COLLABORATION SECURITY FOR MODERN INFORMATION SYSTEMS

Richard Whittaker
Florida International University
11200 S.W. 8th Street, Miami, FL 33199

Gonzalo Argote-Garcia, Peter J. Clarke, Raimund K. Ege
Florida International University
11200 S.W. 8th Street, Miami, FL 33199

Keywords: Security Architecture, Personal Data Protection, Agent Security, Mediation System.

Abstract: One of the main approaches to accessing heterogeneous data is via the use of a mediation framework. The current problem with mediation systems is that they are viewed as black boxes from the perspective of their clients. As clients enter their data, they are unable to control the access to their data from entities within the mediation system. In this paper we present a solution in the form of a security framework, named Collaboration Security Framework that addresses the needs of all entities, i.e. external clients, mediators or data sources, to have autonomy in applying security policies during collaboration. As a result all entities participating in a collaboration have control over the access to their data by applying local, global and collaboration channel security rules, which can be changed at runtime and that are security model independent.

1 INTRODUCTION

As Information Systems change the landscape of information delivery, even with their current advancements, most Information Systems still fall short on taking an active role in supplying data and knowledge to their clients (Wiederhold, 1992). An architecture for Information Systems that shows promise is the Three-Layer Mediation Architecture (Ege et al., 2004). When compared to Federated Systems (Sheth and Larson, 1990), Mediation Systems are better suited for environments, which tend to scale and where their information sources are dynamic (Park and Ram, 2004). Through the use of mediators within a Mediation System, the system can package and deliver information to the client in a way that makes the complexity of information processing transparent to the client.

However, current Mediation Systems are limited in addressing issues of enterprise environments that are concerned with security to their stakeholders during utilization of these systems (Bhatti et al., 2005). For these Mediation Systems to progress in these environments, new methods of providing security must be developed. One key requirement that must be met is that all entities collaborating within a Mediation System must have autonomy over their security. Currently this is not the case since Mediation Systems are viewed as black boxes from the perspective of their clients. As clients enter their data, they are unable to control the access to their data from entities within the mediation system. This places the clients in a vulnerable position since they have no knowledge of how, why or what part of their data is being processed during the collaboration with the Mediation System.

This lack of knowledge and control can be catastrophic in areas dealing with data sensitive transactions. In this paper we present the solution in the form of a security framework, named Collaboration Security Framework (CSF), that addresses the needs of all entities, i.e. external clients, mediators or data sources, to have autonomy in applying security policies during collaboration with each other. This security framework would provide the clients with a security wrapper that encapsulates their data. Our security framework is also independent of any particular security model e.g. BPL (Bell and Padula, 1975), Chinese Wall (Brewer and Nash, 1989), RBAC (Sandhu et al., 1996).

Note, our security framework is only a first line of defense. We do not address the problem of valid entities, who have the proper clearance to access the data, for passing information to non-valid third parties. If this happens, the client would only know a list of candidates that can be held accountable.

The contributions of this paper can be summarized

Whittaker R., Argote-Garcia G., J. Clarke P. and K. Ege R. (2006).
COLLABORATION SECURITY FOR MODERN INFORMATION SYSTEMS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 363-370
DOI: 10.5220/0002095803630370

as follows, we present a security framework that:

- Makes it possible for all entities involved in a collaboration to utilize a standard security mechanism that would provide each entity control over access to their data.
- Provides a security infrastructure that can apply security policies at a local and global level.
- Is security model independent.
- Can provide entities the ability to change their security policies at runtime.

Due to space limitations, we do not provide a proof of completeness, verification mechanism, protocols or security polices reconciliation of the CSF. These topics will be presented in our future work.

This paper is organized as follows: in the next section, we provide a brief overview of mediators and mobile agents. In Section 3 we look at security scopes, which are being applied during collaboration. Section 4, we present cross-domain collaboration and a key security violation that can arise during a cross-domain collaboration. We cover the components of the CSF in Section 5. An example applying the CSF to mediation and related work will be presented in Sections 6 and 7, respectively. We give the conclusion of the paper in Section 8.

2 PRELIMINARIES

In developing our security framework, we assumed that there is an architecture in place that can deal with the semantic and syntactic interoperability of heterogeneous data sources and also that there is an underlying distributed mobile agent environment.

