Secure Collaboration in a Mediator-Free Distributed Environment

Mohamed Shehab, *Member*, *IEEE Computer Society*, Arif Ghafoor, *Fellow*, *IEEE Computer Society*, and Elisa Bertino, *Fellow*, *IEEE Computer Society*

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 an on-demand path discovery algorithms that enable domains to securely discover paths in the collaboration environment. We implemented a simulation of our proposed framework and ran experiments to investigate the effect of several design parameters on our proposed access path discovery algorithm.

Index Terms—Distributed access control, access paths, role mapping, role 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 [3], [33]. Last but not the least, interoperability is essential to support adaptation and evolution in complex enterprises [17]. Such enterprises [41] 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. Collaboration could be viewed as an interoperation between the access control policies of the involved domains. 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 [25], 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 [25], [9], [16]. It is more challenging to handle security in a fully distributed and dynamic interoperation environment where domains join and leave in an ad hoc manner and in the absence of a trusted mediator. The Hurricane Katerina incident is a clear example of the need for the ad hoc and distributed collaboration among the different government agencies. The law enforcement agencies, Coast Guard, military, healthcare, healthcare supply chain (pharma and blood products), transport (EMS and MICUs), rescue teams, and many other agencies had to collaborate with one another to ensure efficient rescue and relief operations. Each of the collaborating agencies has its own access control policies, and to ensure secure collaboration among the agencies, none of these policies should be violated. The collaboration management protocols and algorithms should be able to efficiently and dynamically adapt to changes in the collaboration environment. Such collaboration cannot be managed in a centralized manner because of the complexity involved in ensuring a centralized secure collaboration [25]. We believe that the development of a decentralized secure collaboration solution tailored to dynamic ad hoc environments is crucial to meet the security requirements of this form of collaboration.

- *M. Shehab is with the Department of Software and Information Systems, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223. E-mail: mshehab@uncc.edu.*
- *A. Ghafoor is with the School of Electrical and Computer Engineering, Purdue University, 465 Northwestern Ave., West Lafayette, IN 47907. E-mail: ghafoor@ecn.purdue.edu.*
- *E. Bertino is with the Department of Computer Sciences, Purdue University, 250 N. University Street, West Lafayette, IN 47906. E-mail: bertino@cs.purdue.edu.*

Manuscript received 24 Oct. 2006; revised 20 July 2007; accepted 9 Jan. 2008; published online 1 Feb. 2008.
Recommended for acceptance by K. Hwang.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-0339-1006.
Digital Object Identifier no. 10.1109/TPDS.2008.26.

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 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 an on-demand path discovery algorithm that enable users to discover available secure access paths in the interoperation environment. We also provide experimental results of the access path discovery algorithm based on a simulation implementation of our proposed secure collaboration framework.

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 collaboration 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.
- We implement a simulation of our framework and show by experimental results the effects of several design parameters on our proposed access path discovery algorithm.

The rest of the paper is organized as follows: In Section 2, we review the requirements of secure interoperability, the maximal secure interoperability (MSI), and the drawbacks of the MSI solution proposed by Gong and Qian [25]. Furthermore, In Section 2, we introduce the mediator-free collaboration environment. In Section 3, 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 4 and 5, respectively. The proactive and on-demand path discovery algorithms are presented in Section 6. Possible security attacks and mitigation techniques are discussed in Section 7. The experimental results are presented 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 [20], [14] to model their access control policies. The protocols and algorithms presented in this paper can still be applied when other access control models are adopted. We have chosen RBAC because it is suitable for specifying the security requirements of a wide range of commercial, medical, government applications [43], [6], and moreover, it is being standardized by the National Institute of Standards [20]. A domain that does not use RBAC as its access control model can easily generate an export RBAC policy to join the collaboration.

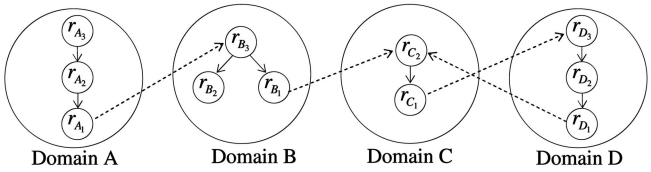

Fig. 1. Collaboration and violations.

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 [13]. For example, a Manager role dominates the Secretary role, giving a manager the ability to acquire access rights given to a secretary. 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

Collaboration among domains involves the interoperation of their access control policies. Domains typically achieve interoperation among their access control policies by introducing cross mappings between these policies. 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. We will refer to such mappings as cross-links. A cross-link (r_x, r_y) indicates that a user acquiring the role r_x in domain $D(r_x)$ is able to acquire role r_y in domain $D(r_y)$. Fig. 1 shows the cross-links as the dotted edges connecting roles in neighboring domains. Solutions developed for schema matching in the area of heterogeneous database systems and more recently approaches based on ontologies [36], [35] can be used for generating such links. 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. These links could be selected when the service level agreements (SLA) are negotiated [47], [15]. Furthermore, the domain administrators agree on a set of restricted cross-links that are prohibited during the collaboration. These restricted access links are similar to negative authorizations adopted in several access control

models [12], [20], [13], [14]. Separation of Duty constraints can be specified using a combination of the negative cross-links [31]. 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 during interoperation.

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, 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 Gong and Qian [25] 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 the restricted access relation is honored. The second requirement ensures the following two properties hold:

- **Autonomy.** It requires that any access permitted within 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.

To generate the resulting arc set A_Q Gong and Qian [25] defined the MSI problem as follows:

Definition 1. 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 secure interoperability is ensured. The MSI solution inherently satisfies the autonomy requirement as the arc set generated $A_Q = \{\cup_{i=1}^{n} A_i \cup S\}$ includes all the arcs A_i for each domain D_i, which ensures that any access permitted within an individual domain is also permitted under secure interoperation. Taking a closer look at the MSI solution, we conclude it has the following drawbacks:

- **NP-completeness.** Gong and Qian [25] 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 about 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; however, if a domain leaves or joins the collaboration, the MSI solution has to be recomputed to ensure both optimality and security. Furthermore, the MSI solution should be recomputed whenever 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 Fig. 2, 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 $\{r_{A_1}, r_{B_3}, r_{B_1}, r_{C_2}, r_{C_1}, r_{A_3}\}$, 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 the rights of users in other domains are not affected.

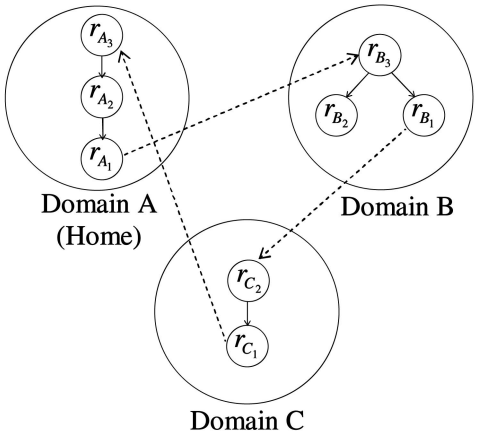

Fig. 2. Example of a violation in a multidomain environment with three domains. The solid lines show the internal access links, while the dotted lines show the interoperation cross-links F.

