Secure Collaboration in Mediator-Free Environments

Mohamed Shehab
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
shehab@purdue.edu

Elisa Bertino
Department of Computer Sciences and CERIAS
Purdue University
West Lafayette, IN, USA
bertino@cs.purdue.edu

Arif Ghafoor
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN, USA
ghafoor@purdue.edu

ABSTRACT

The internet and related technologies have made multidomain collaborations a reality. Collaboration enables domains to effectively share resources; however it introduces several security and privacy challenges. Managing security in the absence of a central mediator is even more challenging. In this paper, we propose a distributed secure interoperability framework for mediator-free collaboration environments. We introduce the idea of secure access paths which enables domains to make localized access control decisions without having global view of the collaboration. We also present a path authentication technique for proving path authenticity. Furthermore, we present both a proactive and on-demand path discovery algorithms that enable domains to securely discover paths in the collaboration environment.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Access controls; H.2.7 [**Database Administration**]: Security, integrity, and protection.

General Terms

Design, Security, Theory.

Keywords

Decentralized Secure Interoperability, Collaboration, Access Path, Path Discovery, Role Based Access Control.

1. INTRODUCTION

The phenomenal growth of the Internet has enabled a globalization that has removed barriers between markets, organizations and societies. The Internet has become integrated into practices of individuals, business, and governments. In such a connected world, there are immense possibilities of collaboration in distributed environments. For example, interoperability has enabled companies to outsource their operations overseas to reduce production and employment costs. Furthermore, interoperability adds to the efficiency of companies by leveraging the use of existing resources other than reinventing the wheel. Even more interestingly, by migrating processes across organizational boundaries companies are able to combine their efforts and become virtual enterprises [1, 17]. Last but not least interoperability is essential to support adaptation and evolution in complex enterprises [8]. Such enterprises [21] are organized according to units with varying degrees of coupling and autonomous coordination linking these units. As such an enterprise evolves to meet new demands, new interoperation links across units may need to be established and existing links removed.

Though interoperability has several advantages and is crucial in the context of new dynamic collaborative applications and adaptive enterprises, it introduces several security and privacy concerns. These concerns have to be addressed to make such interoperability techniques a viable tool in multidomain contexts. In particular, a domain represents a core element in a collaborating environment. A domain is a separate, autonomous entity that manages a group of resources, and has its own administration and access control policies. Because very often domains need to collaborate to share resources, a key step in setting up such a collaboration among domains is represented by the interoperation of access control policies. Domains typically achieve interoperation among their access control policies by introducing cross mappings between these policies. An important requirement is that such interoperation of policies be secure; Gong and Qian [12], among others, have shown that if interoperation between access control policies is not carefully established, security breaches may arise.

Secure interoperability in a multidomain environment is a challenging task even in the presence of a trusted mediator managing security of such collaboration [12, 4, 7]. It is much harder to handle security in a fully distributed and dynamic interoperation environment where domains join and leave in an adhoc manner and in the absence of a trusted mediator. However, we believe that the development of fully decentralized solutions tailored to dynamic environments is crucial to meet the security requirements of next-generation enterprises.

In this paper we develop such a solution. We propose a distributed framework addressing both the security and autonomy requirements of domains in a mediator-free interoperation environment. In our framework the user's access history migrates with the user's access requests to enable domains to make localized access control decisions without

CCS'05, November 7–11, 2005, Alexandria, Virginia, USA.

needing to have a global view of the collaboration environment. We define a set of basic and extended path linking rules that enable domains to make access control decisions. We also provide a path authentication technique that ensures the authenticity of the user's access path as it propagates between domains. Our framework provides both reactive and on-demand path discovery algorithms that enable users to discover available secure access paths in the interoperation environment.

1.1 Contributions and Paper Organization

The contributions in this paper can be summarized as follows:

- We present a mediator-free collaboration environment and discuss the security challenges in such an environment. We define access paths and present access path security requirements in a secure collaboration.
- We provide a framework for enabling secure collaboration in a mediator-free environment, in which access control decisions are dependent on the user's access history in the collaboration environment.
- We discuss several security attacks that can be performed in a mediator-free environment, and provide mitigation techniques to such attacks.

The rest of the paper is organized as follows. In Section 2 we review the requirements of secure interoperability, the maximal secure interoperability and the drawbacks of the maximal secure interoperability solution proposed by Li and Qian [12]. We introduce the mediator-free collaboration environment in Section 3. In Section 4 we present our framework for secure collaboration in a mediator-free environment and define the notion of secure access path. The request execution strategy and path authentication module are discussed in Sections 5 and 6, respectively. The proactive and on-demand path discovery algorithms are presented on Section 7. Possible security attacks and mitigation techniques are discussed in Section 8. The related work is presented in Section 9. Concluding remarks are added in Section 10.

2. PRELIMINARIES

In our framework, we assume that all the domains adopt a role-based access control (RBAC) model [10, 9] to model their access control policies. The analysis performed in this paper and the framework we have developed can still be applied when other access control models are adopted. We have chosen RBAC because it is suitable for specifying the security requirements for a wide range of commercial, medical, government applications [23, 3] and moreover it is being standardized. A domain that does not use RBAC as its access control model can easily generate an export RBAC policy to join the collaboration.

In RBAC, permissions are associated with roles, and users are granted membership in appropriate roles, thereby acquiring the roles' permissions. The access control policy PO_i for domain i is modeled as a directed graph $G_i = \langle V_i, A_i \rangle$ where the vertex set V_i represents roles and the arcs set A_i represents the dominance relationship between roles. For example, if role r_1 dominates r_2, ($r_2 \preceq r_1$), then $(r_1, r_2) \in A_i$. Thus a user acquiring role r_1 can acquire permissions assigned to role r_2 by using the RBAC permission inheritance properties [6]. For $r_x, r_y \in V_i$ an access (r_x, r_y) is legal if and only if $(r_x, r_y) \in G_i^+$ where G_i^+ is the transitive closure of $G_i = \langle V_i, A_i \rangle$. We denote a legal access by $(r_x, r_y) \propto A_i$.