Architectures that provide interoperability can be classified as follows: Mapping-Based, Query-Oriented and Intermediary Approach. Due to the drawbacks of federated schemas and the necessity for clients to have comprehensive understanding of the domain structure and data that are present in Mapping-Based and Query Oriented Approach (Park and Ram, 2004) respectively, we focus our security framework on Information Systems based on the intermediary approach utilizing mediation mechanisms.

Mediation Systems SAW (Dawson et al., 2000), TIHI (Wiederhold et al., 1996) and CHAOS (Liu et al., 2000) all provide semantic and syntactic interoperability of heterogeneous data sources through the use of mediators."These mediators are intelligent middleware that sit between information system clients and sources. They perform functions such as integrating domain-specific data from multiple sources, reducing data to an appropriate level and restructuring the results into object-oriented structures" (Wiederhold and Genesereth, 1997).

A key requirement for our security framework is for an entity involved in a collaboration to move from host to host within a mediation system during the collaboration. Mobile agents are designed for this purpose. Mobile agents can transport their state and execute code within a distributed network. Therefore, mobile agents allow us to create a distributed execution environment, which provide seven key benefits: reduce network load, overcome network latency, encapsulate protocols, execute asynchronously and autonomously, adapt dynamically, are heterogeneous and robust and fault-tolerant (Lange and Oshima, 1999).

For mobile agents to achieve these seven benefits, they need an environment that provides them with necessary infrastructure. Mobile agent environments KOaS (Bradshaw et al., 1995) and Cougaar (Thome et al., 2004) are two examples that provide this infrastructure. By utilizing mobile agents we are able to collaborate in a distributed manner instead of a centralized manner. This makes it possible to send representatives with the needed instructions to conduct a collaboration e.g. security rules and data, to a certain location within the mediation system to represent an entity that needs to collaborate with other entities.

We merge mediation and mobile agent technologies within our Collaborative Security Framework (CSF), which enables entities participating in a mediation transaction to control the access to their data by applying local, global and collaboration channel security rules, which can be changed at runtime. These security rules are applied to protect the internal data of mobile agents.

3 SECURITY SCOPE

As entities collaborate within a mediation system, which utilizes our CSF framework, these entities must adhere to two types of security scopes: global and local scopes. Global security scopes governed security policies that are applied to a group of entities. Whereas, local security scopes governed policies that individual entities apply during their collaboration with other entities. To set the stage for proper analysis, we give the following definitions:

- ***Root Entity:*** *is an entity that is not representing another entity within the collaboration.*
- ***Representative Agent:*** *is an entity the represents a root entity within a collaboration.*
- ***Local Security Policy:*** *is a set of security policies that govern the access to an entity. These polices are controlled by the entity.*
- ***Global Security Policy:*** *is a security policy that is applied to every representative agent with a partic-*

ular representative domain and is controlled by the collaboration root entity.

- ***Representative Domain:*** *is the set of representative agents and their local collaboration channels that represent a root entity in a particular collaboration.*

These definitions are illustrated in Figure 1. It can be seen from the figure that each entity within the representative domain has its own local security policy that governs the entity. Therefore, for representative agents to communicate with each other they must adhere to each other's local security policy.

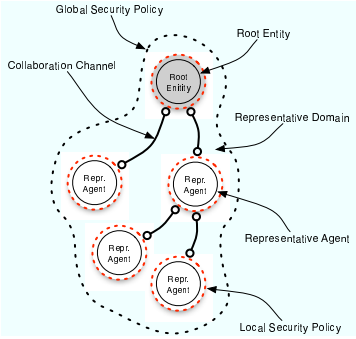

Figure 1: Representative Domain.

All communications between entities within a representative domain is done through a collaboration channel that we define as:

- ***Collaboration Channel:*** *is a bi-directional communication link between two entities that has a security policy that both entities agreed upon to establish a secure communication between them.*

A collaboration channel is established within a representative domain during the initialization of a representative agent, which is made possible by the services provided by the mobile agent environment. During this initialization, a representative agent is created to represent the parent who is creating the agent. Since the parent entity initializes the representative agent it will establish the security protocol between the parent and newly created representative agent.

Since representative agents have their security policies set by their parent during initialization, there may be times where these representative agents need to change their security policies. This can arise due to the environment that a representative agent is collaborating in, we will address this issue in Section 4. For a representative agent to change its security policy, it must request permission from its parent since an agent represents its parent. This request cycle is recursive and therefore, all requests will reach the root entity since it is the parent of all representative agents. Hence, the root entity is in control of all security policies within the representative domain.