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 addition, note that the MSI solution maintains consistency with the autonomy requirement by limiting arc removal to the set of cross-links, refer to the definition of the

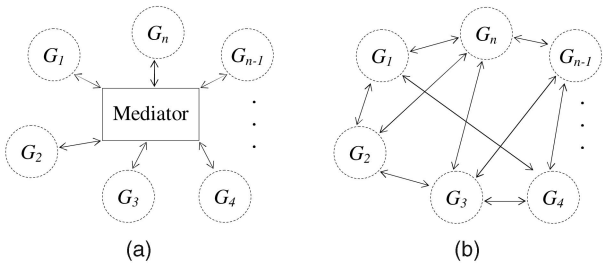

Fig. 3. Collaboration environment with and without a mediator. (a) Mediated. (b) Mediator-Free.

arc set A_Q. McDaniel and Prakash [37] pointed out that this assumption is restrictive, as it limits the search to the set of cross-links without investigating arcs inside domain hierarchies. In the next sections, we propose a secure technique that represents a computationally simple, distributed, dynamic solution, and ensures fairness to the participating domains.

2.2 Mediator-Free Secure Collaboration

In this section, we present the notion of mediator-free secure collaboration environment. In a mediator-free environment, there is no global entity ensuring secure interoperability among the collaborating domains. Fig. 3 shows both the mediated and the mediator-free types of collaboration environments. In a mediator-free environment, we have the following main assumptions:

- Domains have limited information about the collaboration environment. Each domain only has information about its own security policy, the cross-links and restricted links in which it is involved.
- Each domain is responsible for making its own access control decisions. The first priority of each domain is to ensure that its security policy is not violated.
- Domains are willing to collaborate in propagating access control messages and requests across domain boundaries.

To design a mediator-free secure collaboration framework, we need to analyze the functions performed by the mediator, which include the following:

- **Conflict resolution.** From the global view, the mediator computes the a secure solution, which generates the set of collaboration cross-links between the domains that satisfy the secure collaboration requirements.
- **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.

Enforcing secure interoperability in mediator-free 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. In this paper, we present a framework that enables secure collaboration in mediator-free environments and ensures that the security policies of domains are not violated.

3 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 concept of the history-based access control [18], [2] and the Chinese Wall security policy [11], 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 that does 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 possible future paths is dependent on his current access path.

3.1 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 Fig. 4):

- *Request processing module.* Enables domains to generate and evaluate user access requests across domains. Request processing enables domains to accumulate secure access paths and use these access paths to evaluate cross-domain access requests.
- *Path authentication module.* The user path migrates with the user requests, path authentication ensures is required to check the authenticity of the received paths. In addition, path authentication generates path signatures for generated requests.
- *Path discovery module.* Enables users residing in their home domain to discover secure access paths to roles accessible in target domains. Path discovery could be on-demand or proactive depending on the collaboration environment.

The above three modules are included in each domain. The modules interact with each other to ensure the security of their corresponding domains. The details of each module are 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.

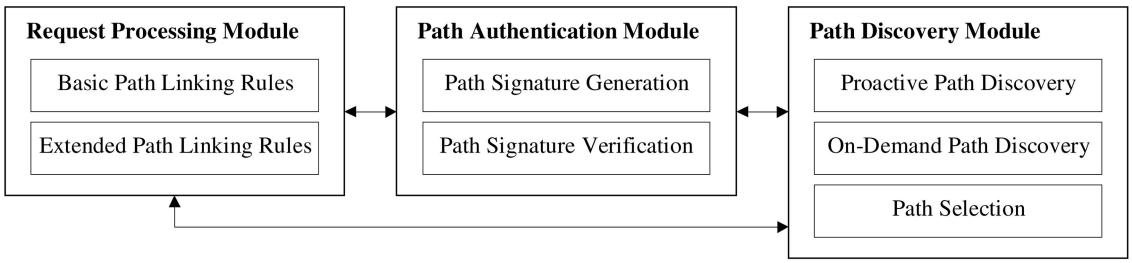

Fig. 4. Modules of the mediator-free secure interoperability framework.

3.2 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. Fig. 5 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 Fig. 5 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.*

The secure interoperability requirements that were discussed in Section 2.1 ensure that the global collaboration environment is secure, where the cross-links set F, the restricted access set R, and the domain hierarchies are honored. Our framework is based on access paths, we map the secure collaboration requirements to secure access path requirements. A secure access path is defined as follows:

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 $D(r_i)$ 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 $D(r_i) = D(r_j)$, 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 obey the dominance relationships between roles according to the domain's role hierarchy. This avoids the escalation of roles within the same domain and 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, respectively. 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 [28], [27] in which the route from source to destination is attached to the packet to enable routing of the packet. Thus, including access path information with access requests is an acceptable scalable assumption; in the following sections, we present techniques to limit the size of the access path.

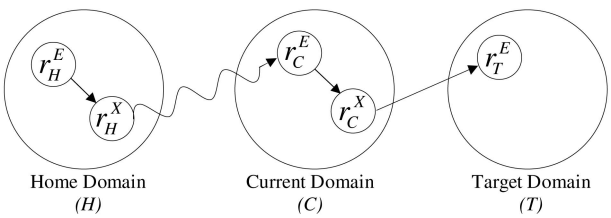

Fig. 5. Types of domains, entry and exit roles.

4 Request Processing Module

In a mediator-free collaboration environment, each domain has a limited view of the collaboration environment. Each domain has knowledge of its own access control policy, and 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, respectively. 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 be the exit role in the current domain, and r_T^E be 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 path linking rules are applied by the target domain when a request is made from a current domain to acquire a role in the target domain. 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 $r_n = r_C^X$. 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 that contradicts our assumption, 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-L3 are computationally simple operations and can be computed in polynomial time with respect to the path size or the number of domains. In addition, note that the path linking rules ensure secure collaboration, assuming a single path is generated per user session, which is consistent with the assumption of RBAC sessions [20], [14]. The basic linking rules are consistent with the autonomy requirement while, at the same time, satisfying the security requirement. Condition L3 of the basic path linking rules ensures that a role is added to the path only if it is allowable in the target domain, which is consistent with the autonomy requirement. Furthermore, the access path is built by encapsulating roles and by verifying the basic linking rules, this does not involve the removal of any cross-links, thus this makes it fair to all the domains involved the collaboration. For example, in Fig. 2, by using the access paths and the basic path linking rules, a user in domain A can securely acquire roles in domains B and C, a user in domain B can securely acquire roles in domains A and C, and a user in domain C can securely acquire roles in domains A and B.

4.1 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 [20], [14], [31]. In this section, we address these constraints in the context of a mediator free secure 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 [31]. Assuming the user's access path is P, then this general 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 constraints are relevant in the context of workflow systems [6], in which certain roles should be acquired before others roles can be activated.