2.1 Secure Interoperability

In a collaboration involving n domains, in which the access control policy of each domain i is modeled as a directed graph $G_i = \langle V_i, A_i \rangle$, $i = 1, \ldots, n$, interoperability is achieved by introducing cross domain pairwise mappings between the n domains. These mappings relate roles in different domains, and are represented by a set of cross domain arcs referred to as the set F. Solutions developed for schema matching in the area of heterogeneous database systems and more recently approaches based on ontologies [19, 18] can be used for generating such links. The details of such approaches are outside of the scope of this paper. In the present work we assume that the cross domain mappings are selected by the administrators of the domains according to the interoperability requirements of each system. Furthermore, the system administrators agree on a set of restricted accesses which is similar to negative authorizations adopted in several access control models. The restricted access is a binary relation R on $\bigcup_{i=1}^{n} V_i$ such that $\forall (u, v) \in R, u \in V_i, v \in V_j$, and $i \neq j$, where these edges in R are prohibited to exist during interoperation.

Given n domains $G_i = \langle V_i, A_i \rangle$, $i = 1, \ldots, n$, set of cross links F and a restricted access relation R, an interoperation $Q = \langle \cup_{i=1}^{n} V_i, A_Q \rangle$, where A_Q is the resulting arc set $A_Q \subseteq \{\cup_{i=1}^{n} A_i \cup F\}$, is secure according to Li Gong et al. [12] if it satisfies all the following conditions:

1. $A_Q \cap R = \emptyset$.
2. $\forall u, v \in V_i, (u, v)$ is legal in A_i if and only if (u, v) is legal in A_Q.

The first requirement ensures that restricted access relation is honored. The second requirement ensures the following two properties hold:

- **Autonomy:** It requires that any access permitted with in an individual domain must also be permitted under secure interoperation.
- **Security:** It requires that any access denied within an individual domain must also be denied under secure interoperation.

2.2 The Maximum Secure Interoperation (MSI)

Definition 1. Maximum Secure Interoperability (MSI) Given n domains $G_i = \langle V_i, A_i \rangle$, $i = 1, \ldots, n$, a set of cross links F and a restricted access relation R, for any positive integer $K \leq |F|$, determine whether a secure solution $Q = \langle \cup_{i=1}^{n} V_i, A_Q \rangle$ exists such that $A_Q = \{\cup_{i=1}^{n} A_i \cup S\}$ where $S \subseteq F$ and $|S| \geq K$.

Simply, the MSI solution finds a maximal subset of the cross links set F such that the secure interoperability is ensured. The MSI solution inherently satisfies the autonomy requirement as $A_Q = \{\cup_{i=1}^{n} A_i \cup S\}$. Taking a closer look at the MSI solution we conclude it has the following drawbacks:

- **NP-Completeness:** Li Gong et al. [12] showed a polynomial reduction of the Feedback Arc Set problem, which is a known NP-complete problem, to the

MSI problem, thus proving that MSI is an NP-complete problem. Thus it is not practical to solve the MSI problem for a large number of collaborating domains. Moreover, any practical solution to this problem would be based on heuristics and in such cases the generated solutions are approximate and are not guaranteed to be optimal.

- **Centralized Algorithm:** The MSI problem assumes full knowledge all domains' access control policies $G_i = \langle V_i, A_i \rangle, i = 1, \ldots, n$, and the sets F and R. To solve the MSI problem a global view of the system is required. A trusted mediator having the global view computes the subset of F that satisfies the constraints of MSI. The mediator represents a bottleneck and therefore such solution is not scalable in distributed environments with a large number of interacting parties.
- **Static Solution:** The MSI solution computed with n collaborating domains is optimal and secure for these n domains; if however a domain decides to leave or join the collaboration, the MSI solution has to be recomputed to ensure both optimality and security. Furthermore, the MSI solution should be recomputed if a domain edits or updates its security policy. This is not practical in dynamic environments in which domains are required to join and leave the interoperation environment transparently without the need for delays and revocations of current coalitions.
- **Fairness Issue:** The MSI solution resolves violations by removing cross links from F. However, in a violation several domains are involved and the removal of cross links will affect a subset of these domains. The following example elaborates on the fairness issue. Consider Figure 1 where domains A, B and C are collaborating. Each domain has an access control hierarchy represented as a graph. The cross links are represented by the dotted lines. A user in domain A acquiring role r_{A_1} could access role r_{A_3} by accessing roles $\{r_{B_3}, r_{B_1}, r_{C_2}, r_{C_1}, r_{A_3}\}$ which is clearly a security violation as $r_{A_1} \preceq r_{A_3}$. Furthermore, using a similar argument a user at r_{B_1} and r_{C_1} could access roles r_{B_3} and r_{C_2} respectively. The MSI solution would remove one or more cross links to break such cycle. Assume that the MSI solution removes edge (r_{C_1}, r_{A_3}); this solution eliminates the security violation but users in domain C are unable to access roles in domain A. This solution is not fair as it restricts access by users of domain C, whereas rights of users in other domains are not affected.

From the above discussion we conclude that the MSI solution is NP-Complete, requires a trusted mediator, is static, and moreover it is not fair to all the participating domains. In the next sections we propose a secure technique which represents a computationally simple, distributed, dynamic solution, and ensures fairness to the participating domains.

