Information Flow in Disaster Management Systems

Achim D. Brucker
SAP Research
Vincenz-Priessnitz-Str. 1
76131 Karlsruhe, Germany
Email: achim.brucker@sap.com

Dieter Hutter
DFKI GmbH
Enrique Schmidt Str. 5
28359 Bremen, Germany
E-mail: hutter@dfki.de

***Abstract*—Collaborations between organizations in the public sector, e. g., fire brigades, polices, military units, is often done via liaison officers. A liaison officer liaises between two organizations by providing a single point of contact and ensuring the efficient communication and coordination of their activities. Usually an organization embeds a liaison officer in another organization to provide face-to-face coordination. Liaison officers demand special requirements to the security mechanism of the IT infrastructure of the organization that act as host for a liaison officer.**

This holds, in particular, for Disaster Management Information Systems (DMIS). Such systems need, on the one hand, to support various ways of communication in a flexible and ad hoc manner. On the other hand, these systems need to protect, by law, the leakage of sensitive data.

In this paper, we present a novel mechanism, based on role-based access control (RBAC), for supporting the flexible and secure information exchange between organizations using liaison officers. Our mechanism enables liaison officers to decide on their own authority which information they wants share with their home organizations while allowing the host organization to limit the access of liaisons officers to their system in a fine-grained manner.

***Keywords*-disaster management, information flow, access control, liaison officer**

I. Introduction

Liaison officers play an important role in the communication between large organizations, especially in the public sector. A liaison officer liaises between two organizations by providing a single point of contact and ensuring the efficient communication and coordination of their activities. This is done to achieve the best utilization of resources or employment of services of one organization by another. Usually, the *home organization* embeds a liaison officer in the *host organization* to provide face-to-face coordination. Liaison officers demand special requirements to the security mechanism of the IT infrastructure of the organization that acts as the host for a liaison officer.

Especially in disaster management scenarios, the fast exchange of necessary information is vital. As the liaison officer acts as a single communication point, he is responsible for handling requests from both his home organization and his host organization. Thus, for fulfilling their duties, liaison officers should be deeply embedded into the host organization and should be able to decide on their own which information needs to be exchanged with their home organization. Overall, the "honest" liaison officers should act as a member of the host organization and not, in an abstract sense, be in charge for the home organization.

This setting creates several security challenges for the implementation of IT systems supporting operational headquarters and crisis management teams, e. g., *Disaster Management Information Systems* (DMIS). On the one hand, such systems need to support various ways of communication in a flexible and ad hoc manner and, at the end, should even allow the liaison officer to declassify information. On the other hand, these systems need to protect, either based on strategic reasoning or based on data protection and privacy laws, the leakage of sensitive data. Usually, several liaison officers, from different home organizations, are working, at the same time, in an operational headquarter. As the different home organizations are subject to different security policies, the security policies for each liaison officer need to be defined separately. Thus, while analyzing the compliance of such a system to a given security policy, one has also to consider scenarios where information is leaked by interleaved actions (e. g., declassification), either intended or unintended, by liaison officers from different organizations.

In this paper, we present a novel approach, based on *role-based access control* (RBAC), for enabling the flexible and secure information sharing between organizations. In more detail, our contributions are three-fold:

1) a formal requirements analysis of collaboration scenarios based on liaison officers
2) a collaboration of RBAC-based systems supporting the fine-grained dynamic control of liaison officers,
3) the (partial) hiding of internal role hierarchies of organizations to other cooperating organizations.

The rest of the paper is structured as follows: after introducing the communication scenario in Section II, we present the derived requirements and their formalization in Section III. In Section IV, we present the foundations of a flexible access control mechanism, based on RBAC, that implements the derived requirements. Finally, we report on related work and draw conclusions in Section V.

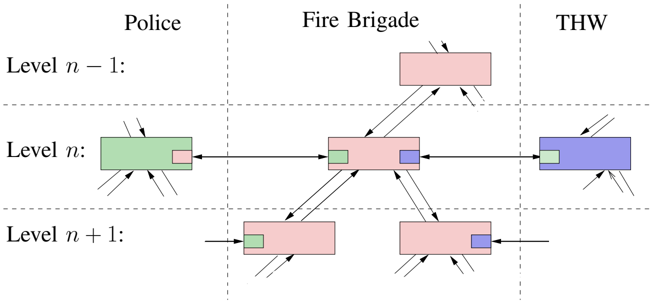

Figure 1. Communication channels across organizations

II. Collaboration of Organizations

In this section, we describe how collaborations between different organizations are organized in the presence of liaison officers. We introduce the general idea of collaborations based on liaison officers by a scenario that was developed, together with the German police organizations and the German fire brigades within the research project SoKNOS (http://www.soknos.de), a project concerned with the development of service-oriented architectures for supporting networks of public security.

A. The Role of a Liaison Officer

In our case study, we assume a natural disaster, e. g., a large flooding, whose management requires the collaboration between the police, several fire brigades, and the German Federal Agency for Technical Relief (THW). The assignment of the different squads is organized, for each organization, by a hierarchy of crisis management teams. Figure 1 illustrates the overall organization of the different operational headquarters (crises management teams).

Each crises management team is supported by a Disaster Management Information System (DMIS) providing, e. g., support for maintaining the current situation, planing, and simulation. Moreover, such systems provide communication channels both within the crisis management team and to the outside. Here, communication to the outside comprises: first communication to operational headquarters (on the proceeding and succeeding level in the hierarchy) within the same organization, second, communication to operational headquarters of different organizations (e. g., police, fire brigades, health organizations) with a state-controlled order with respect to public security, and third, any other private organization. The latter also includes the general public or the press.