5 Path Authentication

The access path is attached to the user's request, 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 path ordering. Each domain D_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 D_{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 D_k, respectively. The signature $S.P_i$ of path P_i is computed as follows:

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

where $\circ$ is the concatenation operator, $\bigoplus$ is the XOR operator, $h()$ is a secure one-way hash function, $SIGN_K(M)$ is a signature function that uses key K to sign message M, and the $nonce$ is a random number generated by the home domain and or can be equal to the session identifier, which is included in the path information. Domain D_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 D_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 [44]:

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

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

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

The target domain can easily check the authenticity of a path P_i by recursively computing the above equation

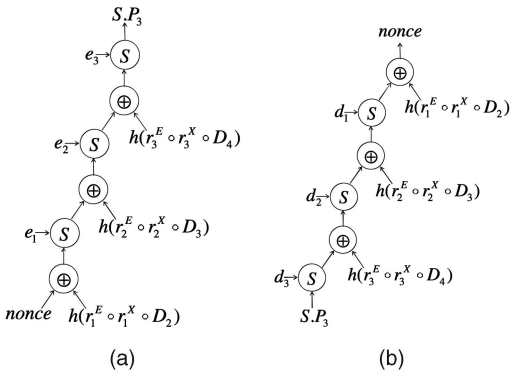

Fig. 6. The signature generation and verification for $\mathrm{Path} = \{r_1^E, r_1^X, r_2^E, r_2^X, r_3^E, r_3^X\}$, and S represents the signature function used. (a) Signature generation. (b) Signature verification.

and comparing $S.P_0$ with the $nonce$. 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. The path signature is carried with the path as it propagates between domains. Our signature scheme results in a single signature with a fixed bit-length that is independent of the path length. Fig. 6 depicts the signature generation and verification.

6 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. On the other hand, to collaborate through multihop collaborations domains need to build one or more candidate access paths to target domains. To enable domains to discover available multihop 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 security of the discovered path(s). In this section, we present the *on-demand* path discovery algorithm.

6.1 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 paths are needed. Neighboring domains do not exchange periodic path message updates; instead, messages are exchanged between neighboring domains only when cross-links are updated. Note that cross-link related to domain D_i can be divided into outgoing and incoming cross-links, referred to by F_i^O and F_i^I, respectively, where $F_i^O \cup F_i^I = F_i$. When a home domain needs to establish a path to a certain role in a target domain, a path request message is generated by the home domain and is sent on its outgoing cross-links. Upon

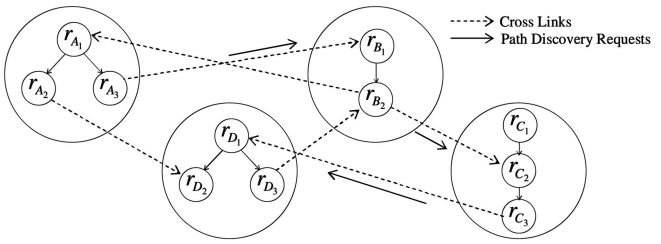

Fig. 7. Example of On-demand Path Discovery.

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. To ensure that paths do not grow uncontrollably, path lengths are limited to a maximum, which we refer to as P_{max}, only paths with lengths lower than P_{max} are propagated. Fig. 7 shows an example of an 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. In addition, 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.

After sending the path request, the home domain waits for a time-out period. If no reply arrives from the target domain, then this means there are no secure paths from the home domain to the target domain. The value of a time-out period is dependent on variables in the collaboration environment such as the number of collaborating domains and P_{max}. The time-out value is also dependent on the type of application, for example, certain applications are time sensitive and would require low time-out value. The path authenticity is ensured by using the path signature scheme discussed in Section 5, where the authentication signature is computed as the request message propagates between the domains.

The major advantage of on-demand path discovery is that it maintains paths to roles in target domains of interest to the current domain instead of maintaining a complete map of the whole collaboration environment. 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 delay incurred at the beginning of the collaboration caused by propagation of the path request message. Another problem is the large number of messages generated by a path request to arrive to the requested role. In the following sections, we present the *link selection (LS)* and *request inhibition (RI)* to reduce the number of propagated messages.

6.2 Link Selection

A single path discovery request message could generate several new forwarded path request messages. This could lead to a large number of messages generated by the path discovery algorithm. For example, in Fig. 8, assume a path

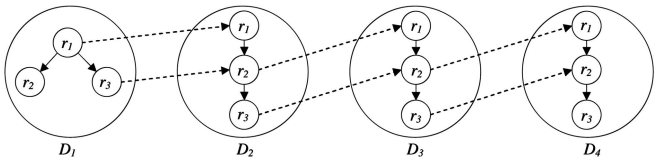

Fig. 8. Exponential message forwarding example.

discovery request arrives to role r_1 of domain D_1, executing the path discovery algorithm domain D_1 forwards this request on all outgoing cross-links reachable from role r_1. Domain D_1 forwards requests on cross-links $(r_1^{D_1}, r_1^{D_2})$ and $(r_3^{D_1}, r_2^{D_2})$. When D_2 receives the forwarded requests, it forwards each request on cross-links $(r_2^{D_2}, r_1^{D_3})$ and $(r_3^{D_2}, r_2^{D_3})$, which implies that domain D_2 sends four requests to domain D_3. In turn, D_3 forwards each of the received requests on cross-links $(r_2^{D_3}, r_1^{D_4})$ and $(r_3^{D_3}, r_2^{D_4})$, thus domain D_3 sends eight requests to domain D_4. Note the large number of messages generated. Each request is forwarded to all reachable outgoing cross-links without checking dominance relationships between the entry roles of target domains to which the requests are being forwarded. For example, in Fig. 8, domain D_1 forwards path requests on cross-links $(r_1^{D_1}, r_1^{D_2})$ and $(r_3^{D_1}, r_2^{D_2})$; however, the entry role $r_1^{D_2}$ dominates role $r_2^{D_2}$, which implies that all role reachable from $r_2^{D_2}$ are also reachable from $r_1^{D_2}$. LS limits the number of messages forwarded by selecting cross-links that have dominating entry roles in target domains. The LS algorithm is described in Fig. 9. The LS algorithm starts by the set of all cross-links that are reachable from the entry role, where the path request message originated. This set is refined by the LS algorithm to include only the cross-links that have dominating entry target roles. Note that the current domain only requires local information such as the domain policy and the set of outgoing cross-links to execute the LS algorithm. In the example in Fig. 8, if domains apply the LS algorithm, each domain ends up forwarding only a single message to its neighboring domain.

6.3 Request Inhibition

In this section, we present another approach to reduce the number of forwarded messages. The RI limits the number of times a path request is forwarded on the same cross-link. Each path request is assigned a unique identifier that is

Input: T = Crosslinks reachable from entry role r_j^E.
Output: Q = Selected cross links reachable from entry role r_j^E.
Algorithm:
1) $Q = T$
2) For all cross links $l = (r_j^X, r_k^E) \in S$
   a) If there exits a crosslink $c = (s, r) \in S$ such that $r_k^E \prec r$ then $Q - l$.
3) Return Q.

Fig. 9. Algorithm executed by domain j upon receiving a path request.

