust authority) believes that the trustee (e.g., information source) will provide exactly the complete set of relevant objects as indicated by the Venn diagram. Likewise, the trust type "incomplete" indicates a trust authority believes that, when queried, an information source will provide only a subset of relevant objects; but not the complete set. The trust type "over-the-top" indicates a belief that

an information source will provide at least one irrelevant object in addition to the complete set of relevant objects. Lastly, an information source trusted according to the "inconsistent" trust type is believed to be capable of providing at least one relevant object. The trustor, however, believes that the "inconsistent" information source will not be able to provide the complete set of relevant objects and will also provide at least one irrelevant object.

To see the applicability of these example trust types to a distributed information system, consider a client medical application which queries for medications to treat a disease. Obviously, the client would be satisfied by a "complete" answer that returns all relevant medications. Depending, however, on the intended use for the information (i.e., the type or severity of disease), the client may be willing to settle for an "incomplete" answer if the cost for retrieving a "complete" answer is significantly higher.

The example trust types in Figure 3 require a trust authority to know a complete data set in order to assign a rating. Despite having such knowledge, a trust authority may not be configured to answer client queries for information directly. Recall that the trust authority may only be conceptual –as described previously where the trust authority consisted of a set of clients sharing their experiences interacting with sources. This conceptual trust authority cannot answer queries; it can only assign trust ratings based on collective knowledge of what the information source should contain.

Trust type definitions are established for a specific application domain. There are a variety of ways this can be accomplished: a distributed information systems (or mediator, i.e., value-added mediation) design team may create relevant trust type definitions for the application domain along with the appropriate inference rules for trust mediation. Trust authorities and clients use these definitions to assign trust ratings and specify trust requirements. Alternatively, trust authorities may independently develop trust types and publish these definitions for clients to use when specifying trust requirements. The systems designer would be responsible for developing the appropriate inference rules for the given trust types, which would then be used in our trust mediation algorithm as discussed in Section 4.

Once trust type definitions have been established, trust authorities analyze sources to observe the values of the properties specified in a particular trust type definition. Figure 4 gives an overview of this process. The analysis process could involve a general review of the qualities of the information source or utilize a series of key query probes to determine the consistency of the probe result with expected values according to the trust type definition. If the analysis matches the values specified in the trust type definition, the trust authority labels the source with the corresponding trust type and records this trust rating in the trust broker. A *trust rating*, then, is a trust type assigned to a source by a trust authority indicating that the source can be trusted ac-

cording to that particular trust type. Multiple trust authorities may assign different trust ratings to identical sources if the technique used to evaluate sources differs among trust authorities.

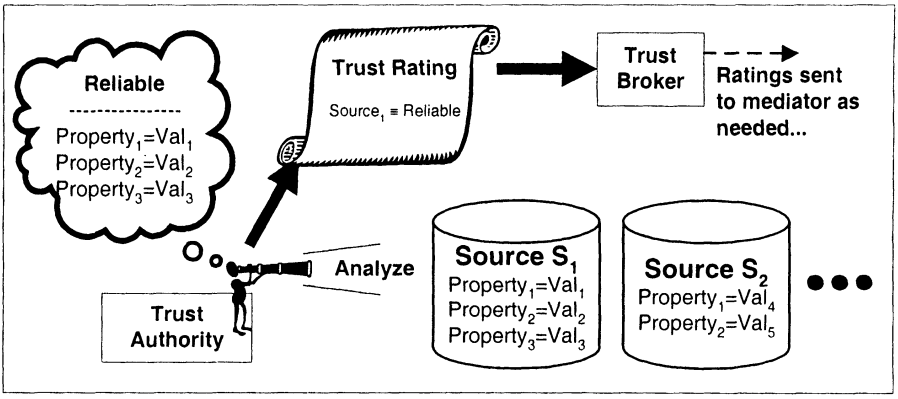

Figure 4. Overview of trust rating assignment process.

We handle this potential inconsistency by assuming that trust authorities agree upon a rating before it is stored in the trust broker. Even if we relax this assumption, we could alter our mediation algorithm to take a pessimistic or optimistic approach to resolving inconsistencies. In the example shown in Figure 4, a trust authority analyzes two sources S_1 and S_2 using the trust type "reliable". In S_1, values of the relevant properties match exactly those given by the trust type definition. In S_2, the values do not match exactly. Therefore, the trust authority assigns a trust rating of "reliable" to S_1 only and sends this rating to the trust broker. The trust authority may continue to evaluate these sources and other sources using different available trust type definitions.