The communication between the organization in the public sector is based on liaison officers. On each organizational level, two organizations that need to communicate efficiently, exchange liaison officers. A *liaison officer* is an officer that is deputed from his *home organization* into the crisis management team of the *host organization*. Usually, this is done mutually. For example, the local police delegates one of his officers to the fire brigade and vice versa, A liaison officer mainly manages the communication of the local crises management team with his own organization. As a rule of thumb, a liaison officer should be loyal to his host organization, i. e., he is obliged to ensure the secure and successful operation of the host organization, even if this can cause drawbacks for his own home organization. This is exceptionally important, as the host organization needs to trust the liaison officer to ensure the efficient communication between host and guest organization.

Moreover, within one crisis management team, several liaison officers can be working at the same time. In our example (see Figure 1) this situation occurs for the crisis management team of the fire brigades on level n. Within this team, a liaison officer from the police and one from the THW are working at the same time. With respect to security concerns, we need to keep this situation in mind, if we want also to ensure the security of the system in case of two dishonest liaison officers working together.

While our running example is based on the collaborations in the context of IT systems for disaster management, the general concept of liaison officers are used much more widely, e. g., in the collaboration between the different police organizations in Europe (e. g., EUROPOL), military organizations (e. g., NATO), or cross-national collaboration in general. Therefore, our approach can be applied directly to the various IT systems used by these organizations.

B. A Supporting IT Architecture

Within the SoKNOS project a DMIS, following the principles of a service-oriented architecture (SOA), is developed. Overall, the SoKNOS DMIS follows a decentralized approach, i. e., each crisis management team has its own, local DMIS. In particular, every DMIS has its own, locally administrated, access control system providing providing RBAC. The different DMIS systems can be connected via a common service bus. Therefore, the DMIS support the direct communication (including the sharing of services and databases) between the different crisis management teams.

Due to space reason, we restrict ourselves on a brief summary of the access control relevant features of the SoKNOS DMIS. With respect to security (i. e., authentication and authorization), the main components are:

- The *Single Sign-On and Certificate Engine* manages the user database (accounts) of the local DMIS, validates the credential of a user and issues both a long living session ticket and short living (i. e., only valid for one specific service call) service ticket. The session ticket, providing a single sign-on, is only shared between the user's client (i. e., the Portal) and the Single Sign-on and Certificate Engine. The session ticket is never revealed to a service; for service calls, the Single Sign-on and Certificate Engine issues a one-time service tickets. Moreover, the system supports proxy tickets that allows a service to act on behalf of a user, e. g., authorizing inner service calls. i. e., whenever a service needs to do further service calls that require authentication for fulfilling its purpose.

- The *Policy-Decision Engine (PDE)* manages the central access-control policy of the DMIS and provides a generic policy-decision point that is used by each component of the local DMIS. In general, we support hierarchical RBAC as means for writing access control policies. In particular, we supports break-glass concepts that can be classified into emergency levels [1].
- The *Security Plugin* provides the means for both end users and administrators for configuring the services offered by the Single Sign-On Engine and the Policy-Decision Engine. For example, administrators can manage accounts (e. g., create new users, delete users) or assign roles to users and to remove roles from users. The manager of the crisis management team can activate or deactivate emergency levels.

All security relevant information is encapsulated in an additional (encrypted) communication channel and not send through a complex middleware. This simplifies the security analysis dramatically and reduces the security related requirements that the chosen middleware needs to satisfy.

The overall architecture allows, even in a landscape with several DMIS systems connected via the SOA infrastructure, for the local management of security properties. In particular, the activation of accounts and the mapping of accounts to roles is done locally, i. e., separately for each and every DMIS. This construction ensures that each crisis management team can locally decide the access rights for the services and data the team is offering.

C. An Example: Exchanging Situational Information

In this section, we show the practicability of our approach by a small case study. Assume two operational headquartered, one from the police and one from the fire brigade, on the same level n in the management hierarchy. Both headquarters are supported by a DMIS, i. e., the police is supported by the system $\text{DMIS}_n^{\text{police}}$ and the fire brigade by system $\text{DMIS}_n^{\text{fire}}$ and they exchanged liaison officers.

In our example, we assume a flooding threatening a chemical plant which results in a potential leakage of dangerous chemicals to the outside. We are in a situation where an expert of the fire brigade simulated which area would be affected by such a leakage and visualized his findings in the situational map in $\text{DMIS}_n^{\text{fire}}$. This forecast is also of high value for the police, e. g., as a basis for deciding which roads need to be blocked. Therefore, he decides that the situational map should be shared with a group of people at the operational headquarter of the police. For establishing this collaboration, the following tasks needs to be executed:

1) The liaison officer of the police receives the inquiry of his home organization concerning a current simulation of the flooding.
2) The liaison officer of the police acknowledges the availability of a current simulation and asks who (from his home organization) should get access to it.
3) The liaison officer of the police receives, from his home organization, the request of releasing the current simulation to the police officers $p_1, \ldots, p_n$.
4) The liaison officer of the polices creates a new role r_{sim} and assigns the users $p_1, \ldots, p_n$ to this role. Depending on the trust relation between the fire brigade and the liaison officer of the police, the liaison officer may need to send a formal request for implementing this change of the security configuration to the team of the fire brigade or he may be able to implement the change himself.
5) The liaison officer of the police grants the role r_{sim} permissions for reading the current simulation.
6) The liaison officer of the police reloads the access control policy that is affected by the newly configured roles. This activates his changes to the role-hierarchy and the new permissions.
7) The liaison officer of the police send a notification to the contact person in his home organization that the users $p_1, \ldots, p_n$ can access the current simulation of the fire brigade. Of course, this notification also includes an explanation on how to access the simulation, i. e., the Uniform Resource Identifier (URI) of a Web service.