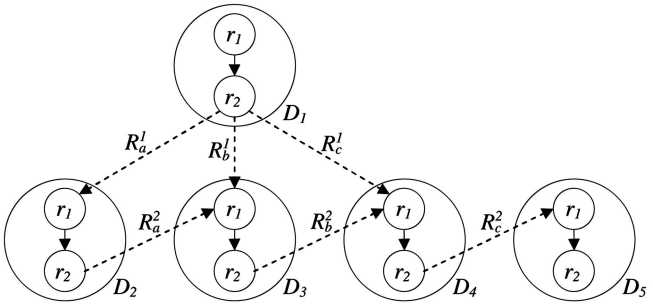

Fig. 10. RI example.

generated by the home domain. The request path identifier is replicated with each forwarded path request to enable domains to identify path requests. A simple identifier of a path request is the start role and the requested role. As path request is forwarded, the domain keeps a log of the request identifier and the cross-link on which the request was forwarded. Using the logged information, domains are able to ensure that path requests are only forwarded at most once on each cross-link. Note that the log can be flushed periodically or when it reaches a certain size. RI is an opportunistic technique as it gives preference to requests that arrive first at a given cross-link. This technique also promotes the propagation of requests that follow shorter paths, as these requests are more likely to arrive at a cross-link first. For example, in Fig. 10, assume that domain D_1 sends a path request from role r_2, requesting an access path to role r_2 in domain D_5. Domain D_1 generates requests R_a^1, R_b^1, and R_c^1 forwarded to domains D_2, D_3, and D_4, respectively. Note that all the requests R_a^1, R_b^1, and R_c^1 have the same request identifier. Upon receiving request R_a^1, domain D_2 forwards request R_a^2 to domain D_3. Similarly, domain D_3 forwards R_b^2 to domain D_4 and domain D_4 forwards R_c^2 to domain D_5. Note that cross-links $(r_2^{D_3}, r_1^{D_4})$ and $(r_2^{D_4}, r_1^{D_5})$ have logged the request identifiers of R_b^2 and R_b^2, respectively. The request R_a^2 is not further propagated from D_3 to domain D_4 on cross-link $(r_2^{D_3}, r_1^{D_4})$ as log indicates that a request having a similar identifier has already been propagated on this cross-link, similar is the case with request R_b^2. Note that this limits the propagation of the path requests, and the only successful propagated path request is R_c^2, which leads to the shortest path from role $r_1^{D_1}$ to role $r_2^{D_5}$. This technique will not always lead to the generation of the shortest path; however, it tends to favor the propagation of requests with shorter paths.

6.4 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.

7 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 collaboration framework is resilient to these attacks. We assume all access control messages exchanged between domains are sent over secure reliable channels.

- **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 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, $\widetilde{P}_i = \{r_1^E, r_1^X, \ldots, r_k^E, r_k^X, r_y^E, r_y^X, r_{k+1}^E, r_{k+1}^X, \ldots, r_i^E, r_i^X\}$. 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, $\widetilde{P}_i = \{r_1^E, r_1^X, \ldots, r_{k-1}^E, r_{k-1}^X, r_{k+1}^E, r_{k+1}^X, \ldots, r_i^E, r_i^X\}$. 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 that did not exist. If the cross-links involve domains other than the colluding domains, then this is easily detected when the path signature is verified. However, if these cross-links only involve the colluding domains, then this implies that these domains have agreed to established such a cross-link. In this case, this cross-link is similar to any other cross-link included in the path, and the path signature is also valid. The decision to accept or reject this path is dependent on the target domain's access control policy.
- **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 we assume that for each session, a new $nonce$ is used to authenticate the path.
- **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 4.1, and the permissible path size can be set to double the number of domains present in the collaboration. Another form of denial of service could be performed when malicious domain floods neighboring domains with path requests. Domains can prevent such an attack by restricting the frequency of requests [38].
- **Violations of the restricted relation R.** In this attack, a malicious domain involved in a restricted access 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.

8 Experimental Results

This section describes the performance evaluations of the path discovery algorithm obtained by a simulation implementation of the secure collaboration environment. All the experiments were performed on Intel Pentium IV CPU 3.2-GHz with 512 Mbytes of RAM and running Linux. We used the Java J2SE v5.0 and the Psim-J simulation library [32]. Each domain is simulated as a process having an input message queue in which the path discovery requests are received and forwarded. Each domain has a domain hierarchy, which is approximated by a binary tree. Domains maintain statistics about the forwarded messages and the discovered access paths. Neighboring domains are selected with a neighborhood probability p, then the cross-links are randomly generated between the neighboring domains. To evaluate the performance of the path discovery algorithm each domain in the collaboration environment generated path request messages and recorded several metrics to enable the evaluation of the path discovery algorithm. The collected metrics for each path request include the discovered path length, the number of forwarded messages, the number of discovered domains, and the number of replies received for each request. Each domain generated 1,000 path request

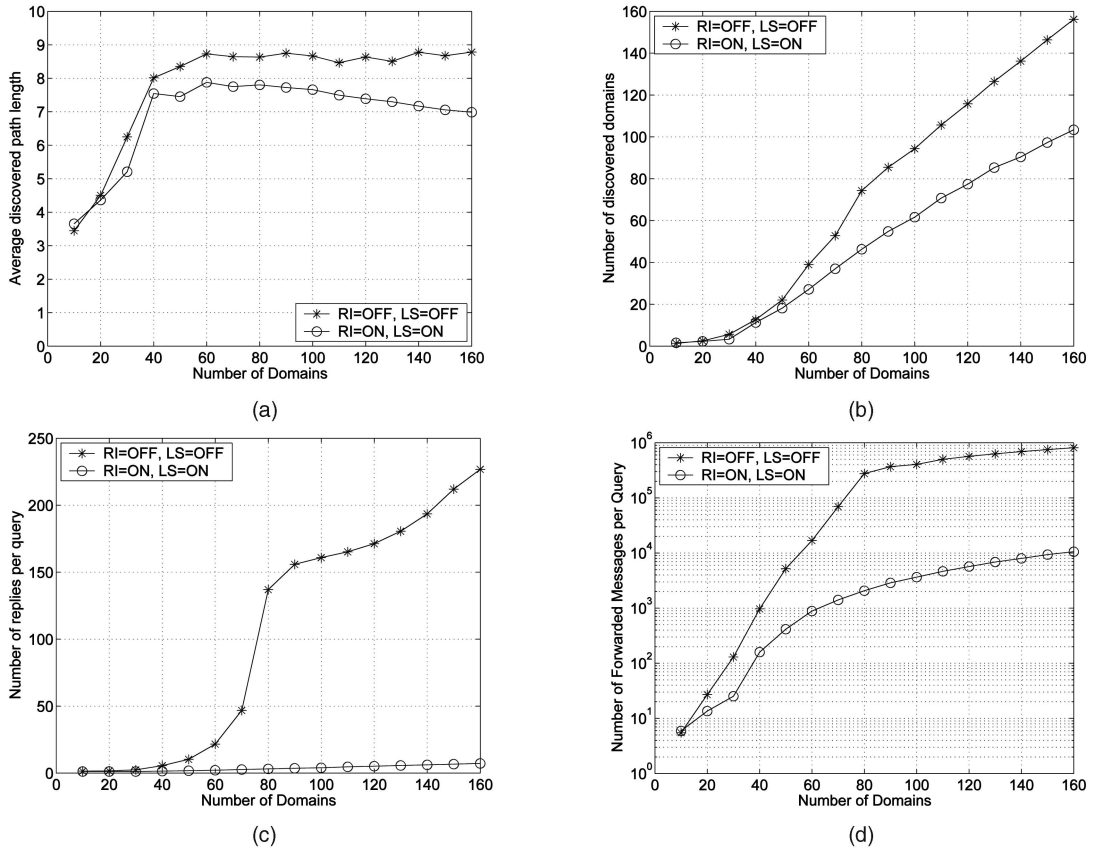

Fig. 11. Path discovery and the LS&RI algorithms.

messages, and the collected statistics were averaged over all requests and all domains. In the following sections, we present the effects of LS and RI and several parameters such as $Pmax$, p, and the number of domains have on the collected path discovery metrics.