3. MEDIATOR-FREE SECURE COLLABORATION

In this section we present the key notion of our framework, that is, the notion of mediator-free secure collaboration environment, which does not require a mediator having

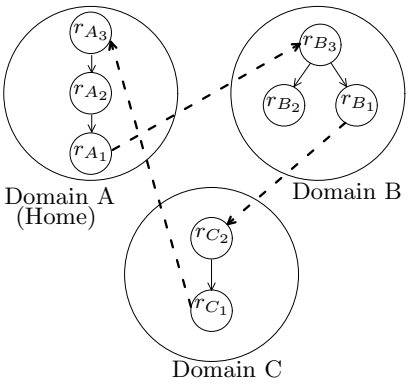


Figure 1: Example of a violation in a multidomain environment with three domains. The solid lines show the internal access links, while the dotted lines show the interoperation cross links F.

a global view in order to ensure secure interoperability. Figure 2 shows both types of collaboration environments. To design a mediator-free environment we need to analyze the functions performed by the mediator, which include:

- **MSI computation:** From the global view the mediator computes the MSI solution, which generates the optimal set of collaboration cross links between the domains.
- **Role Querying and Routing:** By using the global view of the collaboration environment the mediator is able to answer queries of the form "is r_2 reachable from r_1?" where r_1 and r_2 are in different domains. Furthermore, the mediator can easily determine paths between reachable roles in different domains.

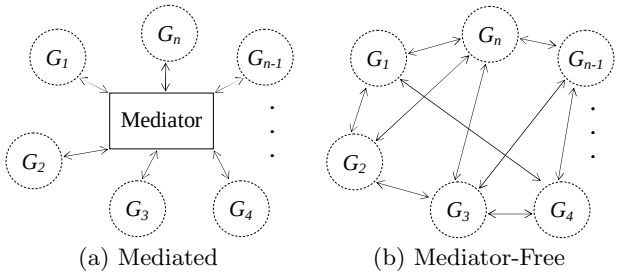


Figure 2: Collaboration environment with and without a mediator.

A mediator-free collaboration is a completely distributed form of collaboration. In this environment the domains have to collaborate in making access control decisions to avoid violations. In a mediator-free environment none of the collaborating domains has the global view of all the access control policies; instead the domains view the collaboration environment only through their established cross links. Enforcing secure interoperability in such an environment is a challenging task as it requires domains to collaborate in both sharing of resources and making access control decisions. In a mediator-free secure collaboration the mediator functions should be executed across the collaborating domains according to a distributed strategy.

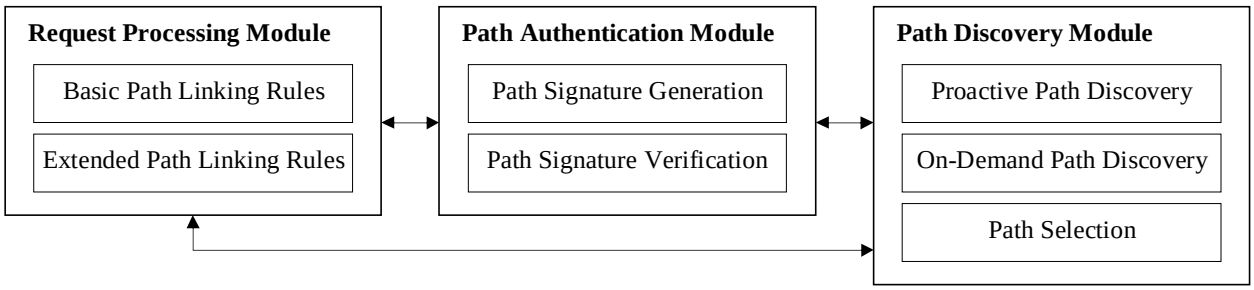


Figure 3: Modules of the mediator-free secure interoperability framework.

The following assumptions apply to a mediator-free environment:

1. Each domain's first priority is to ensure that its security policy is not violated.
2. Domains have limited information about the collaboration environment. Each domain is only required to know its security policy, the cross links and restricted links in which it is involved.

4. FRAMEWORK FOR SECURE MEDIATOR-FREE COLLABORATION

In this section, we present our framework for enabling secure collaboration in a mediator-free environment. Our framework provides a secure interoperability solution that prevents security violations as access requests are being made. Our solution requires no complicated preprocessing, and allows the complete set of cross links to exist. Furthermore, it enables domains to make localized access control decisions without the need for the global view of the collaboration environment. These characteristics make our solution very suitable for the enforcement of access control in a mediator-free environment.

Our framework utilizes the user's current access history during the collaboration session to dynamically grant or deny future access requests. We refer to the user's access history as the *user's access path*, which is the sequence of roles acquired by the user during the current session. Our proposed solution shares the ideas of the Chinese Wall security policy [5], as the user's access history controls his future accesses. The basis of the Chinese Wall policy is that users are only allowed access to information which is do not conflict with any other information that they have already accessed. In this context, the user's access path represents the user's session history and the user's view of the possible future paths is dependent on his current access path.

4.1 Framework Overview

Our framework enables domains to make localized access control decisions based on the user's access history in the collaboration environment. It is composed by the following major modules (see Figure 3):

M1. Request Processing Module: This module generates and evaluates user access requests across domains.

M2. Path Authentication Module: Because the user path migrates with the user requests, this module checks the authenticity of the received paths. Also, this module generates path signatures for generated requests.

M3. Path Discovery Module: This module enables users residing in a home domain to determine which roles are accessible in target domains.