Similarly, the controlled disclosure of information to the general public can be managed. In this case, a member of the operational headquarter, i. e., the press officer, acts similar to the role of a liaison officer.

III. Requirement Analysis and its Formalization

To fulfill these needs we use a fine-grained access-control for each DMIS. The requirements can be further refined, in particular the following requirements need to be taken into account:

1) The operational headquarter that "owns" a piece of information should be the only one that can grant access to it.
2) The access control needs to be configurable at runtime, i. e., new coalitions between organizations needs to be established during a disaster. Thus, the access control configuration needs to be adaptable by end-users.
3) There are processes that require the enforcement of separation of duty principles ("four-eyes-principle"). For example, it might be the case that information can only be published to the general public after two members of operational headquarter agreed on it. It might even be the case, that two organizations need to agree before taking a specific action.
4) Liaison officers should be able to decide (to some extend) on their own, which information is shared with

their home organizations.

5) System must be able to restrict (and log) the actions of liaison officers. The degree of restrictions depends on (personal) trust.
6) Several liaison officers from different home organizations (with different levels of trust and, therefore, different rights) are, at the same time, in a single host organization. Therefore, the system must also be able to prevent information leakage against several, collaborating, liaison officers.

To fulfill these requirements, we propose to realize the access control for each individual DMIS by a by hierarchical RBAC. Every organization has its own implementation of RBAC, i. e., its own role hierarchy. We define role-based access control consisting of roles R, users U, objects O, actions A, permission P, and an ordering $\preceq$ as follows:

Definition 1 (RBAC): An RBAC is a sextuple $\mathcal{R} = (\mathsf{R}, \mathsf{U}, \mathsf{O}, \mathsf{A}, \mathsf{P}, \preceq)$, together with corresponding relations $\mathrm{UA} \subseteq \mathsf{U} \times \mathsf{R}$, $\mathrm{PA} \subseteq \mathsf{P} \times \mathsf{R}$ and functions $Ob : \mathsf{P} \to 2^{\mathsf{O}}$ and $Op : \mathsf{P} \to 2^{\mathsf{A}}$ We write $\mathrm{PA}(r) = \{p \in \mathsf{P} \mid (p, r) \in \mathrm{PA}\}$.

$\preceq \subseteq \mathsf{R} \times \mathsf{R}$ is a partial ordering on roles and defines the hierarchy of roles in $\mathcal{R}$, i. e., $r \preceq r'$ implies $\forall u \in \mathsf{U}.\ \mathrm{UA}(u, r') \to \mathrm{UA}(u, r)$ and $\forall p \in \mathsf{P}.\ \mathrm{PA}(p, r') \to \mathrm{PA}(p, r)$.

Let $\preceq$ be a relation then, as usual, $\preceq^*$ denotes its transitive closure. Furthermore, the closure $[r]_{\preceq}$ of a role r with respect to an ordering $\preceq \subseteq R \times R$ is the set $\{r' \in R \mid r' \preceq^* r\}$ of all roles subsumed by r. Given a user $u \in \mathsf{U}$ then $\mathsf{R}_u = \{r \in \mathsf{R} \mid (u, r) \in \mathrm{UA}\}$ specifies the set of roles of u in a given $\mathcal{R}$.

An RBAC satisfies a static separation if and only if an action has to be signed off by two different roles and there exists no subject that is participating in both roles. This can be generalized to an arbitrary number n of roles and formalized:

Definition 2 (static separation of duty): An RBAC $\mathcal{R}$ *satisfies* a static separation of duty $\mathsf{SSD} \subseteq 2^{\mathsf{R}} \times N$ iff $\forall (R, n) \in \mathsf{SSD}.\ \forall u \in \mathsf{U}.\ |\{r \mid \mathrm{UA}(u, r) \wedge r \in R\}| < n$.

An access control solution for DMIS needs to support the fine-grained access control for both communications within an operational headquarter (e. g., preventing information leakage by two or more collaborating liaison officers from different organizations) and for the communications to the outside. We need to support the dynamic establishment of new coalitions. Liaison officers establish a two way communication: first, a liaison officer sends messages or requests to his home organization and, second, a liaison officer sends messages or requests to his host organization.

IV. Security Mechanisms

In the following we introduce the notion of an *interface* for an RBAC representing the entrance a liaison officer offers his home organization for the host organization. It exports a subset of roles and the ordering within these roles to the outside. For example, each rescue organization has a person in the control post that is responsible for the food supply of its staff in the field. So one of the interface roles may be dedicated to issues of food supply.

The interface comes along with its own set of roles and set of users. The interface roles only exist in the interface and are mapped (dynamically) to roles of the hosting system. The idea is that the liaison officer is able to adjust the interface dynamically by changing the mapping between an interface role and the attached roles in the host organization. Interface users are kept separate from standard users of the hosting system. This allows us to restrain interface users from using their interface role in a hosting DMIS to hop off to another DMIS, which would result an unwanted transitive trust relationship between different organizations.

A security policy of the hosting organization regulates the abilities of the liaison officer, i. e., which roles in the hosting organization can be assigned to an interface role. In particular liaison officers are not allowed to map interface roles for one guest organization to interface roles of another because otherwise this could result in a potentially malicious collaboration of liaison officers.