8.1 Effects of Link Selection and Request Inhibition

Several experiments were executed with and without the LS and RI algorithms to investigate their effects on the path discovery. In the following presented experiments, the neighborhood probability p was set to 0.1, the maximum path length $Pmax$ was set to 15, and the number of domains in the collaboration environment was varied between 10 to 160 domains. Fig. 11 shows the effects of LS and RI on the different discovery metrics. The LS and RI reduce the average discovered path length (see Fig. 11a), this is inherently the effect of RI that favors shorter paths, as discussed in Section 6, this also reduces the number of discovered domains, as indicated in Fig. 11b. Furthermore, the LS&RI algorithms limit the query flow, which leads to lower number of the returned replies, as can be seen in Fig. 11c. The major advantage of the LS&RI algorithms is that they limit the number of forwarded messages, Fig. 11d shows that the LS&RI algorithms reduce the number of forwarded messages up to two degrees of magnitude. Experiments were performed for different p values, and they showed similar trends.

8.2 Effects of $Pmax$

The value of $Pmax$ controls the maximum allowable discovered path length. In this section, the value of $Pmax$ was varied and metrics were collected for collaboration environments, which include 50, 75, 100, and 150 domains. The neighborhood probability p was set to 0.1, and in all experiments, LS&RI algorithms were used. Fig. 12 shows the generated results of such experiments for 50, 75, 100, and 150 domains. Note that beyond a certain $Pmax$, all the collected metrics (Table 1) reach a plateau, for example, in Fig. 12a, the average path length for 150 domains case stays at 7 for values of $Pmax \geq 20$. By setting $Pmax$ to values lower than the plateau, we are able to control the behavior of the discovery algorithm, for example, in Fig. 12c, by setting $Pmax$ to 10, we are able to lower the number replies per query to about three replies.

8.3 Effects of the Neighborhood Probability p

The neighborhood probability is the probability by which domains become neighbors. In the presented experiments, both the number of domains and p were varied. The value of $Pmax$ was set to 15. Fig. 13 shows the simulation results. At high p values, more domains are direct neighbors, which should directly decrease the average discovered path length (see Fig. 13a). The denser the collaboration environment, the larger the number of discovered domains, as indicated in Fig. 13b. Note that as the number of neighbors increase, so does the number of messages forwarded and the number of replies per request, as indicated in Figs. 13c and 13d.

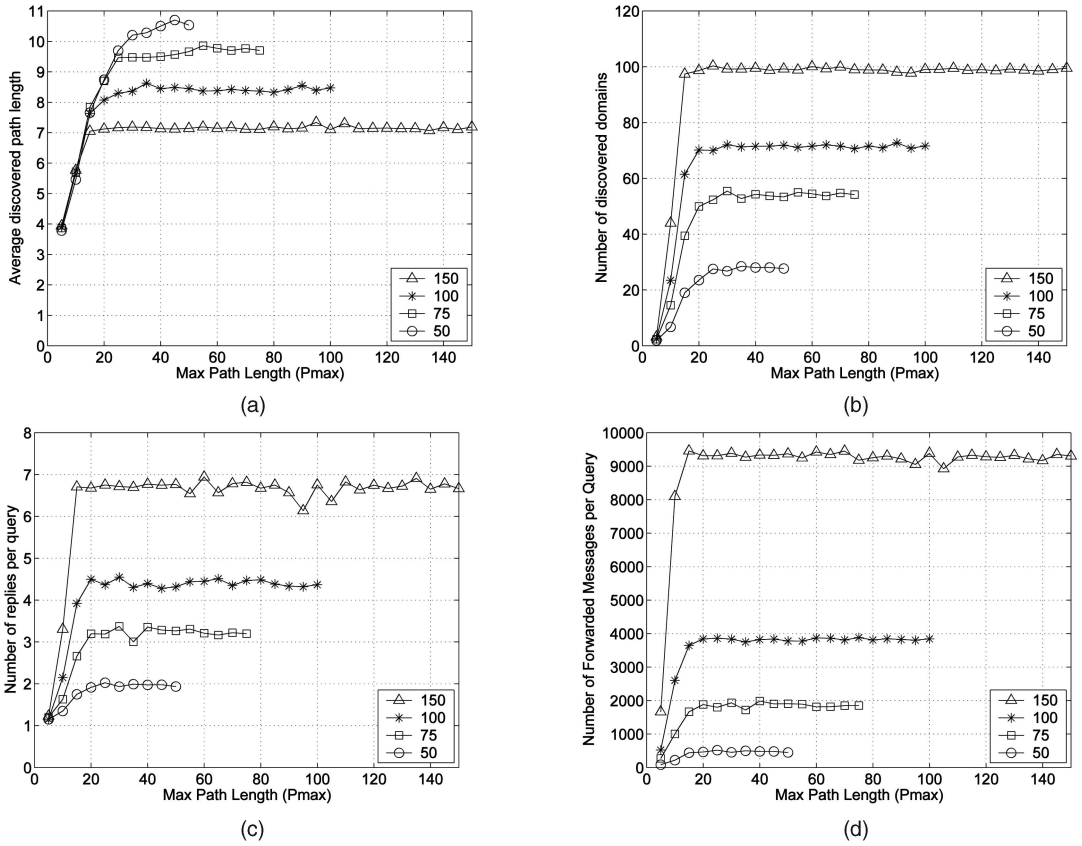

Fig. 12. Varying the value of $Pmax$.

9 Related Work

The problem of secure interoperation in a multidomain environment has been addressed in [24] and [9]. In particular, Gong and Qian [24] characterized the security and autonomy properties that must be satisfied to compose a global secure policy. Furthermore, Gong and Qian formulated the MSI problem and determined its complexity to be NP-Complete. The provided solutions require a trusted third party that has a global view of the collaboration environment to resolve policy conflicts and enable the secure policy composition and integration.

Bonatti et al. [9] handle the problem of integrating multiple heterogenous databases that involves the merging of multiple security orderings into a single unified ordering that preserves the security relationships between orderings. Their solutions is based on a logic programming approach that is based on constraint relaxation to achieve a global secure policy that obeys the maximum set of ordering constraints. Their solution assumes the presence of a central entity that has all the logic programming ordering constraints and manages the global database policy integration.