The above three modules are included in each domain. The modules interact with each other to ensure the security of their corresponding domains. The detail of each module of our framework is discussed in detail in further sections. As access paths constitute an important dimension in our framework, we define in what follows access paths and secure access path requirements.

4.2 Access Paths in an Interoperation Environment

In a user session we identify three main types of domains, namely *home, current* and *target* domains. The home domain is the domain at which the user session starts. The current domain is the domain from which the user generates access requests. The target domain is the domain to which the user is requesting access to. When a user enters a domain the user is assigned an *entry role*. Similarly, when the user leaves a domain to access another domain the user is assigned an *exit role*. Note that the entry and exit roles may coincide. Figure 4 shows the home, current and target domains. The entry and exit roles are referred to as r^E and r^X respectively, where the user's access path in Figure 4 is $P = \{r_H^E, r_H^X, \ldots, r_C^E\}$.

Definition 2. The user's access path is defined as the sequence of entry and exit roles acquired by a user during a given session from the home domain to the current domain.

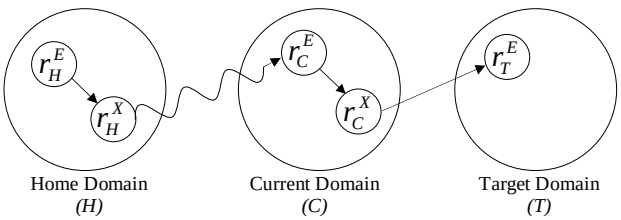


Figure 4: Types of domains, entry and exit roles.

The secure interoperability requirements presented by Li Gong et al. [11] which were mentioned in Section 2.1 ensure that all the possible paths in secure interoperation do not

violate the access control policy of each domain and that both the cross links set F and the restricted access set R are honored. Here we present the notion of secure access path.

Definition 3. Let $P = \{r_1, r_2, \ldots, r_n\}$ be an access path, where $i < j$ implies that role r_i was acquired before r_j. Moreover, let $Domain(r_i)$ denotes the domain of role r_i. P is secure if it satisfies the following conditions:

C1. For all $i < j$ and $r_i, r_j \in P$, if $Domain(r_i) = Domain(r_j)$ then $r_j \preceq r_i$.

C2. For all $r_i, r_{i+1} \in P$, if $Domain(r_i) \neq Domain(r_{i+1})$ then $(r_i, r_{i+1}) \in F$.

C3. For all $i < j$ and $r_i, r_j \in P$, $(r_i, r_j) \notin R$.

Condition C1 ensures that roles acquired from the same domain are acquired according to the domain's role hierarchy. This ensures that the access control policies of the domains included in the path are not violated. Conditions C2 and C3 ensure that sets F and R are honored. The user's access path is attached to user requests to enable domains to make localized access control decisions. This is analogous to source routing techniques for limited bandwidth wireless sensor networks [14, 13] in which the route from source to destination is attached to the packet to enable routing of the packet. Thus, including path information is an acceptable assumption; in the following sections we present techniques to limit the size of the access path.

5. REQUEST PROCESSING MODULE

In a mediator-free collaboration environment each domain has a limited view of the collaboration environment meaning that each domain has a complete knowledge of its own access control policy, that is, the cross links and the restricted access links in which it is involved. Let $F_T \subseteq F$ and $R_T \subseteq R$ be the cross links and restricted access links that the target domain T is involved in. An access request from another domain includes the requested role, and the user's current access path. Given this limited information the target domain can decide whether to reject or accept the access request. In such an environment each domain is mainly concerned with ensuring that its access control policy is not violated. By verifying the following basic path linking rules a target domain is able to securely grant or revoke a request.

Definition 4. **The basic path linking rules:** Let P be a secure path, r_C^X the exit role in the current domain, r_T^E the requested role in the target domain. The target domain must verify the following conditions in order to grant access to the requested role:

L1. $(r_C^X, r_T^E) \in F_T$

L2. For all $r \in P$, $(r, r_T^E) \notin R_T$

L3. For all $r \in P$, if $Domain(r) = T$ then $r_T^E \preceq r$

The next theorem proves that the basic path linking rules assure the security of the computed path if all the conditions $L1 - 3$ are verified before a link is added to the path. Note that $\circ$ is the concatenation operator.

THEOREM 1. *Let P_i be a secure path, and $P_{i+1} = P_i \circ r_T^E$ be an updated path that satisfies the basic path linking rules. Then P_{i+1} is also a secure path.*

***Proof**, The initial path $P_i = (r_1, r_2, \ldots, r_n)$ is secure, where $r_1 = r_H^E, r_2 = r_H^X$ and $r_n = r_C^X$ refer to Figure 4. We proceed using a proof by contradiction. Assume to the contrary that the new path P_{i+1} is not secure after satisfying all the basic path linking rules. If this is the case, then a violation exists in path $P_{i+1} = P_i \circ r_T^E$. This violation can be due to P_i or (r_C^X, r_T^E) or (r_k, r_T^E), where $r_k \in P_i, 1 \leq k \leq n$. Since P_i is the initial path and it is assumed to be secure then it cannot contain a violation. Rule L1 checks that $(r_C^X, r_T^E) \in F_T$ and Rule L2 ensures that $(r_C^X, r_T^E) \notin R_T$ thus this link cannot be the cause of the violation. We are now left with only links (r_k, r_T^E); however rule L2 ensures that such links are not in the R_T and rule L3 checks the integrity of adding such links and insures that the ordering among the roles in the domain's internal roles hierarchy is not violated; thus these links cannot result in security violations. In this case as all the possible links that could lead to a violation have been proven to be secure after verifying the basic path linking rules which contradicts our assumption and thus path P_{i+1} can only be a secure path.* □

Note that the basic linking rules applied by the target domain are based on the target domain access policy, the reduced cross link and restricted sets F_T and R_T, and the user's access path. Thus, the target domain is able to make secure access control decisions without a global view of the collaboration environment. All the computations performed to execute rules L1-3 are computationally simple operations and can be computed in polynomial time. Furthermore, the basic linking rules do not remove any cross links and thus the solution is fair to all the domains in the collaboration environment.