As stated above the collaboration channel is bi-directional, therefore it is possible for a parent of a representative agent to request the representation agent to change its security policies and data. This may be necessary due to a runtime security policy adjustment.

To set the desired behavior and structure within a representation domain we establish the following properties:

- ***Root Entity Cardinality:*** *There is only one root entity within a representative domain.*
- ***Root Entity Collaboration Channels:*** *A root entity can only have collaboration channels linking it to representation agents who are in the same representative domain.*
- ***Representative Agent Cardinality:*** *A parent entity may have zero or more representative agents representing it within the representative domain.*
- ***Parent Cardinality:*** *A representative agent has only one parent entity.*
- ***Collaboration Channel Signature:*** *Every collaboration channel has a unique transmission signature.*

We assume all entities, i.e. root entity and representation agents within a representative domain, are working together to protect the root entity in collaborating with other representative agents that are located in different representative domains. Autonomy comes into play when entities within a particular representative domain collaborate with other entities within another representative domain. This cross-domain collaboration and its security issues will be addressed in the following section.

4 CROSS-DOMAIN COLLABORATION

During collaboration it is possible to have entities from different representative domains collaborate. This cross-domain collaboration is an area that current mediation systems are not addressing adequately. SAW (Dawson et al., 2000), TIHI (Wiederhold et al., 1996) and CHAOS (Liu et al., 2000) do not provide any cross-domain security mechanism for their clients. Security policies are geared to apply data filtering to the output data from the mediation system to the client. These systems do not provide any security mechanism for the clients to apply cross-domain security polices. Hence, contributing to the black box scenario from the perspective of the clients.

Current work by (Shehab et al., 2005) started to address the issue of cross domain security by enabling entities roles based on RBAC (Sandhu et al., 1996)

to cross into another domain with the possibility of changing the role of the entity within the new domain. This may affect the security policies of an entity since the newly assigned role may have a new set of security policies that pertain to this role. There is a drawback and an assumption about the collaboration environment that is not present within our environment. The drawback to their security framework is that the migrating entity has no control over the role that will be assigned to it in the new domain. Also, there is an assumption that a specific security model can be applied across all domains. This is not the case within our environment and in the real world. Our environment, models a more real world environment, where different organizations have different security models governing their security policies.'

For these organizations to collaborate there must be a path that enables data to flow across domain boundaries. How data flows across these domains is governed by the security policies and transmission infrastructure. We assume that the underlying transmission infrastructure is in place, mainly TCP/IP. The issue of interest is how to establish a cross-domain communication path between entities in different domains that adheres to each entity's security. To establish cross-domain communication, we utilize our security framework's collaboration channel that acts as a bridge between the collaborating representative domains. This is possible due to the services provided by the collaborative channel. Note, that this communication channel may have a different set of security policies than that of the collaborating entities. One can think of the collaboration channel, as the land buffer between two countries at a border that is not owned by either countries but has a policy governing the buffer that both countries agree on.

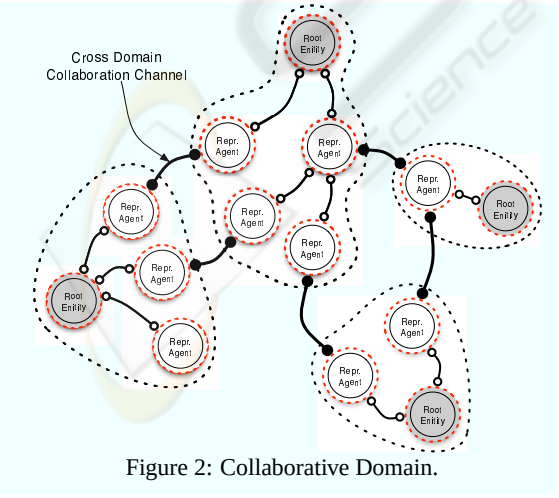

Figure 2: Collaborative Domain.

Figure 2 illustrates the collaborations between representative agents of different representative domains. Note that, the collaboration channels that are established between representative agents in these domains link representative domains. To initiate a cross-domain collaboration channel the representative agent must find a linking partner. This is possible through the use of directory services (Barker, 1995).