TABLE 1
Implication of Collected Metrics

Metric	Implication
Discovered Path Length	Is an indication of on average number of hops the path discovery protocol propagates before the target role is located.
# of forwarded messages	Is an indication on the average number of messages forwarded by the discovery protocol per request. This metric represents the cost of path discovery protocol.
# of discovered domains	Is an indication of the connectivity of the collaboration environment. Represents the average number of domains that are reachable from a given domain.
# of replies per request	Gives an indication of the average number of replies received for each request generated. It is also an indication of the average number of different access paths discovered for each target role.

Secure policy reconciliation is another related problem that provides a mechanism by which divergent policy requirements of session participants can be met. McDaniel and Prakash [37] have addressed this problem and provided a tractable algorithm for the reconciliation of two policies. The algorithm attempts to identify a policy instance compliant with the stated policy requirements. They show that the reconciliation of three or more policies is not tractable. In [37], the policies are submitted to a central entity referred to as the *initiator*. The initiator uses the reconciliation algorithm to create a policy instance compliant with the session and each domain policy.

Several research efforts [34], [16], [10], [46] have been devoted to establish secure interoperation by composing a consistent and conflict-free global interoperation policy that governs all the interdomain information and resource exchange. Policy composition is mainly concerned with the detection and resolution of conflicts in the interoperation policies. Lupu and Sloman [34] investigated the various precedence relationships that can be established between policies in order to allow inconsistent policies to coexist within the system. Furthermore, they presented a conflict

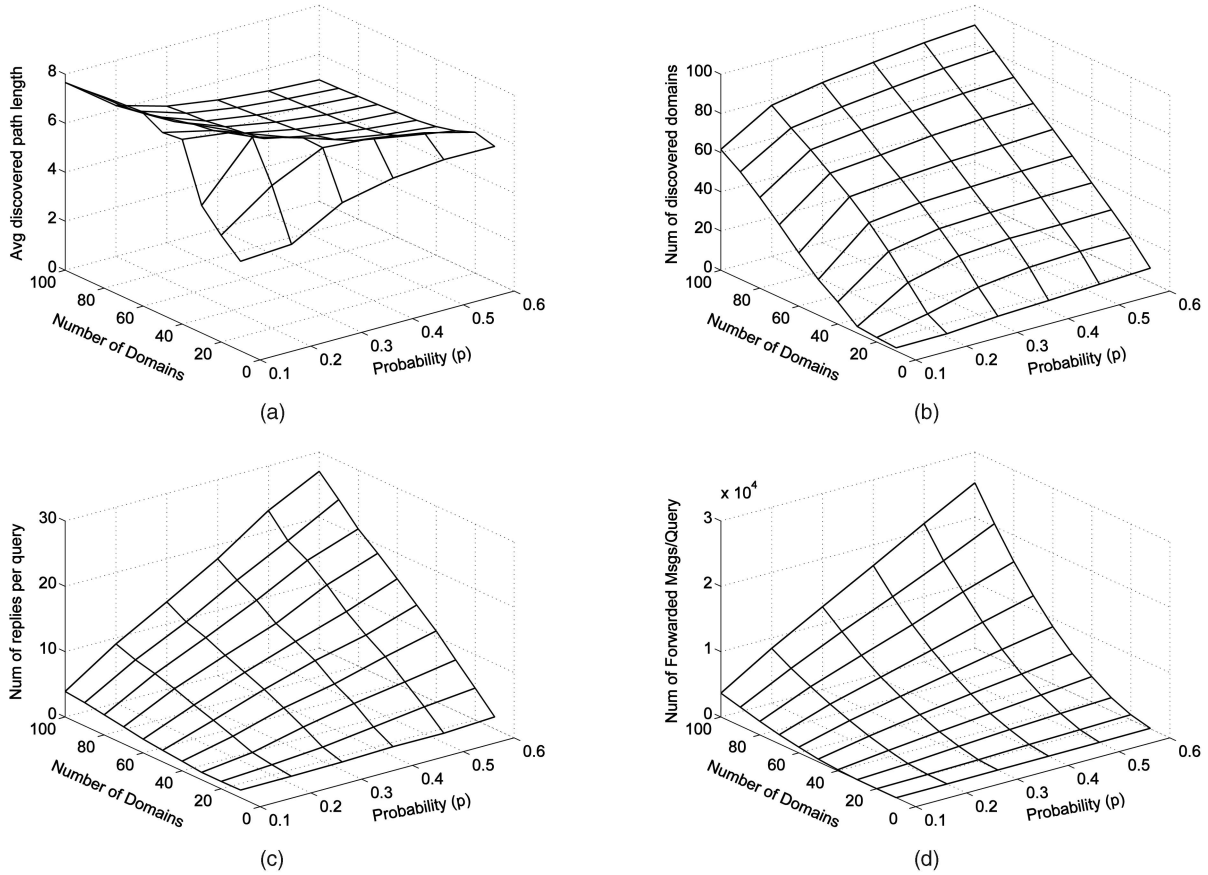

Fig. 13. Varying the neighborhood probability p.

analysis tool as part of the RBAC management framework. Dawson et al. [16] provided a policy integration framework based on a mediator to establish secure interoperation among heterogenous databases with hierarchical policies. This approach assumes a mandatory access control policy such as the Bell and LaPadula [5] 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 makes this approach not suitable for a dynamic distributed environment. Conflict resolution is handled by the policy editor, which determines the consistency of the global interoperation policy. Cyclic conflicts in the global interoperation policy are resolved by withdrawing cross-domain relationships that are included in the conflict. This approach provides on guarantee on the optimality of the resulting global policy. Other approaches related to centralized database collaboration have been proposed in [39], [29], [48], and [50]. In addition, these approaches have limited applicability for mediator-free environments because of the assumption of a central mediator having a global view of the collaboration environment. Bonatti et al. [10] addressed the problem of combining authorization specifications by proposing an algebra of security policies as a composition language and illustrated how to formulate complex policies in the algebra and reason about them.

Shafiq et al. [46] proposed a policy integration framework for merging RBAC polices of multiple domains into a global access control policy that governs information and resource accesses between the collaborating domains. This approach formulates the MSI problem as an integer programming (IP)-based problem to provide an optimal conflict resolution based on maximizing the interdomain role accesses. The framework assumes a central third party to solve the IP problem and to manage all cross-domain access requests.

Keoh et al. [30] presented a policy-based technique for the establishment of Ad hoc communities, where an ad hoc network is perceived as a community of autonomous devices that collaborate and share resources with each other. Such technique assumes unified policies among the different collaborating entities and does not support multihop collaborations.

Automated trust negotiation (ATN) [55], [54], [57], [52], [56] is an approach to access control and authentication in open and distributed systems. ATN enables resource access by assigning an access control policy to each resource describing the properties of the parties that can access these resources. The properties consist of digital credentials and certificates. Trust negotiation enables a domain to gradually establish trust with other domains through the iterative exchange and disclosure of digital credentials. Furthermore, ATN systems provide mechanisms to ensure the privacy of the presented credentials [51], [53], [45]. ATN enables users to access resources across domain boundaries. In a secure collaboration environment, ATN would enable domains to establish cross-links with other domains. ATN systems only

establish point-to-point collaboration and do not provide a framework for multihop collaboration, as in secure collaboration environments. Furthermore, ATN systems do not provide a mechanism for access path propagation and discovery.