5.1 Extended Path Linking Rules

In addition to the basic path linking rules, the extended rules provide more constraints on the user's access path. Such constraints are useful for securing many applications with special path requirements. The restricted access relation R is only capable of representing simple binary mutual exclusion constraints of the form (r_1, r_2) stating that roles r_1 and r_2 must not be accessed by the same user in the same session. Other path restrictions are desirable for certain applications. Cardinality and SoD constraints are crucial for securing many applications in a commercial environment. Many researchers have highlighted the importance and use of cardinality and SoD constraints in RBAC models [10, 9, 16]. However, no one has addressed the these constraints in the context of a multidomain collaborative environment.

A more general type of such constraints requires that no user be a member of t or more roles in a set of m roles $\{r_1, r_2 \ldots, r_m\}$ in a given session [16]. Assuming the user's access path is P, then this type of constraint can easily be checked by verifying that $|P \cap \{r_1, r_2, \ldots, r_m\}| \leq t$, where $|x|$ denotes the cardinality of the set x.

Cardinality constraints are constraints on the size of the access path. A cardinality constraint of the form $|P| \leq Pmax$ bounds the number of roles acquired in a session to a number $Pmax$ of roles.

Ordering constraints enforce conditions on the order according to which the roles have to be acquired. Such con-

straints are relevant in the context of workflow systems [3], in which certain roles should be acquired before others roles can be activated.

6. PATH AUTHENTICATION

The access path is attached to the user's requests as it migrates across domains. A technique is required to ensure that this path is authentic and has not been tampered with. The authentication scheme proposed is based on a signature that is generated by all the domains included in the access path. The authentication scheme should preserve both the path contents and the ordering. Each domain i has a private key e_i and a public key d_i.

The path signature is computed as the user request is sent from the current domain to the target domain. For a user currently in domain i and requesting access to a target domain $i+1$ the current path is $P_i = \{r_1^E, r_1^X, \ldots, r_i^E, r_i^X\}$, where r_k^E and r_k^X, $k = 1, \ldots, i$, are the entry and exit roles in domain k respectively. The signature $S.P_i$ of path P_i is computed as follows:

$$S.P_i = \begin{cases} SIGN_{e_i}(S.P_{i-1} \oplus h(r_i^E \circ r_i^X \circ i+1)) & \text{if } i \geq 1 \\ seed & \text{if } i = 0 \end{cases}$$

where $\circ$ is the concatenation operator, $\bigoplus$ is the XOR operator, *seed* is a random number generated by the home domain which is included in the path information, $h()$ is a secure one-way hash function, and $SIGN_K(M)$ is a signature function that uses key K to sign message M. The signature is generated using modular exponentiation similar to techniques used in RSA signatures [22]. Domain i already has the signature $S.P_{i-1}$ of path P_{i-1}, thus domain i can easily compute $S.P_i$ as r_i^E, r_i^X, $i+1$ and e_i are known by domain i. The path signature $S.P_i$ is signed using the private key e_i thus this signature cannot be forged. The signature function has the following property:

$$SIGN_{d_i}(SIGN_{e_i}(M)) = M$$

Presented with P_i, $S.P_i$ and the *seed* the target domain $i+1$ can easily verify the path signature by performing the following operation:

$$SIGN_{d_i}(S.P_i) \oplus h(r_i^E \circ r_i^X \circ i+1) = S.P_{i-1}, \text{ for } i \geq 1$$

The target domain can easily check the authenticity of a path P_i by recursively computing the above equation and comparing $S.P_0$ with the *seed*. Note that the signature verification is performed using the public key information of the involved domains; thus the verification does not require contacting the involved domains.

7. PATH DISCOVERY

Cross links are the main enablers of collaboration. Domains are able to collaborate with neighboring domains through the established collaboration cross links. Neighboring domains are single hop collaborations as they only involve two domains. Single hop collaborations are easy to achieve and initiate as domains already have full knowledge of their established cross links. One the other hand, in order to collaborate through multi-hop collaborations domains need to build one or more candidate access paths to target domains. To enable domains to discover available multi-hop collaborations a distributed path discovery algorithm is required. The discovery algorithm enables domains in an interoperation environment to discover paths to roles in other domains, whether reachable through one or more intermediate domains. Furthermore, the discovered paths should follow the path linking rules to ensure the discovered path(s) security. In this section, we present two path discovery algorithms, *proactive* and *on-demand* path discovery algorithms.

7.1 Proactive Path Discovery

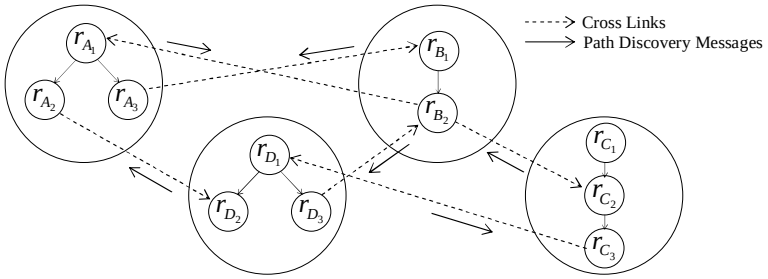
The proactive path discovery algorithm computes the paths from the roles in current domain to roles in other domains a *priori*. Each domain generates and maintains a *role routing table*, which is a partial map of the collaboration environment representing the view with respect to the current domain. Neighboring domains exchange periodic discovery updates via cross links indicating reachable domains through these links. Note that cross link related to domain i can be divided into outgoing and incoming cross links, referred to by F_i^O and F_i^I respectively, where $F_i^O \cup F_i^I = F_i$. The periodic messages are sent by domains on their *incoming* cross links. Figure 5(a) shows the direction of the periodic discovery updates in an example collaboration environment.