A. Interfaces

We start with the formal definition of *interfaces* for the DMIS, which are tailored to the individual liaison officers of the host organization. Each interface is controlled by a liaison officer who has restricted rights to adjust the roles that a member of his home organizations may acquire in the hosting DMIS. For this reason the host exposes an interface consisting of (a hierarchy of) roles and a set of guest users which is disjunct from regular users of the host system. Formally we define:

Definition 3: Let $\mathcal{R}$ be an RBAC. An *interface* I for $\mathcal{R}$ is a tuple $(\mathsf{R}_I, \mathsf{U}_I, \preceq_I, \mathrm{UA}_I)$ such that $\preceq_I \subseteq \mathsf{R}_{\mathcal{R}} \times \mathsf{R}_I$, $\mathsf{R}_I \cap \mathsf{R}_{\mathcal{R}} = \emptyset$, $\mathsf{U}_I \cap \mathsf{U}_{\mathcal{R}} = \emptyset$, and $\mathrm{UA}_I \subseteq \mathsf{U}_I \times \mathsf{R}_I$.

The interface for a guest organization is attached to the host system in the obvious way. We combine interface and system roles as well as interface- and system-users. Overall, we have multiple interfaces for different organizations.

Definition 4: Let $\mathcal{R}$ be an RBAC and $\mathcal{I} = \{I_1, \ldots, I_n\}$ be a set of interfaces for $\mathcal{R}$ with $I_j = (\mathsf{R}_{I_j}, \mathsf{U}_{I_j}, \preceq_{I_j}, \mathrm{UA}_{I_j})$. The *extension* $\mathcal{R}_{\mathcal{I}}$ of $\mathcal{R}$ by $\mathcal{I}$ is the RBAC $(\mathsf{R}_{\mathcal{R}_{\mathcal{I}}}, \mathsf{U}_{\mathcal{R}_{\mathcal{I}}}, \mathsf{O}_{\mathcal{R}_{\mathcal{I}}}, \mathsf{A}_{\mathcal{R}_{\mathcal{I}}}, \mathsf{P}_{\mathcal{R}_{\mathcal{I}}}, \preceq_{\mathcal{R}_{\mathcal{I}}})$ with $\mathsf{O}_{\mathcal{R}_{\mathcal{I}}} = \mathsf{O}_{\mathcal{R}}$, $\mathsf{A}_{\mathcal{R}_{\mathcal{I}}} = \mathsf{A}_{\mathcal{R}}$, $\mathsf{P}_{\mathcal{R}_{\mathcal{I}}} = \mathsf{P}_{\mathcal{R}}$, $\mathrm{PA}_{\mathcal{R}_{\mathcal{I}}} = \mathrm{PA}_{\mathcal{R}}$, $Ob_{\mathcal{R}_{\mathcal{I}}} = Ob_{\mathcal{R}}$, $Op_{\mathcal{R}_{\mathcal{I}}} = Op_{\mathcal{R}}$ and

- $\mathsf{R}_{\mathcal{R}_{\mathcal{I}}} = \mathsf{R}_{\mathcal{R}} \cup \bigcup_{1 \leq j \leq n} \mathsf{R}_{I_j}$,
- $\mathsf{U}_{\mathcal{R}_{\mathcal{I}}} = \mathsf{U}_{\mathcal{R}} \cup \bigcup_{1 \leq j \leq n} \mathsf{U}_{I_j}$,
- $\preceq_{\mathcal{R}_{\mathcal{I}}} = (\preceq_{\mathcal{R}} \cup \bigcup_{1 \leq j \leq n} \preceq_{I_j})^*$.
- $\mathrm{UA}_{\mathcal{R}_{\mathcal{I}}} = \{(u, r) \in \mathsf{U}_{\mathcal{R}_{\mathcal{I}}} \times \mathsf{R}_{\mathcal{R}_{\mathcal{I}}} \mid \exists r' \in \mathsf{R}_{\mathcal{R}_{\mathcal{I}}}.\ (u, r') \in \mathrm{UA}_{\mathcal{R}} \cup \bigcup_{1 \leq j \leq n} \mathrm{UA}_{I_j} \wedge r \preceq_{\mathcal{R}_{\mathcal{I}}} r'\}$.

Extensions of the hosting system have to satisfy some restrictions to keep the ordinary roles of the system disjoint from roles of the interface (and analogously the set of ordinary users disjoint from the set of interface users). Furthermore, to avoid interferences between liaison officers we demand that two interfaces do not share any common interface role or interface user.

Definition 5: A set of interfaces $\mathcal{I} = \{I_1, \dots, I_n\}$ is *admissible* for an RBAC $\mathcal{R}$ iff

- $\forall j \in 1, \dots, n.\ \mathsf{R}_{\mathcal{R}} \cap \mathsf{R}_{I_j} = \emptyset \wedge \mathsf{U}_{\mathcal{R}} \cap \mathsf{U}_{I_j} = \emptyset$, and
- $\forall i, j \in 1, \dots, n.\ \mathsf{R}_{I_i} \cap \mathsf{R}_{I_j} \neq \emptyset \vee \mathsf{U}_{I_i} \cap \mathsf{U}_{I_j} \neq \emptyset \implies i = j$,

The $\mathcal{R}$-roles R_u of a user $u \in \mathsf{U}_{I_j}$ for some interface $I_j \in \mathcal{I}$ is defined by $\mathsf{R}_u = \{r \in \mathsf{R}_{\mathcal{R}} \mid r \preceq_{\mathcal{R}_{\mathcal{I}}} r' \wedge (u, r') \in \mathrm{UA}_{I_j}\}$. The following lemma guarantees that adding interfaces to an RBAC does not affect the roles of original users nor do different interface interfere with each other.