Delegation is an important factor for a secure distributed computing environment. There are many definitions of delegation in the literature [21], [1], [22], [4], [58]. Delegation involves a subject passing its authority to other subjects to carry out some functions on behalf of the former. Process to process delegation in an object-oriented environment has been introduced by Nagaratnam and Lea [40]. Sandhu et al. address delegation related to role administrators in ARBAC97 model [42]. Delegation was treated as an attribute of a role by Goh and Baldwin [23]. The RBDM0 system allowed user-to-user delegation based on roles [4]. This allowed users to delegate role membership to other users. Zhang et al. [58] present the RDM2000, which extends RBDM0 and supports delegation in presence of a role hierarchy and provides a rule-based declarative language to specify and enforce delegation and revocation policies. Wang and Osborn [49] provided a delegation scheme in the role graph model that allows the delegation of roles within the same organization and, at the same time, obey the role graph constraints. Similar approaches have been introduced to enable delegation and revocation in database systems [26], [19], [7], [8]. Delegation is enabled by a delegation server that maintains the delegation graph to be able to handle sophisticated delegation and revocation. Role-based delegation schemes [23], [42], [4], [58], [49] are able to handle delegations and revocations within domain boundaries and provide no mechanisms to handle cross-domain delegations.

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 an on-demand path discovery algorithm that enable domains to discover available multihop collaborations. We analyzed several security attacks that could be performed and showed how our framework can easily handle such attacks. We also provided experimental results generated by simulating the collaboration environment.

References

[1] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin, "A Calculus for Access Control in Distributed Systems," *ACM Trans. Programming Languages and Systems,* vol. 15, no. 4, pp. 706-734, Sept. 1993.

[2] M. Abadi and C. Fournet, "Access Control Based on Execution History," *Proc. 10th Ann. Network and Distributed System Symp. (NDSS),* 2003.

[3] H. Afsarmanesh, C. Garita, and L. Hertzberger, "Virtual Enterprises and Federated Information Sharing," *Proc. Ninth Int'l Conf. Database and Expert Systems Applications (DEXA '98),* pp. 374-383, Aug. 1998.

[4] E. Barka and R. Sandhu, "A Role-Based Delegation Model and Some Extensions," *Proc. 16th Ann. Computer Security Application Conf.,* pp. 11-15, Dec. 2000.

[5] D. Bell and L. LaPadula, "Secure Computer Systems: Mathematical Foundations," Technical Report MTR-2547, vol. 1, Mar. 1973.

[6] E. Bertino, E. Ferrari, and V. Atluri, "The Specification and Enforcement of Authorization Constraints in Workflow Management Systems," *ACM Trans. Information and Systems Security,* vol. 2, no. 1, pp. 65-104, Feb. 1999.

[7] E. Bertino, P. Samarati, and S. Jajodia, "Authorizations in Relational Database Management Systems," *Proc. First ACM Conf. Computer and Comm. Security (CCS '93),* pp. 130-139, 1993.

[8] E. Bertino and R. Sandhu, "Database Security-Concepts, Approaches, and Challenges," *IEEE Trans. Dependable Secure Computing,* vol. 2, no. 1, pp. 2-19, 2005.

[9] P. Bonatti, M. Sapino, and V. Subrahmanian, "Merging Heterogenous Security Orderings," *J. Computer Security,* vol. 5, no. 1, pp. 3-29, 1997.

[10] P. Bonatti, S.D.C. Vimercati, and P. Samarati, "An Algebra for Composing Access Control Policies," *ACM Trans. Information and System Security,* vol. 5, no. 1, Feb. 2002.

[11] D. Brewer and M. Nash, "The Chinese Wall Security Policy," *Proc. IEEE Symp. Security and Privacy (SP '89),* pp. 206-214, 1989.

[12] D. Clark and D. Wilson, "A Comparison of Commercial and Military Computer Security Policies," *Proc. IEEE Symp. Security and Privacy (SP '87),* pp. 184-194, 1987.

[13] J. Crampton, "On Permissions, Inheritance and Role Hierarchies," *Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03),* pp. 85-92, Oct. 2003.

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

[15] A. Dan, D. Davis, R. Kearney, R. King, A. Keller, D. Kuebler, H. Ludwig, M. Polan, M. Spreitzer, and A. Youssef, "Web Services on Demand: WSLA-Driven Automated Management," *IBM Systems J.,* special issue on utility computing, vol. 43, no. 1, pp. 136-158, Mar. 2004.

[16] S. Dawson, S. Qian, and P. Samarati, "Providing Security and Interoperation of Heterogeneous Systems," *Distributed Parallel Databases,* vol. 8, no. 1, pp. 119-145, 2000.

[17] A. Desai and N. Awad, "Special Issue on Adaptive Complex Enterprises," *Comm. ACM,* vol. 48, no. 5, May 2005.

[18] G. Edjlali, A. Acharya, and V. Chaudhary, "History-Based Access Control for Mobile Code," *Proc. First ACM Conf. Computer and Comm. Security (CCS '98),* pp. 38-48, 1998.

[19] R. Fagin, "On an Authorization Mechanism," *ACM Trans. Database Systems,* vol. 3, no. 3, pp. 310-319, 1978.

[20] D. Ferraiolo, R. Sandhu, S. Gavrila, D. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control," *ACM Trans. Information and Systems Security,* vol. 4, no. 3, pp. 224-274, Aug. 2001.

[21] M. Gasser and E. Mcdermott, "An Architecture for Practical Delegation a Distributed System," *Proc. IEEE CS Symp. Research in Security and Privacy,* pp. 7-9, May 1990.

[22] H. Gladney, "Access Control for Large Collections," *ACM Trans. Information Systems,* vol. 15, no. 2, pp. 154-194, Apr. 1997.

[23] C. Goh and A. Baldwin, "Towards a More Complete Model of Role," *Proc. Third ACM Workshop Role-Based Access Control,* Oct. 1998.

[24] L. Gong and X. Qian, "The Complexity and Composability of Secure Interoperation," *Proc. IEEE Symp. Security and Privacy (SP '94),* pp. 190-200, 1994.

[25] L. Gong and X. Qian, "Computational Issues in Secure Interoperation," *IEEE Trans. Software and Eng.,* vol. 22, no. 1, Jan. 1996.

[26] P. Griffiths and B. Wade, "An Authorization Mechanism for a Relational Database System," *ACM Trans. Database Systems,* vol. 1, no. 3, pp. 242-255, 1976.

[27] Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Adhoc Networks," *Proc. Eighth Ann. Int'l Conf. Mobile Computing and Networking (MobiCom '02),* pp. 12-23, Sept. 2002.

[28] D. Johnson, D. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Adhoc Networks," *Ad Hoc Networking,* pp. 139-172, 2001.