The content of the periodic message is chosen by the advertising domain to indicate paths to roles accessible through the cross link over which the message is sent. To avoid loops, paths that include the domain to which the update message is to be sent are dropped or truncated. For example in Figure 5(a) the update message sent from domain B to domain A across (r_{A_3}, r_{B_1}) will report the roles reachable via r_{B_1} in all domains other than domain A, this clearly avoids loops. To avoid infinitely growing paths, a cardinality constraint should be set on the path length; the path length could be limited to double the number of estimated collaborating domains. The proactive path discovery algorithm is similar to link state routing; however, there are several differences. For example, in a collaboration environment cross links are not necessarily bidirectional and routing metrics are not necessarily based on distance instead on higher level logic dictated by the cross links and the domain hierarchies.

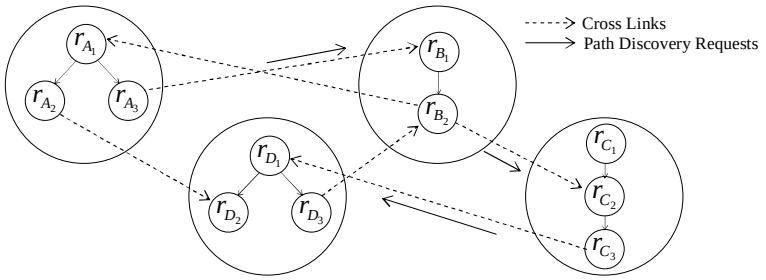
To ensure the authenticity of the reported paths, a path signature is computed based on technique similar to the path authentication scheme discussed in Section 6. However, the path signature is computed in the reverse direction, from the target domain to the home domain, as the path is discovered from target to home. The advantage of a proactive path discovery is that when a domain needs to collaborate with a target domain, the path is already available and thus there is no latency. Furthermore, this technique is reactive to collaboration environment changes such as changes in the cross links, domain policies, and the entry or exit of collaborating domains. The disadvantage is that some paths may never be used during the collaboration period. Another problem is that the dissemination of path information will periodically consume network bandwidth.

7.2 On-Demand Path Discovery

The on-demand path discovery algorithm computes paths from the roles in the current domain to roles in a target domain only when such path is needed. Neighboring domains do not exchange periodic path message updates; instead simple "Hello" messages are sent between domains that share cross links to announce that the link is still alive. When a home domain needs to establish a path to a certain role in a



(a) Proactive Path Discovery



(b) On-demand Path Discovery

Figure 5: Examples of proactive and on-demand path discovery

target domain, a path request message is generated by the home domain and is sent on its outgoing cross links. Upon receiving the path request message, the receiving domain performs a path evaluation based on the basic path linking rules. If the path is accepted, the domain updates the path and resends the request through its cross link excluding the cross links that involve domains already included in the path. This ensures that the path is loop free and reduces the number of resent requests. Figure 5(b) shows an example of a on-demand path discovery initiated by domain A to determine roles reachable at domain D from role r_{A_3}. The solid arrows show the path discovery messages; note that domains B does not forward the request on the cross link (r_{B_2}, r_{A_1}) as domain A is in the computed path. Also, note that domain D does not forward the request on the cross link (r_{D_3}, r_{B_2}) as domain B is in the computed path. Figure 6 shows the algorithm executed when a domain j receives a path request from a neighboring domain.

After sending the path request the home domain waits for a timeout period of T_{max}. If no reply arrives from the target domain then this means there are no secure paths from home domain to the target domain. The value of T_{max} is assigned based on the number of collaborating domains. The path authenticity is ensured by using the path signature scheme discussed in Section 6 as the path request message takes the path taken by the actual access request.

The major advantage of on-demand path discovery is that it saves network bandwidth because it limits the amount of bandwidth consumed in the exchange of path discovery information by maintaining paths to only those target domains to which the domains need to collaborate with. The home domain could include constraints on the requested path, to further reduce the path discovery traffic. For example, the request could include a list of domains that should or should not be included in the path discovery. On-demand path discovery also obviates the need for disseminating path discovery information periodically, or flooding such information whenever a cross link changes or when a domain leaves or joins the collaboration environment. The primary problem with on-demand path discovery is the large latency at the beginning of the collaboration caused by propagation of the path request message.

7.3 Path Selection

Both path discovery algorithms could return multiple secure paths between the home and target domains. The home domain, selects one path according to a selection criteria. The selection criteria is based on the path properties which include:

- **Path length:** The path having the shortest length in terms of the number of visited domains is selected.
- **Visited domains:** Select the path that contains a certain set of domains or visits domains according to a certain sequence.
- **Composite domain reputation:** Domains could be given reputation metrics and the path reputation is computed using the domains included in the path, and the path having the highest reputation be selected.

8. SECURITY ANALYSIS

In this section we discuss some security attacks that could be performed in a mediator-free collaboration environment. Moreover, we show that our secure framework is resilient to these attacks.