Lemma 1 (Noninterference): Let $\mathcal{R}$ be an RBAC and $\mathcal{I}$, $\mathcal{I}'$ be two sets of interfaces both admissible for $\mathcal{R}$, then

- $\forall u \in \mathsf{U}_{\mathcal{R}}\ \forall r \in \mathsf{R}_{\mathcal{R}_{\mathcal{I}}}.\ (u, r) \in \mathrm{UA}_{\mathcal{R}}$ iff $(u, r) \in \mathrm{UA}_{\mathcal{R}_{\mathcal{I}}}$
- If $I \in \mathcal{I} \cap \mathcal{I}'$ then $\forall u \in \mathsf{U}_I\ \forall r \in \mathsf{R}_{\mathcal{R}_{\mathcal{I}}} \cup \mathsf{R}_{\mathcal{R}_{\mathcal{I}'}}.\ (u, r) \in \mathrm{UA}_{\mathcal{R}_{\mathcal{I}}}$ iff $(u, r) \in \mathrm{UA}_{\mathcal{R}_{\mathcal{I}'}}$.

Proof: The first clause of the noninterference lemma is an easy consequence of the fact that all $r \in \mathsf{R}_{I_j}$ are maximal with respect to. $\preceq_{\mathcal{R}_{\mathcal{I}}}$. Therefore they cannot be inherited by a role $r' \in \mathsf{R}_{\mathcal{R}}$. An analogous argument holds for the second clause since roles of one interface cannot be inherited by roles of other interfaces. Since different interfaces do not share common users, the roles of an interface user is independent of the definition of any other interfaces. ■

B. Static Separation of Duty

Users of the interfaces have to adhere to the static separation of duty constraints in the hosting DMIS. Interfaces do not have any static separation of duty constraints of their own but inherit the constraints of the hosting system. This is covered in the following definition.

Definition 6: An interface I for $\mathcal{R}$ *respects* a static separation of duty SSD of $\mathcal{R}$ iff $\forall u \in U_I.\ \forall (\mathsf{R}, n) \in \mathsf{SSD}.\ |\mathsf{R}_u \cap \mathsf{R}| \leq n$.

Since individual interfaces are encapsulated and do not interfere with each other, it is easy to prove the following lemma:

Lemma 2: Let $\mathcal{R}$ be an RBAC satisfying a static separation of duty SSD. An extension $\mathcal{R}_{\mathcal{I}}$ of $\mathcal{R}$ by a set $\mathcal{I} = \{I_1, \dots, I_n\}$ of interfaces satisfies SSD if I_j respects SSD for all $j \in 1, \dots, n$.

Proof: Consider an arbitrary user $u \in \mathsf{U}_{\mathcal{R}_{\mathcal{I}}}$ and some $(\mathsf{R}, n) \in \mathsf{SSD}$. Suppose $u \in \mathsf{U}_{\mathcal{R}}$ then we know from the first clause of Lemma 1 that u has the same roles in $\mathcal{R}_{\mathcal{I}}$ as in $\mathcal{R}$ and therefore the constraint (R, n) is satisfied for u also in $\mathcal{R}_{\mathcal{I}}$. Now suppose, $u \in \mathsf{U}_{I_j}$ is an interface user. According to the second clause of Lemma 1 u has the same roles in $\mathcal{R}_{\mathcal{I}}$ as in $\mathcal{R}_{\{I_j\}}$. But since I_j preserves SSD we know that $|\mathsf{R}_u \cap \mathsf{R}| \leq n$ and therefore $\mathcal{R}_{\{I_j\}}$ satisfies SSD. ■

The Lemma 2 provides the sufficient constraints under which an extended RBAC will satisfy SSD constraints of the original RBAC. However, these constraints refer to roles and their hierarchy in $\mathcal{R}$. Technically speaking, this means that to guarantee these constraints one has to know about the roles and their hierarchy in the host organization. There we define

Definition 7: Let $\mathcal{R}_I$ be an extension of an RBAC $\mathcal{R}$ and $I \in \mathcal{I}$. Then $rg_I : \mathsf{R}_I \to 2^{\mathsf{R}_{\mathcal{R}}}$ is given by $rg(r) = \{r' \in \mathsf{R}_{\mathcal{R}} \mid r' \preceq_{\mathcal{R}_I} r\}$ for all $r \in \mathsf{R}_I$. Let $\mathsf{R} \subseteq \mathsf{R}_I$ then $rg(\mathsf{R}) = \bigcup_{r \in \mathsf{R}} rg(r)$.

If we suppose that a home organization should not learn about this information then it will be unable to check whether a collection of interface roles might violate the static separation of duty constraints of the host organization. There are two different general ways in which the interface could violate the constraints.

First, the liaison officer might map an individual interface role to competing roles in the hosting system. In this case no user can acquire the role because it would instantly violate the SSD constraints. Since this situation would render the role useless, we restrict the mappings of interface roles in the following way.

Definition 8: Let $\mathcal{R}_{\mathcal{I}}$ be an extension of an RBAC $\mathcal{R}$ and SSD be a static separation of duty for $\mathcal{R}$. An interface $I \in \mathcal{I}$ *complies* to SSD iff $\forall r \in \mathsf{R}_I.\ \forall (\mathsf{R}, n) \in \mathsf{SSD}.\ |\mathsf{R} \cap rg_I(r)\}| < n$.

Second, a user of the home organization acquires different interface roles and due to the mapping of the interface roles would gain competing roles in the host organization. While the host can check whether an accumulation of interface roles will violate SSD constraints, the home organization cannot since it has no access to the internal roles and hierarchies of the host. Hence the next step is to provide SSD constraints for the interfaces defined in terms of interface roles that ensure the satisfiability of the SSD constraints in the host organization. This step requires some preliminary work that is covered by the following paragraphs.