Once a cross-domain collaboration channel is established the two linked representative agents can start collaborating (request - response cycle). During their collaboration, they will only transmit their own local data, whose access is governed by their security policies. Therefore, even though there is a common consensus between the collaborating representative agents on their collaboration channel, each agent can still control the access to its data locally. It must be stated that our framework does not provide the interpolation of the security policies, but instead the CSF provides the topology for establishing the collaboration. Research of (Wallace, 1996), (Bistarelli et al., 1997) and (Bistarelli, 2004) have focused on security interpolation utilizing constraint programming and semirings.

4.1 Collaborative Security

Having established cross-domain collaborations we focus on the concept of security on these collaborations. What exactly is a secure collaboration and what is the scope of the security policies that govern this secure collaboration? We define a secure collaboration as follows:

- ***Secure Collaboration:*** *A cross-domain collaboration is secure if for every representative domain that has a collaboration channel to other representative domains all collaboration chains that one can use to traverse from a root entity to another root entity is secure.*

Therefore, secure collaborations are not controlled by a single governing body. Instead segmented partitions of security policies are governed by the global security policies of each representative domain and the local security polices of each representative agent. Each representative agent has its own autonomy over its local security policy. This gives the representative agent control over who is accessing their data and how. Hence, removing the black box scenario.

4.2 Collaborative Security Violation

Having established secure collaboration we must be aware of the security violations that can arise during a cross-domain collaboration. Figure 3 illustrates the classic transitive access leakage, where an entity was given access within a representative domain and it uses its access right to gain access to an unauthorized entity during utilizing improper path traversals.

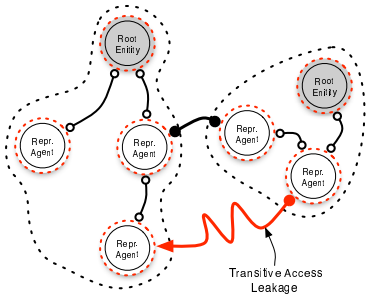

Figure 3: Transitive Security Violation.

It has been shown that, to check if a system based on an access right security mechanism is secure, i.e. that no entity gains unauthorized access during runtime, is undecidable (Harrison et al., 1976). Based on prior research (Gong and Qian, 1994) it has been proven that a cross-domain collaboration is secure is also undecidable. In special cases an optimal solution in polynomial time, Simplified Maximum-Access Secure Interoperation (Gong and Qian, 1996) can be used to check if there exists a violation during a cross-domain collaboration.

Therefore, if the security rules have the proper restriction set on them, a system based on the CSF framework may be checked to see if the system is secure in polynomial time. Note, the CSF is not designed around a specific security model. The CSF provides services to apply security during a cross-domain collaboration. Hence, mediation systems that incorporate the CSF into their architecture may use the Simplified Maximum-Access Secure Interoperation (Gong and Qian, 1996) to check if the system is secure. This is possible, since the CSF does not provide any service to modify any security rule within the mediation system.

5 COLLABORATION SECURITY FRAMEWORK

In this section we present key components of the Collaboration Security Framework (CSF) that provides the underlying structure for all entities involved in a collaboration to utilize a standard security mechanism that gives each entity control over accessing its data. To make this possible the CSF provides the ability for root entities and representative agents to apply security policies at local and global views that are security model independent and also can be changeable at runtime. One may find similarities between CSF and Kerberos (Steiner et al., 1988) where clients trust the identity judgement of entities within the CSF and Kerberos. A key different between CSF and Kerberos, is that in Kerberos a client requesting a service must identity itself through its credentials. Whereas, in CSF, a service is provided if there is a proper security interpolation between different Representative Domains.

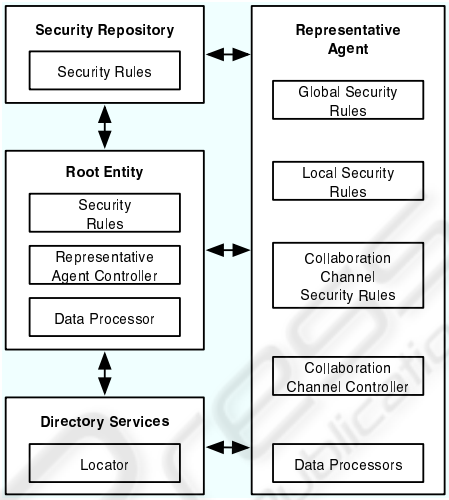

Figure 4: Collaborative Security Framework.