- **Path Corruption.** The access path is one of the main elements required when making access control decisions. A malicious domain may attempt to alter the access path by removing or adding entries to the

Input: Request = (Target Domain T, Path P, Path Signature $S.P$) arriving to domain j from domain i via cross link (r_i^X, r_j^E).
Algorithm:

1. Check Path signature $S.P$, if invalid drop request, End.
2. Check the Path linking rules on P and link (r_i^X, r_j^E), if insecure path then drop request, End.
3. If $|P| > Pmax$ then drop request, End.
4. If domain j is equal to the target domain T
   (a) Update path $P_{new} = P \circ r_j^E$.
   (b) Generate path signature $S.P_{new} = SIGN_{e_j}(S.P \oplus h(r_j^E))$.
   (c) Send Path Request Reply =(P_{new}, $S.P_{new}$) to Home domain in P
5. For all cross links $L = (r_j^X, r_k^E) \in F_j^O$ and $(r_j^E, r_j^X) \propto A_j$ and domain $k \notin D.P$,
   (a) Update path $P_{new} = P \circ r_j^E \circ r_j^X$.
   (b) Generate path signature $S.P_{new}$.
   (c) Send Request = (Target Domain T, P_{new}, $S.P_{new}$) to domain k.
6. End.

Figure 6: Algorithm executed by domain j upon receiving a path request.

current access path. The path corruption could be divided into two types of attacks, namely path insertion and deletion.

The *path insertion attack* is performed by an attacker in an attempt to insert a domain in the path. Given $P_i = \{r_1^E, r_1^X, \ldots, r_i^E, r_i^X\}$, the attacker attempts to change it by inserting roles r_y^E, r_y^X in the path string, $\widetilde{P}_i = \{ r_1^E, r_1^X, \ldots, r_k^E, r_k^X, r_y^E, r_y^X, r_{k+1}^E, r_{k+1}^X, \ldots, r_i^E, r_i^X\}$. Figure 7(b) shows such an attack. The attacker is unable to generate the signature of the new path $\widetilde{S.P_i}$ as this requires the generation of new signatures $\widetilde{S.P_j}$, $k \leq j \leq i$ and this requires the knowledge of the secret keys e_j, for $k \leq j \leq i$. This shows that this path cannot be authenticated by the attacker.

The *path deletion attack* is performed by an attacker in an attempt to delete a domain in the path. Given $P_i = \{r_1^E, r_1^X, \ldots, r_i^E, r_i^X\}$ the attacker attempts to change it by deleting roles r_k^E, r_k^X in the path string, $\widetilde{P}_i = \{r_1^E, r_1^X, \ldots, r_{k-1}^E, r_{k-1}^X, r_{k+1}^E, r_{k+1}^X, \ldots, r_i^E, r_i^X\}$. Figure 7(c) shows such an attack. The attacker is unable to generate the signature of the new path $\widetilde{S.P_i}$ as this requires the generation of new signatures $\widetilde{S.P_j}$, $j \in \{k-1, k+1, \ldots, i\}$, which requires the knowledge of the corresponding secret keys. This shows that this path cannot be authenticated by the attacker.

Note that other types of attacks such as path reordering are not possible because the attacker cannot prove the authenticity of such path. Another type of attack in which domains in the collaboration collude to forge an access path, in this case two or more domains agree to provide cross links which did not exist. However, if these cross links only involve the colluding domains then this cannot be an attack because both domains agreed to provide such cross link. If the cross links involve domains other than the colluding domains, then this is easily detected from the path signature and when the path linking rules are executed.

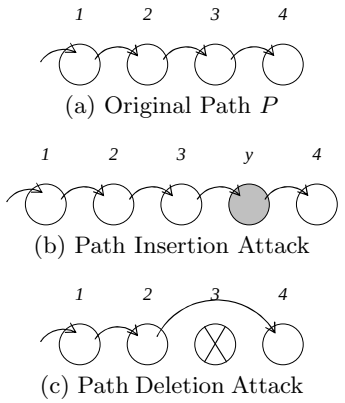


Figure 7: Types of path corruption attacks.

- **Path Replay Attacks.** An attacker could capture a request submitted during a valid session and try to replay such a request. This attack is not possible as for each session a new *seed* is used to authenticate the path and thus the captured request will have an old seed.
- **Denial of Service.** An attacker would request a role via a path that contains a loop $P = \{r_1, r_2, \ldots, r_n, r_1, r_2, \ldots\}$ and repeat such requests infinitely to increase the path size infinitely. Such an attack can be easily dealt with by introducing a bound on the permissible path size, which is basically the path cardinality constraint mentioned in Section 5.1, and the permissible path size can be set to double the number of domains present in the collaboration.
- **Violations of the Restricted Relation R.** In this attack a malicious domain involved in a restricted ac-

cess relation does not honor such relations. In such a case this domain gives access to a user that violates the restricted access relation R. This attack is easily detected by the neighboring domain, as such role access will be recorded in the user's access path. Furthermore, violating the restricted access relation will only directly affect the security of the malicious domain. Thus domains that do not abide by the path linking rules cause security violations to their own security policies.

9. RELATED WORK

The problem of secure interoperation in a multi-domain environment has been addressed in [11, 4]. In particular, Li Gong et al. [11] characterized the properties that must be satisfied to compose a global secure policy. They proposed the maximal secure interoperability problem and determined its complexity to be NP-Complete. In all such approaches a trusted third party that has a global view of the collaboration environment is required to perform the secure policy composition and integration.

Dawson et al. [7] presented a mediator based approach to provide secure interoperability for heterogeneous databases. This approach assumes a mandatory access control policy, such as the Bell LaPadula [2] policy, which is not flexible and not applicable in many commercial applications. Furthermore, all access requests go through the central mediator which has a global view of the collaboration environment. This approach is thus not appropriate for a dynamic distributed environment. Other approaches related to centralized database collaboration have been proposed in [20, 15, 24, 25]. Also these approaches have limited applicability because assume a centralized global view of the systems to be interoperated.