Definition 9: Let $\mathcal{R}_{\mathcal{I}}$ be an extension of an RBAC $\mathcal{R}$ with $\mathsf{SSD} = \bigcup_{i \in 1, \dots, k} (\mathsf{R}_i, n_i)$ and $\mathsf{R}_i \subseteq \mathsf{R}_{\mathcal{R}}$. Its *extensional projection* $\mathsf{Ext}_I(\mathsf{SSD})$ of SSD to an interface $I \in \mathcal{I}$ is defined by $\mathsf{Ext}_I(\mathsf{SSD}) = \{\mathsf{R} \subseteq \mathsf{R}_I \mid \forall i \in 1, \dots, k.\ |\mathsf{R}_i \cap rg(\mathsf{R})| < n_i\}$.

Assuming that the set R_I of roles in an interface I of $\mathcal{R}$ is finite, we can trivially calculate $\mathsf{Ext}_{\mathsf{R}_I}(\mathsf{SSD})$ which consists of all sets of interface roles that are compatible to each other in the sense that an interface user can acquire all roles of such a set without violating the static separation of duty constraints of the hosting system. Knowing the extensional projection to an interface I, we are also able to formulate static separation of duty constraints in terms of interface roles R_I.

Definition 10: Let Ext be a set of sets of roles closed under subset relation (i. e., if $\mathsf{R}' \in \mathsf{Ext}$ and $\mathsf{R}'' \preceq \mathsf{R}'$ then $\mathsf{R}'' \in \mathsf{Ext}$) and R be a set of roles. Then the *maximal overlap* $mo(\mathsf{Ext}, \mathsf{R})$ of Ext and R is the maximum of $\{|\mathsf{R}' \cap \mathsf{R}| \mid \mathsf{R}' \in \mathsf{Ext}\}$.

$mo(\mathsf{Ext}, \mathsf{R})$ specifies the maximal number of roles in R that can occur in some element of Ext. Hence, to specify the specific constraints SSD_I for the interface I we enumerate all subsets $\mathsf{R} \subseteq \mathsf{R}_I$ and compute $mo(\mathsf{Ext}, \mathsf{R})$:

Definition 11: Let $\mathcal{R}_\mathcal{I}$ be an extension of an RBAC $\mathcal{R}$. Then the interface constraints SSD_I of SSD and $I \in \mathcal{I}$ is defined by $\mathsf{SSD}_I = \{(\mathsf{R}, mo(Ext_{\mathsf{R}_I}(\mathsf{SSD}), \mathsf{R}) + 1) \mid \mathsf{R} \subseteq \mathsf{R}_I\}$. An interface I satisfies SSD_I iff for all $u \in U_I$ and all $(\mathsf{R}, n) \in \mathsf{SSD}_I$. $|\mathsf{R}_u \cap \mathsf{R}| < n$.

Lemma 3: Let $\mathcal{R}_\mathcal{I}$ be an extension of an RBAC $\mathcal{R}$ and let $\mathcal{R}$ satisfying a SSD. Then, $\mathcal{R}_I$ satisfies SSD iff each interface $I \in \mathcal{I}$ satisfies the interface constraints SSD_I of SSD and I.

Proof: Suppose, there is an interface user $u \in U_I$ violating SSD_I. Thus, there is an $(\mathsf{R}, n) \in \mathsf{SSD}_I$ and $|\mathsf{R}_u \cap \mathsf{R}| \geq n = mo(\mathsf{Ext}_{\mathsf{R}_I}(\mathsf{SSD}), \mathsf{R}) + 1$. This means that $|\mathsf{R}_u \cap \mathsf{R}| > mo(\mathsf{Ext}_{\mathsf{R}_I}(\mathsf{SSD}), \mathsf{R})$ and therefore $\mathsf{R}_u \cap \mathsf{R} \not\subseteq \mathsf{Ext}_{\mathsf{R}_I}(\mathsf{SSD})$. Thus, $rg(\mathsf{R}_u)$ violates SSD.

Now suppose, there is a user $u \in \mathcal{R}_I$ violating SSD. Obviously, u has to be a user of some interface I because $\mathcal{R}$ satisfies SSD and Lemma 2 holds. Hence, $\exists(\mathsf{R}, n) \in \mathsf{SSD}$. $|\mathsf{R} \cap rg(\mathsf{R}_u)| \geq n$ and therefore $\mathsf{R}_u \notin \mathsf{Ext}_I(\mathsf{SSD})$. With u as counter example SSD_I is not satisfied. ■

Lemma 3 allows us to interpret interfaces as abstract views on a host RBAC. It exports the necessary information about the separation of duty constraints to the outside without disclosing the internal structure of the host organization.

A home organization makes use of the interfaces of a host organization by identifying some of its roles and users with roles and users of the interface.

Definition 12: A *guest access* to an interface $I \in \mathcal{I}$ of $\mathcal{R}_\mathcal{I}$ by an (extended) RBAC $\mathcal{R}'_{\mathcal{I}'}$ is a pair (ϕ, ψ) of partial mappings with $\phi : \mathsf{U}_{\mathcal{R}'} \to \mathsf{U}_I$ and $\psi : \mathsf{R}_{\mathcal{R}'} \to \mathsf{R}_I$.

An guest access (ϕ, ψ) to an interface I *satisfies* the static separation of duty constraints SSD_I of I iff for all users u in the domain of ψ holds that the set $\{\psi(r) \mid (u, r) \in \mathrm{UA}_{\mathcal{R}'}\}$ satisfies the SSD constraints of I.

Notice that only "standard" users of the home organization (i. e., $u \in \mathsf{U}_{\mathcal{R}'}$) but not users of its interfaces $\mathcal{I}'$ are allowed to access some interface of $\mathcal{R}_I$. This prevents guests from hopping along a chain of interfaces to third party organizations, which would result in an uncontrolled transitive information flow between different organizations. It also prevents guests from getting back to their home organization along some paths of third-party organizations, which would raise the problem of keeping all the role hierarchies of these organizations including the mappings of guest accesses in a consistent state.