[29] D. Jonscher and K. Dittrich, "An Approach for Building Secure Database Federations," *Proc. 20th Int'l Conf. Very Large Data Bases (VLDB '94)*, pp. 24-35, Sept. 1994.
[30] S. Keoh, E. Lupu, and M. Sloman, "PEACE: A Policy-Based Establishment of Ad-Hoc Communities," *Proc. 20th Ann. Computer Security Applications Conf. (ACSAC '04)*, pp. 386-395, Dec. 2004.
[31] N. Li, Z. Bizri, and M. Tripunitara, "On Mutually Exclusive Roles and Separation of Duty," *Proc. ACM Conf. Computer and Comm. Security (CCS '04)*, Oct. 2004.
[32] Psim-J Simulation Library, http://science.kennesaw.edu/jgarrido/psimj.html, 2008.
[33] H. Ludwig, C. Bussler, M. Shan, and P. Grefen, "Cross-Organisational Workflow Management and Co-Ordination—WACC," *99 Workshop Report*, vol. 20, no. 1, 1999.
[34] E. Lupu and M. Sloman, "Conflicts in Policy-Based Distributed Systems Management," *IEEE Trans. Software Eng.*, vol. 25, no. 6, pp. 852-869, Nov. 1999.
[35] J. Madhavan, P. Bernstein, A. Doan, and A. Halevy, "Corpus-Based Schema Matching," *Proc. 21st Int'l Conf. Data Eng. (ICDE '05)*, Apr. 2005.
[36] J. Madhavan and A. Halevy, "Composing Mappings among Data Sources," *Proc. 29th Int'l Conf. Very Large Databases (VLDB)*, 2003.
[37] P. McDaniel and A. Prakash, "Methods and Limitations of Security Policy Reconciliation," *Proc. IEEE Symp. Security and Privacy*, pp. 73-87, May 2002.
[38] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, *Internet Denial of Service: Attack and Defense Mechanisms.* Prentice Hall, 2005.
[39] M. Morgenstern, T. Lunt, B. Thuraisingham, and D. Spooner, "Security Issues in Federated Database Systems: Panel Contributions," *Proc. Results of the IFIP WG 11.3 Workshop Database Security*, pp. 131-148, 1992.
[40] N. Nagaratnam and D. Lea, "Secure Delegation for Distributed Object Environments," *Proc. Usenix Conf. Object Oriented Technologies and Systems*, Apr. 1998.
[41] R. Ramnath and D. Landsbergen, "IT-Enabled Sense-and-Respond Strategies in Complex Public Organizations," *Comm. ACM*, vol. 48, no. 5, pp. 58-64, May 2005.
[42] R. Sandhu, V. Bhamidipati, and Q. Munawer, "The ARBAC97 Model for Role-Based Administration of Roles," *ACM Trans. Information and System Security*, vol. 2, no. 1, Feb. 1999.
[43] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-Based Access Control Models," *IEEE Computer*, vol. 29, no. 2, pp. 38-47, Feb. 1996.
[44] B. Schneier, *Applied Cryptography*, second ed. John Wiley & Sons, 1996.
[45] K. Seamons, M. Winslett, and T. Yu, "Limiting the Disclosure of Access Control Policies during Automated Trust Negotiation," *Proc. Symp. Network and Distributed System Security (NDSS '01)*, Feb. 2001.
[46] B. Shafiq, J. Joshi, E. Bertino, and A. Ghafoor, "Secure Interoperation in a Multidomain Environment Employing RBAC Policies," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 11, pp. 1557-1577, 2005.
[47] *Use SLAs in a Web Services Context, Part 1: Guarantee Your Web Service with a SLA*, http://www-128.ibm.com/developerworks/library/ws-sla/, Oct. 2004.
[48] S. Vimercati and P. Samarati, "Authorization Specification and Enforcement in Federated Database Systems," *J. Computer Security*, vol. 5, no. 2, pp. 155-188, 1997.
[49] H. Wang and S. Osborn, "Delegation in the Role Graph Model," *Proc. 11th ACM Symp. Access Control Models and Technologies (SACMAT '06)*, pp. 91-100, 2006.
[50] G. Wiederhold, M. Bilello, and C. Donahue, "Web Implementation of a Security Mediator for Medical Databases," *Proc. IFIP TC11 WG11.3 11th Int'l Conf. Database Security*, pp. 60-72, 1998.
[51] W. Winsborough and N. Li, "Protecting Sensitive Attributes in Automated Trust Negotiation," *Proc. ACM Workshop Privacy in the Electronic Soc.*, pp. 41-51, 2002.
[52] W. Winsborough and N. Li, "Towards Practical Automated Trust Negotiation," *Proc. Third Int'l Workshop Policies for Distributed Systems and Networks (POLICY '02)*, pp. 92-103, June 2002.
[53] W. Winsborough and N. Li, "Safety in Automated Trust Negotiation," *Proc. IEEE Symp. Security and Privacy*, pp. 147-160, May 2004.
[54] W. Winsborough, K. Seamons, and V. Jones, "Automated Trust Negotiation," *Proc. DARPA Information Survivability Conf. and Exposition*, vol. 1, pp. 88-102, Jan. 2000.
[55] T. Yu, X. Ma, and M. Winslett, "PRUNES: An Efficient and Complete Strategy for Automated Trust Negotiation over the Internet," *Proc. Seventh ACM Conf. Computer and Comm. Security*, pp. 210-219, 2000.
[56] T. Yu and M. Winslett, "Unified Scheme for Resource Protection in Automated Trust Negotiation," *Proc. IEEE Symp. Security and Privacy*, pp. 110-122, May 2003.
[57] T. Yu, M. Winslett, and K. Seamons, "Interoperable Strategies in Automated Trust Negotiation," *Proc. Eighth ACM Conf. Computer and Comm. Security*, pp. 146-155, Nov. 2001.
[58] L. Zhang, G. Ahn, and B. Chu, "A Rule-Based Framework for Role-Based Delegation and Revocation," *ACM Trans. Information and System Security*, vol. 6, no. 3, pp. 404-441, 2003.

Mohamed Shehab received the PhD degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana. He is currently working as an assistant professor in the Department of Software and Information Systems, University of North Carolina, Charlotte. His research interests include information security, distributed access control, distributed workflow management systems, and watermarking of relational databases. He is a member of the IEEE Computer Society.

Arif Ghafoor is currently a professor in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana, and is the director of Distributed Multimedia Systems Laboratory and Information Infrastructure Security Research Laboratory. He has been actively engaged in research areas related to database security, parallel and distributed computing, and multimedia information systems and has published extensively in these areas. He has served on the editorial boards of various journals. He is a fellow of the IEEE Computer Society. He has received the IEEE Computer Society 2000 Technical Achievement Award for his research contributions in the area of multimedia systems.

Elisa Bertino is a professor of computer science and electrical and computer engineering at Purdue University and serves as a research director of CERIAS. Her main research interests include security and database systems. In those areas, she has published more than 300 papers. She is the coordinating editor in chief of the *Very Large Database Systems Journal* and serves on the editorial boards of several journals. She is a fellow of the IEEE Computer Society and the ACM. She received the 2002 IEEE Computer Society Technical Achievement Award for outstanding contributions to database systems and database security and advanced data management systems.