4. ALGORITHM FOR TRUST MEDIATION

We now describe the algorithm that the mediator uses to combine trust requirements and trust ratings during the processing of client queries. Using the trust typology described in the previous section, we employ a technique for performing static trust typing of the query plan analogous to static typing in programming languages. In a statically typed programming language, the type of the result of an operation can be calculated statically by examining the types of the operands and applying a set of type inference rules to the operation. We take the same approach for our trust mediation algorithm. The operations of interest are the operations of a typical mediator query plan, i.e., the relational algebra. All query plans can be specified in terms of these operations. Therefore, we have developed a set of inference rules for these ba-

sic operations which can be applied recursively to the operations specified in a query plan in order to calculate the trust type of the result. It is important to note that the operations of the relational algebra have well-defined semantics in terms of set theory. Defining trust types using set theory allows us to prove soundness of our algorithms by demonstrating that trust type labeling applied to the result of an integration operation produces the same trust rating as inferred statically from the input ratings.

Recall the basic operation of our framework. A client formulates both a query based on the mediated schema and a meta-query (trust requirements) using published trust type definitions and sends both to the mediator. The mediator formulates one or more query plans based upon the global schema and then applies our algorithm for trust mediation to the query plan selected for exection to determine the trust rating for the result without having to execute the query plan. Consider generic query plan:**qp** = operand$_1$ *op* operand$_2$.

The operands can be sources (relations) or the result of nested query plans that produce some portion of the total answer for the client query. Our trust mediation algorithm applied to **qp** follows:

trust_mediation algorithm (qp){
  if operand$_1$ is a source
    let t_operand$_1$ = trust rating for source (look up in trust broker)
  else // operand must be a nested query plan
    let t_operand$_1$ = **trust_mediation algorithm** (operand$_1$)

  if operand$_2$ is a source
    let t_operand$_2$ = trust rating for source (look up in trust broker)
  else // operand must be a nested query plan
    let t_operand$_2$ = **trust_mediation algorithm** (operand$_2$)

  return inference_rule(*op*, t_operand$_1$, t_operand$_2$)
}

The last step of the mediation algorithm shown above is a simple table lookup indexed by the operation and operand ratings. Table 1 gives example inference rules along with abbreviated soundness proofs. For example, Inference Rule #1 states that if source $\mathbf{S_1}$ has a trust rating of "complete" (C) and source $\mathbf{S_2}$ has a trust rating of "complete" (C), the result of an intersection operation using $\mathbf{S_1}$ and $\mathbf{S_2}$ will also have a trust rating of "complete". The intuition behind the proof is that the trust rating for the result of the integration operation, if calculated a posteriori given the trust type definition for "complete" shown in Figure 3, will be identical to that produced by our inference rule. To demonstrate completeness, we have developed a complete set of axioms for each of the operations and for each combination of trust rating for the input relations which are included in [13]. For brevity here, we include the entire set of rules in [13]. We claim that they are easily derived from the definitions of trust types and relational algebra operators.

Table 1. Sample inference rules and abbreviated soundness proofs.

Inference Rule	*Proof*	
Inference Rule #1: $\frac{S_1:C \ , \ S_2:C}{S_1 \cup S_2:C}$	1. $S_1 = S_{1,TA}$	"Def. of complete"
	2. $S_2 = S_{2,TA}$	"Def. of complete"
	3. $\{S_1 \cup S_2\}=\{S_{1,TA} \cup S_{2,TA}\}$	"Def. of union"
	4. $S_1 \cup S_2$ is C.	"Def. of complete"
Inference Rule #2: $\frac{S_1:C \ , \ S_2:O}{S_1 - S_2:C \ or \ I}$	1. $S_1 = S_{1,TA}$	"Def. of completef"
	2. $S_{2,TA} \subset S_2$	"Def. of over-the-top"
	3. $\{S_1 - S_2\} \subseteq \{S_{1,TA} - S_{2,TA}\}$	"Def. of difference"
	4. $S_1 - S_2$ is C or I.	"Def. of complete, incomplete"

Consider again a medical scenario. A client application queries a mediator for non-interacting disease medications. Source S_1, rated complete, provides chemotherapy medications. Source S_2, rated complete, provides non-chemotherapy medications. Source S_3, rated over-the-top, provides drug interaction data. To respond to the client query, the mediator generates the query plan **qp** to produce answer **A:** $\mathbf{A = (S_1 \cup S_2) - S_3}$. Applying our inference algorithm, even without performing the query, the mediator can conclude that the data will either be *complete* or *incomplete*. If the client application were interested giving initial advice to a patient, an incomplete result might be acceptable. But if the client application is being used by a doctor who has a patient that has tried many medications unsuccessfully, the doctor needs a complete result. In most cases, an over-the-top result would be unacceptable as potential for harm through drug interactions could be life-threatening. Based on the particular client trust requirement specified, the mediator can decide whether to incur the expense of executing the query plan or inform the client that an answer is not possible with the specified trust requirements.