C. Dynamic Aspects

As mentioned before, each interface of a host organization corresponds to a particular home organization. The interface is controlled by a liaison officer who is a member of the home organization but who is physically situated in the crisis team of the host organization. The liaison officer has to resolve the conflicting interests of both home and host organization. He must determine an appropriate fragment of information provided by the host organization that is suitable to his home organization. On the one hand the disclosure of information must not violate any legal regulations and on the other hand his home organization needs appropriate information to plan and decide on further steps. In practice the degree to which a liaison officer gets access to information in the host organization depends on the trust in the individual (human) person.

To support the work flow of a liaison officer by a DMIS the liaison officer has to be authorized to dynamically modify the mapping between interface roles and roles of the hosting system, which in general causes also a change of the role hierarchy and the corresponding static separation of duty constraints in the interface. Encoding different levels or types of confidentiality into (a hierarchy of) different roles, the liaison officer is able to make additional information accessible to this home organization by adding corresponding roles of the host system to the range of the mapping of the interface roles.

The liaison officer is a user of the host system and maintains a subset of roles in the host system that he can assign to interface roles. The set of roles he can maintain is in general different from the set of roles he can acquire personally (otherwise he would be not able to assign competing roles to different interface users) and are disjoint from roles of any other interface.

A *liaison officer* LO_I of an extended RBAC $\mathcal{R}_I$ for an interface $I \in \mathcal{I}$ is a user $LO_I \in \mathsf{U}_\mathcal{R}$ maintaining a set $\mathsf{R}_{LO_I} \subseteq \mathsf{R}_\mathcal{R}$ of roles. He is authorized to alter a mapping $\preceq_I$ to some $\preceq'_I$ in the interface I iff $\forall(r, r') \in (\preceq_I \setminus \preceq'_I) \cup (\preceq'_I \setminus \preceq_I)$. $r \in \mathsf{R}_{LO_I}$ holds.

Thus the liaison officer is only allowed to change the mapping $\preceq_I$ of the interface as long as only roles in R_{LO_I} are affected. Notice that this restriction does not only prevent the liaison officer from unfaithfully equipping an interface role with too many permissions but it also guarantees that specific roles cannot be withdrawn by a liaison officer. This applies in particular to roles corresponding to administrative cooperations, which are regulated by national laws.

The question arises as to how the change of $\preceq_I$ is done in practice. Suppose, a guest user has acquired a role r' in the hosting system via some interface role r. He uses this role r' to execute a service on the host system. If the liaison officer withdraws r' from the mapping of the interface role r, then the guest user looses the right to execute the service. As a result, this service is stopped immediately and the user has to authorize itself again (this time gaining the changed roles). This procedure simplifies access control in contrast to a procedure in which a user would have persistent roles till the end of the corresponding session. In such a case we would have to maintain a history of different RBACs corresponding to the security policies of the points in time in which individual sessions have been created.

D. Multiple Hosting Systems

Up to now we considered the case of one DMIS hosting various home organizations. In practice, we are faced with the situation that there are multiple hosts, like for instance, the police and the fire brigade and that there is a liaison officer of the police situated inside the control center of the fire brigade as well as a liaison officer of the fire brigade located in the control center of the police.

Suppose, there are two DMIS with corresponding extended RBACs $\mathcal{R}_{\mathcal{I}}$ and $\mathcal{R}'_{\mathcal{I}'}$. Within $\mathcal{I}$ there is an interface I provided for $\mathcal{R}'_{\mathcal{I}'}$ and vice versa $I' \in \mathcal{I}'$ denotes an interface for $\mathcal{R}_{\mathcal{I}}$. Then in theory, there is the problem that a user in $\mathcal{R}_{\mathcal{I}}$ might acquire a role in $\mathcal{R}'_{\mathcal{I}'}$ via the interface I' which allows him to make use of the interface I to hop back to its own system $\mathcal{R}_{\mathcal{I}}$ but now acquiring a role that is superior to his original role. This problem is known as the *elevation of rights problem.*

We counteract this problem by the simple restriction that interface users of $I \in \mathcal{I}$ in a system $\mathcal{R}_{\mathcal{I}}$ are not allowed to make use of any interface I' provided by a system $\mathcal{R}'_{\mathcal{I}'}$ to $\mathcal{R}_{\mathcal{I}}$. Technically, we enforce this restriction by restricting the accesses (ϕ, ψ) to the interface I'. We simply restrict the domain of ϕ to users of $\mathcal{R}$ and thus preventing interface users in $\mathcal{R}_{\mathcal{I}}$ to make use of the access. Roughly speaking, this restriction corresponds to some kind of intransitive flow policy between different organizations.

Liaison officers can only change the interface for their own home organization. In Lemma 1 we proved that the change of one interface does not affect the roles or users of the original RBAC or any other interface. This prevents potential malicious collaborations between various liaison officers. Also there is no interference possible between liaison officers located in different organizations since we do not allow guests in a host organization to access an interface of a third organization.

V. Discussion

A. Related Work

We see several lines of related work. The security challenges of dynamic coalitions are, e. g., discussed in [2]. With respect to the application scenario, the closest related work is [3]. The authors of [3] report on the collaboration between EUROPOL, the European Police Office and Eurojust, the European Judicial Cooperation which is also based on liaison officers. In fact, each of the 27 member states has an appointed Liaison Officer and a Contact Point to interact with EUROPOL and Eurojust. Moreover, they propose an extension of the XACML [4] for supporting distributed roles, i. e., dRBAC [5]. Distributed roles are known to the collaborating partners and can, locally, be mapped to system roles. We avoid the need of distributed roles by allowing the liaison officer to create new roles on the fly. Moreover, we provide for each hosted liaison officers a separate subhierarchy within the role hierarchy.