10. CONCLUSIONS

In this paper, we have presented a mediator-free collaboration environment in which domains collaborate in making localized access control decisions. We presented a framework to enable collaboration in such an environment where domains collaborate securely without needing a trusted mediator and without needing a global view of the collaboration environment. In our framework the user's access path is used to provide domains with enough information to make secure access control decisions using both basic and extended path linking rules. We also provided a path authentication scheme that ensures that the path is not tampered with as it propagates between domains.

Furthermore, we have provided proactive and on-demand path discovery algorithms that enable domains to discover available multi-hop collaborations. We also analyzed several security attacks that could be performed and showed how our framework can easily handle such attacks.

11. ACKNOWLEDGMENTS

The research of Mohamed Shehab and Arif Ghafoor has been supported by the sponsors of the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University, and the National Science Foundation under NSF Grant IIS-0209111. The research by Elisa Bertino was supported in part by the NSF under the Project "The Design and Use of Digital Identities", by an IBM Fellowship, and by the sponsors of CERIAS at Purdue University.

12. REFERENCES

[1] H. Afsarmanesh, C. Garita, and L. Hertzberger. Virtual Enterprises and Federated Information Sharing. In *DEXA'98: Proceedings of the 9th International Conference Database and Expert Systems Applications*, pages 374–383, Aug 1998.

[2] D. Bell and L. LaPadula. Secure Computer Systems: Mathematical Foundations. *Technical Report MTR-2547*, 1, March 1973.

[3] E. Bertino, E. Ferrari, and V. Atluri. The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. *ACM Transactions on Information and Systems Security*, 2(1):65–104, Feb 1999.

[4] P. Bonatti, M. Sapino, and V. Subrahmanian. Merging Heterogenous Security Orderings. *Journal of Computer Security*, 5(1):3–29, 1997.

[5] D. Brewer and M. Nash. The Chinese Wall Security Policy. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 206–214, 1989.

[6] J. Crampton. On Permissions, Inheritance and Role Hierarchies. In *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*, pages 85–92. ACM Press, Oct 2003.

[7] S. Dawson, S. Qian, and P. Samarati. Providing Security and Interoperation of Heterogeneous Systems. *Distributed Parallel Databases*, 8(1):119–145, 2000.

[8] A. Desai and N. Awad. Special Issue on Adaptive Complex Enterprises. *Communications of ACM*, 48(5), May 2005.

[9] D. Ferraiolo, D. Kuhn, and R. Chandramouli. Role-Based Access Control. *Artech House*, Apr 2003.

[10] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli. Proposed NIST Standard for Role-Based Access Control. *ACM Transactions on Information and Systems Security*, 4(3):224–274, Aug 2001.

[11] L. Gong and X. Qian. The Complexity and Composability of Secure Interoperation. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 190–200. IEEE Computer Society, 1994.

[12] L. Gong and X. Qian. Computational Issues in Secure Interoperation. *IEEE Transaction on Software and Engineering.*, 22(1), Jan 1996.

[13] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A Secure On-demand Routing Protocol for Adhoc Networks. In *MobiCom'02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pages 12–23. ACM Press, Sept 2002.

[14] D. Johnson, D. Maltz, and J. Broch. DSR: The Dynamic Source Routing Protocol for Multihop Wireless Adhoc Networks. *Ad hoc networking*, pages 139–172, 2001.

[15] D. Jonscher and K. Dittrich. An Approach for Building Secure Database Federations. In *VLDB'94: Proceedings of the 20th International Conference on Very Large Data Bases*, pages 24–35, Santiago de Chile, Chile, Sept 1994. Morgan Kaufmann.

[16] N. Li, Z. Bizri, and M. Tripunitara. On Mutually Exclusive Roles and Separation of Duty. In *CCS '04: Proceedings of ACM Conference on Computer and Communications Security*, Oct 2004.

[17] H. Ludwig, C. Bussler, M. Shan, and P. Grefen. Cross-Organisational Workflow Management and Co-ordination. In *WACC'99: Proceedings of the Workshop on Cross-Organisational Workflow Management and Co-ordination*, Feb 1999.

[18] J. Madhavan, P. Bernstein, A. Doan, and A. Halevy. Corpus-Based Schema Matching. In *ICDE '05: Proceedings of the Twenty First International Conference on Data Engineering*, April 2005.

[19] J. Madhavan and A. Halevy. Composing Mappings Among Data Sources. In *VLDB'2003 : Proceedings of the Twenty Ninth International Conference on Very Large Databases*, 2003.

[20] M. Morgenstern, T. Lunt, B. Thuraisingham, and D. Spooner. Security Issues in Federated Database Systems: Panel Contributions. In *Results of the IFIP WG 11.3 Workshop on Database Security V*, pages 131–148. North-Holland, 1992.

[21] R. Ramnath and D. Landsbergen. IT-Enabled Sense-and-Respond Strategies in Complex Public Organizations. *Communications of ACM*, 48(5):58–64, May 2005.

[22] R. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, Feb 1978.

[23] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-Based Access Control Models. *IEEE Computer*, 29(2):38–47, Feb 1996.

[24] S. Vimercati and P. Samarati. Authorization Specification and Enforcement in Federated Database Systems. *Journal of Computer Security*, 5(2):155–188, 1997.

[25] G. Wiederhold, M. Bilello, and C. Donahue. Web Implementation of a Security Mediator for Medical Databases. In *Proceedings of the IFIP TC11 WG11.3 Eleventh International Conference on Database Securty XI*, pages 60–72, London, UK, UK, 1998. Chapman & Hall, Ltd.