5. CONCLUSIONS AND FUTURE WORK

In this paper we introduced a framework for managing trust relationships between clients and information sources. We have shown our type-inference style approach for analysis of information sources whereby trust ratings are used to derive a cumulative trust rating for the result of a query. As with static typing, this inference is performed prior to query evaluation, which allows a mediator to determine if the result of a query will satisfy client trust requirements, without committing to the expense of query evaluation. We are currently developing a prototype of our trust type inference engine using Prolog. We are using the Amzi! Logic Server which provides necessary tools for embedding Prolog in Java [14]. Our goal for this prototype is to demon-

strate the feasibility and benefits of deployment into Willow – a distributed information system for survivability through reconfiguration [15]. Future work will focus on enhancements to our trust model to accommodate additional trust types and inference rules.

Acknowledgements

Brian Toone's work was supported in part by a United States Department of Education Government Assistance in Areas of National Need (DOE-GAANN) grant #P200A980307. Michael Gertz and Premkumar Devanbu gratefully acknowledge support from the NSF ITR Program, Grant No. 0085961. This material is also based in part upon work sponsored by SPAWAR and the Defense Advanced Research Projects Agency under Contract Number N66001-00-8945. The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

References

[1] P. Devanbu, "The ultimate reuse nightmare: honey, I got the wrong DLL," in Proc. of the 5th Symposium on Software Reusability (SSR '99), 1999, ACM pp. 178-180.

[2] J. Biskup, U. Flegel, and Y. Karabulut, "Secure mediation: requirements and design," in Proc. Int'l Working Conf. on Database Security.: DBSec 1999, Kluwer pp. 127-40.

[3] S. Dawson, S. Qian, and P. Samarati, "Providing security and interoperation of heterogeneous systems," Distributed and Parallel Databases, vol. 8, pp. 119-45, 2000.

[4] Y. Karabulut, "Credential management for secure mediators," presented at 11th GI-Workshop Grundlagen von Datenbanken, Thüringen, Germany, 1999.

[5] K. S. Candan, S. Jajodia, and V. S. Subrahmanian, "Secure mediated databases," in the 12th Int'l Conf. on Data Engineering, IEEE Computer Society 1996, pp. 28-37.

[6] S. Dawson, S. Qian, and P. Samarati, "Secure interoperation of heterogeneous systems: a mediator-based approach," in the IFIP 14th Int'l Conf. on Information Security, 1998.

[7] J. Biskup and Y. Karabulut, "A hybrid PKI model with an application for secure mediation," in Proc. of 16th Annual IFIP WG11.3 Working Conference on Data and Application Security, July 2002.

[8] F. Naumann, U. Leser, and J. C. Freytag, "Quality-driven integration of heterogeneous information systems," in Proc. of the 25th Int'l Conf. on VLDB, Morgan Kaufmann, 1999.

[9] S. Garfinkel, PGP: pretty good privacy. O'Reilly & Associates, 1995.

[10] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in IEEE Symposium on Security and Privacy, IEEE Computer Society, 1996.

[11] R. Domenig and K. R. Dittrich, "An overview and classification of mediated query systems," SIGMOD Record, vol. 28 no. 3, pp. 63-72, 1999.

[12] A. Kini and J. Choobineh, "Trust in electronic commerce: definition and theoretical considerations," in 31st Hawaii Int'l Conf. on System Sciences, IEEE Computer Soc., 1998.

[13] B. Toone, "Inference rules for trust mediation," Dept. of Comp. Sci., Univ. of Calif. Davis, http://wwwcsif.cs.ucdavis.edu/~toone/research/trust/inference.html, 2002.

[14] M. Kroening, "Java meets Prolog for advisors, analysts and agents," PC AI, vol. 10, pp. 27-31, 1996.

[15] J. Knight, et. al., "The willow survivability architecture," in Proceedings of the Information Survivability Workshop, IEEE Computer Society, 2001.