An overview of the CSF is illustrated in Figure 4, which is composed of four main components: *Security Repository, Root Entity, Directory Services and Representative Agent.* The CSF is a simple framework that achieves its underlying objective. In the following sub-sections we briefly overview each component and its modules, helping in understanding how the CSF achieves its goals in the deployment example given in the next section.

Security Repository

The Security Repository holds all the security models and rules that will be utilized by all entities within the representative domains.

- ***Security Rules:*** *This module is responsible for providing the security rules that are used by root entities and representative agents to build their security policies.*

Root Entity

The Root Entity is the entity that has control on how its data is being accessed during a collaboration. The root entity utilizes representative agents as delegates to achieve this task.

- ***Security Rules:*** *This holds the root entity security rules that are global and that proceed over the root entity's representative domain.*
- ***Representative Agent Controller:*** *This module is used to control the representative agent that represents the root entity within a representative domain.*

- ***Data Processor:*** *Processes the internal data incoming and out-going.*

Directory Service

The directory service component is used by root entities and representative agents to locate entities that can allow the collaboration to proceed.

- ***Locator:*** *Locates entities within the mediation system. Entities can query the locator to find an entity from other representative domains that can advance the collaboration.*

Representative Agent

The representative agent represents the root entity within its representative domain. It collaborates on behalf of the root entity with other representative agents that may or may not be in its representative domain.

- ***Global Security Rules:*** *The security rules of the representative agent's root entity. They are the global security rules of the representative domain.*
- ***Local Security Rules:*** *The security rules that are unique to the representative agent, which may be requested to be changed by the representative agent or by the representative agent's parent.*
- ***Collaboration Channel Security Rules:*** *The security rules that govern the incoming and out-going transmission data of the representative agent.*
- ***Collaboration Channel Controller:*** *This module establishes the creation and termination of collaboration channels.*
- ***Data Processor:*** *Processes the internal data, incoming and out-going.*

6 APPLYING CSF TO MEDIATION

In this section we present a deployment example of a mediation system based on the CSF framework and show how the mediation system gives each entity access control over its data and security policies at a local and global view that are security model independent and can be changed at runtime.

Figure 5 illustrates a runtime snap shot of the mediation system. As shown, there are six representative domains with their corresponding root entities in gray. One in the Application Layer whose representative domain spans from the Application Layer into the Mediation Layer. This is a key difference from other mediation systems, which is made possible by the underlying agent environment. Within the Mediation Layer there are three representative domains that

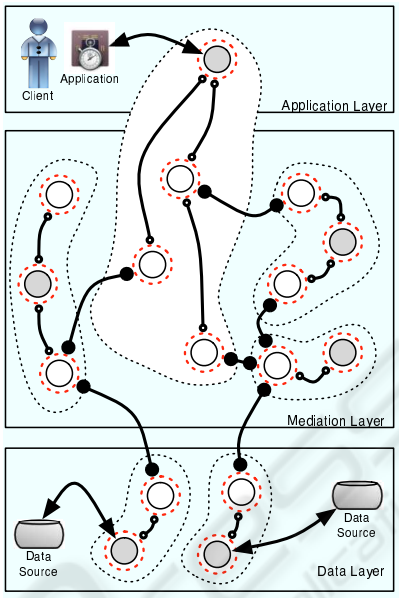

Figure 5: Applying CSF to Mediation.

are responsible for the semantic and syntactic mapping and two representative domains within the Data Layer that provide a wrapper for the two data sources.

As a client interacts with the mediation system, the client's application within the Application Layer utilizes the functionalities provided by the CSF to initialize a root entity to protect the data that the client's application enters into the mediation system. During the initialization the client's application specifies the communication TCP/IP port and the security policies of the root entity. To establish the security polices the root entity requests the needed security rules from the CSF's Security Repository. Thereafter, the client's application and root entity are in constant communication.