Security research on delegation, e. g., [6], [7] focuses on determining the set of rights (e. g., expressed as changes to a given policy) that need to be transferred. Such techniques can be integrated into our approach, e. g., as systems that assist the liaison officer while defining new roles for staff members of his home organization.

Moreover, there are works on merging RBAC policies of collaborating organizations [8] and on mining or mapping roles (i. e., finding similar roles) between different RBAC policies [9], [10], [11]. All these approaches have in common that they try to minimize the overall number of roles by reusing already existing roles. In contrast, our solution allows for a fine-grained configuration for each guest organizations at runtime.

While there exists a substantial body of literature [1], [12], [13], [14], [15] extending RBAC, hierarchical RBAC is sufficient to capture the requirements in the disaster management area [16]. As our approach does not rely on a specific RBAC model, it can be combined with most of these models.

B. Conclusion and Future Work

We presented an approach for supporting the flexible and secure information exchange between organizations using liaison officers. In particular, our mechanism allows the liaison officers to decide on its own authority which information he wants to exchange with is home organization while allowing the host organization to limit the access of liaisons officers to their system in a fine-grained manner.

As our approach is based on creating roles dynamically, there is the danger of creating complex role hierarchies that are difficult to maintain. One possibility to reduce this risk, could be the integration of role-mining approaches for finding similar roles and allowing liaison officers for reusing already existing roles. While this reduces the number of roles, it requires a careful analysis showing that such a system still satisfy the required security properties.

Future work includes the analysis of collaboration scenarios with liaison officer in an environment requiring data labeling, e. g., Bell-LaPadulla [17]. In this case, the fine grained de-classification of information may require, on the one hand, that data is labeled with a set of labels, and, on the other hand, for each operation an own space for security labels (similar to the own subspace of roles in our presented approach) may be required.

Acknowledgments

This work has been supported by the German "Federal Ministry of Education and Research" in the context of the project "SoKNOS." The authors are responsible for the content of this publication.

References

[1] A. D. Brucker and H. Petritsch, "Extending access control models with break-glass," in *ACM symposium on access control models and technologies (SACMAT)*, B. Carminati and J. Joshi, Eds. New York, NY, USA: ACM Press, 2009, pp. 197–206.

[2] J. Charles E. Phillips, T. Ting, and S. A. Demurjian, "Information sharing and security in dynamic coalitions," in *ACM symposium on Access control models and technologies*. New York, NY, USA: ACM Press, 2002, pp. 87–96.

[3] H. K. Lee, Heiko, and Luedemann, "Lightweight decentralized authorization model for inter-domain collaborations," in *ACM workshop on Secure web services*. New York, NY, USA: ACM Press, 2007, pp. 83–89.

[4] "eXtensible Access Control Markup Language (XACML), version 2.0," 2005.

[5] E. Freudenthal, T. Pesin, L. Port, E. Keenan, and V. Karamcheti, "dRBAC: distributed role-based access control for dynamic coalition environments," in *International Conference on Distributed Computing Systems*, 2002, pp. 411–420.

[6] D. W. Chadwick and A. Otenko, "The PERMIS X.509 role based privilege management infrastructure," in *ACM symposium on Access control models and technologies*. New York, NY, USA: ACM Press, 2002, pp. 135–140.

[7] C. Ye and Z. Wu, "Using XML and XACML to support attribute based delegation," in *International Conference on Computer and Information Technology*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 751–756.

[8] B. Shafiq, J. B. Joshi, E. Bertino, and A. Ghafoor, "Secure interoperation in a multidomain environment employing RBAC policies," *IEEE Transaction on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1557–1577, Nov. 2005.

[9] L. D. Martino, Q. Ni, D. Lin, and E. Bertino, "Multi-domain and privacy-aware role based access control in ehealth," in *International Conference on Pervasive Computing Technologies for Healthcare*, 30 2008-Feb. 1 2008, pp. 131–134.

[10] A. Kamath, R. Liscano, and A. E. Saddik, "User-credential based role mapping in multi-domain environment," in *International Conference on Privacy, Security and Trust*. New York, NY, USA: ACM Press, 2006, pp. 1–1.

[11] G. Geethakumari, A. Negi, and V. N. Sastry, "A cross – domain role mapping and authorization framework for RBAC in grid systems." *International Journal of Computer Science & Applications*, vol. VI, 2009.

[12] D. F. Ferraiolo, R. S. Sandhu, S. I. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed NIST standard for role-based access control," *ACM Transactions on Information and System Security*, vol. 4, no. 3, pp. 224–274, 2001.

[13] M. Wilikens, S. Feriti, A. Sanna, and M. Masera, "A context-related authorization and access control method based on RBAC: A case study from the health care domain," in *ACM symposium on Access control models and technologies*. New York, NY, USA: ACM Press, 2002, pp. 117–124.

[14] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[15] L. Zhang, G.-J. Ahn, and B.-T. Chu, "A role-based delegation framework for healthcare information systems," in *ACM symposium on Access control models and technologies*. New York, NY, USA: ACM Press, 2002, pp. 125–134.

[16] I. Aedo, P. Díaz, and D. Sanz, "An RBAC model-based approach to specify the access policies of web-based emergency information systems," *The International Journal of Intelligent Control and Systems*, vol. 11, no. 4, Dec. 2006.

[17] D. E. Bell and L. LaPadula, "Secure computer systems: Unified exposition and multics interpretation," MITRE, Technical Report MTR-2997, 1976.