To advance the client's request the root entity needs to locate other representative agents within different representative domains that can advance the underlying request. The root entity can find these representative agents by querying the CSF's Directory Service. During this task the root entity also retrieves the security requirements that are needed to establish collaboration channels with these representative agents. Having the needed security requirements, the root entity makes requests to the advancing representative agents to establish collaboration channels with the root entity's own representative agents. Note, during these requests there are possible negotiations on the security policies that govern the collaboration channels. If a negotiation fails, there will be no further communication between this representative agent and the root entity. In this case, the root entity may request another representative agent from the CSF's Directory Ser-

vice. If there is an agreement between the root entity and the advancing representative agent, the root entity will initialize one of its own representative agents with the global, local, and collaboration security rule, and the data that is needed to advance the root's entity request i.e. the client's request.

Upon establishing a collaboration channel the root entity's representative agent and the advancing representative agent are collaborating with the possibility of having their security policies based on different security models i.e. one representative domain security polices based on RBAC (Sandhu et al., 1996) and the other on BPL (Bell and Padula, 1975).

As the mediation system provides semantic and syntactic interoperability to simplify the client's request, the client's representative agent will create a nested web of collaboration channels to other representative agents until the request is simplified to a point where it can be processed by a data source. This is illustrated in Figure 5.

As shown in Figure 5, at each point of a cross-domain collaboration the client's data is protected by a representative agent making it possible for the client to control the access to his or her data. The client may request the root entity to change the global and local security polices of some or all representative agents during runtime by sending the request through the established collaboration channels between these entities. Note, all entities within the mediation system i.e. clients application, mediations or data source utilize the CSF, making it possible to simplify the mediation system security architecture. Therefore, CSF can provide each entity within the mediation system, access control over its data and security policies at a local and global view that are security model independent and can be changed at runtime.

7 RELATED WORK

The problem of semantic and syntactic interoperability of heterogeneous data sources have been addressed by TIHI(Wiederhold et al., 1996), CHAOS (Liu et al., 2000), and (Yang et al., 2004). These mediation systems each utilize a different security mechanism to establish secure collaboration between different entities.

TIHI security mechanism is comprised of a security mediator, which is an intelligent gateway that applies pre-processing and post-processing filter of data requested by a client. The security mediator filters incoming and out-going data from TIHI's data sources. There can be situations where the security mediator cannot handle all data filtering. In these situations, TIHI's security officer processes the data. Note, the security mediator is under the control of the security officer and therefore, the security officer may override any data filtering that was applied by the security mediator. Through the use of security rules, which the security officer loads into the security mediator, TIHI establishes its security throughout the entire mediation system.

In CHAOS, security of the mediation system is based on active objects that are dynamically loaded into mediators. These active objects filter the data that is being presented to the client. CHAOS does not need to rely on a set of primitive security rules. To establish security policies CHAOS provides an API to set the security polices with an active object. Before the data within an active object is sent to a client, the active object's execute method is called by the mediator containing the active object. Calling the execute method on an active object causes the internal data of the active object to be filtered, therefore the client will review a subset of the active object's data that depends on the access rights of the client.

(Yang et al., 2004) based their security mechanism on RBAC (Sandhu et al., 1996) by utilizing the mediation system's Access Control and View Expander components and the mediation spec database. Their mediation system applies security rules that filters data being sent to the client. The Access Control component extracts the policy information to enforce the authorization constraints on the client. To access the proper views of the data source that is in compliance with client's access rights, the mediation system utilizes the View Expander that relies on the mediation spec database to set the proper view access.

These security mechanisms approaches have their limitations in the situation where a client data needs to be protected during processing. This is due to the fact that these security mechanisms do not provide any services that make it possible for a client to control the access to its data during the mediation process.

8 CONCLUSION

In this paper, we have presented the Collaborative Security Framework (CSF), which allows all entities participating in a cross-domain collaboration to control the access to their data by applying local, global and collaboration channel security rules, which can be changed at runtime. These security rules have different scopes, where local security rules are applied to protect the internal data of a representative agent, global security rules make up a representative domain security policy that every entity within a representative domain must adhere to and collaboration channel security rules that govern the collaboration between entities.

The primary abstraction of the CSF is the represen-

tative domain, which represents a client during collaboration. This is made possible through the use of root entities and representative agents. The root entity is the controller of a representative domain and is the only entity within a representative domain that communicates directly with the client. Whereas, representative agents represent the root entity during a cross-domain collaboration, this makes it possible to have decentralize collaborations, where only necessary data needs to be given to a representative agent by its root entity.

Furthermore, the CSF provides an infrastructure so that collaboration across different domains are security model independent, since each representative domain can base its security rules on different security models. This was a key design feature of the CSF, since this reflects more real world situations.

Due to space limitations, we do not provide a proof of completeness, verification mechanism, protocols or security polices reconciliation of the CSF. These topics will be presented in our future work.

REFERENCES

Barker, P. (1995). An analysis of user input to an x.500 white pages directory service. *IEEE/ACM Trans. Netw.*, 3(2):112–125.

Bell, D. and Padula, L. L. (1975). Secure computer systems: Unified exposition and multics interpretation. Technical Report ESD-TR-75-306, MITRE MTR-2997.

Bhatti, R., Ghafoor, A., Bertino, E., and Joshi, J. B. D. (2005). X-gtrbac: an xml-based policy specification framework and architecture for enterprise-wide access control. *ACM Trans. Inf. Syst. Secur.*, 8(2):187–227.

Bistarelli, S. (2004). *Semirings for Soft Constraint Solving and Programming*, volume 2962 of *Lecture Notes in Computer Science*. Springer.

Bistarelli, S., Montanari, U., and Rossi, F. (1997). Semiring-based constraint satisfaction and optimization. *J. ACM*, 44(2):201–236.

Bradshaw, J. M., Dutfield, S., Carpenter, B., Jeffers, R., and Robinson, T. (1995). KAoS: A Generic Agent Architecture for Aerospace Applications. In Finin, T. and Mayfield, J., editors, *Proceedings of the CIKM '95 Workshop on Intelligent Information Agents*, Baltimore, Maryland.

Brewer, D. F. C. and Nash, M. J. (1989). The chinese wall security policy. In *IEEE Symposium on Security and Privacy*, pages 206–214.

Dawson, S., Samarati, P., di Vimercati, S. D. C., Lincoln, P., Wiederhold, G., Bilello, M., and Akella, J. (2000). Secure access wrapper: Mediating security between heterogeneous databases. In *Proc. of the Darpa Information Survivability Conference & Exposition*, Hilton Head, South Carolina.

Ege, R. K., Yang, L., Kharma, Q., and Ni, X. (2004). Three-layered mediator architecture based on dht. In *ISPAN*, pages 313–318.

Gong, L. and Qian, X. (1994). The complexity and composability of secure interoperation. pages 190–200.

Gong, L. and Qian, X. (1996). Computational issues in secure interoperation. *Software Engineering*, 22(1):43–52.

Harrison, M. A., Ruzzo, W. L., and Ullman, J. D. (1976). Protection in operating systems. *Commun. ACM*, 19(8):461–471.

Lange, D. B. and Oshima, M. (1999). Seven good reasons for mobile agents. *Communications of the ACM*, 42(3):88–89.

Liu, D., Law, K., and Wiederhold, G. (2000). Chaos: An active security mediation system. In *Conference on Advanced Information Systems Engineering*, pages 232–246.

Park, J. and Ram, S. (2004). Information systems interoperability: What lies beneath? *ACM Trans. Inf. Syst.*, 22(4):595–632.

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

Shehab, M., Bertino, E., and Ghafoor, A. (2005). Secure collaboration in mediator-free environments. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*, pages 58–67, New York, NY, USA. ACM Press.

Sheth, A. P. and Larson, J. A. (1990). Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236.

Steiner, J. G., Neuman, B. C., and Schiller, J. I. (1988). Kerberos: An authentication service for open network systems. In *Proceedings of the USENIX Winter 1988 Technical Conference*, pages 191–202, Berkeley, CA. USENIX Association.

Thome, M., Helsinger, A., and Wright, T. (2004). Cougaar: a scalable, distributed multi-agent architecture. In *SMC (2)*, pages 1910–1917.

Wallace, M. (1996). Practical applications of constraint programming. *Constraints*, 1(1/2):139–168.

Wiederhold, G. (1992). Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38–49.

Wiederhold, G., Bilello, M., Sarathy, V., and Qian, X. (1996). A security mediator for health care information.

Wiederhold, G. and Genesereth, M. R. (1997). The conceptual basis for mediation services. *IEEE Expert*, 12(5):38–47.

Yang, L., Ege, R. K., Ezenwoye, O., and Kharma, Q. (2004). A role-based access control model for information mediation. In *IRI*, pages 277–282.