# A Calculus for Composite Authorities' Policy Derivation in Shared Domains of Pervasive Computing Environments

Morteza Amini
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
m_amini@ce.sharif.edu

Rasool Jalili
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
jalili@sharif.edu

## Abstract

*The decentralized security management in a pervasive computing environment, requires apportioning the environment into several security domains. In each security domain, an administrator (we call it authority) is responsible for specifying the security policies of the domain. Overlapping of security domains results in the requirement of cooperative security management in the shared/ overlapping domains. To satisfy this requirement, we propose an abstract security model, as well as its supplementary calculus of composite authorities. The security model is based on deontic logic and is independent of the domains' heterogeneity. The model's policy language (we call it MASL) enables multiple authorities to specify their domain policies, including obligations and authorizations. Our proposed calculus of composite authorities, enables the security system to infer policy statements of composite authorities from the cooperating primitive authorities. The calculus offers three styles of cooperative administration including collaborative, disjunctive, and delegative administration. Abstraction and automated composite authorities' policy derivation are the main advantages of the proposed logical model.*

**Keywords:** Logical Security Model, Access Control, Deontic Logic, Pervasive Computing Environment

## 1 Introduction

In pervasive computing environments, resources like information and services are accessible anywhere and anytime via any devices. Thus, users may access resources and services remotely. There are different sorts of users and services and some of them may be unknown or not predefined [16]. The distribution of resources in these environments, forces us to employ decentralized security management. In this approach, the environment is divided into the number of domains based on different factors like geographical situation. For each domain, there is a security agent with an administrator (we call it *authority*) and each resource, like a service, can register itself in one or more domains. Thus, the authorities are responsible for preserving the security of resources that are under their protection.

The requirement of policy specification in different domains by different authorities motivated us to propose a multi-authority version of deontic logic to specify security policies including obligation policies as well as authorization ones. The unresolved problem in multiple security domains is the overlapping of domains (which results in shared domains or subdomains) and their administration issues in such a situation. The main contribution of this paper is proposing a logical system to enable cooperative administration in the overlapping or shared domains.

In management science, *cooperative management*, also called co-management, tries to achieve more effective and equitable systems of resource management. In cooperative management, representatives of user groups, the scientific community, and government agencies should share knowledge, power, and responsibility [6]. In this paper, for security purposes, three styles of cooperative administration are introduced; collaborative, disjunctive, and delegative administration. The axioms that enable us to infer security policies specified by composite authorities (which are obtained based on the cooperative administration styles), as well as the semantics of the proposed syntax, and a soundness proof of the logical system are described in the rest of this paper.

The rest of this paper is organized as follows: After introducing the relative researches in the next section, we introduce our proposed security model and its logical policy language, which is used for security policy specification by multiple authorities in section 3. In section 4, the calculus of composite authorities accompanying with three styles of administration, resulted from the different types of composi-

tion, is presented. The section 5 demonstrates our proposed architecture and process for enforcing obligation policies and access control. In section 6, we present a case study to illustrate the applicability of the proposed model to specify security policies and infer required composite authorities' policies for enforcing cooperative management. Finally, in the last section, we conclude the paper and draw some directions for future work in this research trend.

## 2 Related Work

The important characteristic of the proposed solution in this paper is its logical foundation. Using logics to tackle the problems of specifying and proving the security of distributed systems started from 1988 by Glasgow [10]. Since then, various sorts of logic have been used to model user beliefs and inference abilities, specification of security policies, context and temporal constraints, and historical interactions in distributed environments like pervasive computing or ubiquitous environments.

The access control logic proposed by Abadi, Lampson, and others in [2], provides a logical system (based on the modal logic) for specifying composite principals, access control lists, and access delegation in distributed systems. The first attempt in providing a general framework for authorization made by Woo and Lam in [25]. They proposed the use of Default logic, which is a kind of non-monotonic logic, to specify authorization policies. Undecidability of the logic proposed by Woo and Lam, encouraged Jajodia, *et al.*, [13, 12] to define an authorization specification language (ASL) based on the stratified first-order logic that is not only decidable but also linear. Barker, *et al.*, [5] took a similar approach in specification of multiple types of policies (with emphasis on RBAC policies) using stratified Horn-clauses logic. Freedom in using constrained negation in this langauge in comparison with ASL [12], encouraged the authors to leverage a partial-deduction approach for specializing access control on deductive databases in [4]. Kushik, *et al.*, [18] introduced a constraint logic programming (CLP) based framework to determine access to partial or full ontologies in order to preserve confidentiality in open World Wide Web. In this framework, policies are CLP programs (which are stratified Horn clauses with constructive negation) that prevent disclosing sensitive portions of an ontology by renaming or hiding some concepts or relationships.

We propose and use a multi-authority version of deontic logic in this paper. Using deontic logic in access control was considered by Cuppens, *et al.* [8] for information flow control. Kagal, Finin, and others proposed a policy language called Rei [17] based on Semantic Web languages (RDF and DAML+OIL) for pervasive computing environments. In Rei, concepts of deontic logic like permission, obligation, and prohibition were used. However, this policy language does not support multiple administrators' policies resolution and also cooperative administration.

In this paper we propose a kind of modal logic (multiple-authority version of deontic logic) to provide a platform for specifying security policies by multiple administrators in pervasive and distributed environments abstract from implementation details. This enables us to infer composite authorities' policies from the policies specified by primitive administrators using the logical system build upon this platform. Multiple security domains were introduced in some papers to split the environment into several administration domains. This concept is used in [16] as a security framework in a pervasive computing environment and it is used in [3] for mobile computing environments in controlling mobile users' accesses to their home and visiting (foreign) domains. Furthermore, a policy language named X-RABAC is proposed by Joshi, *et al.* [15, 21], for multi-domain environments. In these researches the security control on shared resources is realized through the specification of mediation policies [22] and enforcing them for inter domains access control [24]. The approach proposed in this paper, tries to get rid of specifying inter domain security policies through automatic inference of composite authorities' policies. A composite authority is a representative of an administrator that states security policies of the shared domains based on the policies of the participating individual domains.

## 3 Logical Security Model

In distributed and pervasive computing environments, the environment is usually separated into multiple security domains [16, 23]. Each security domain has a security manager , henceforth called *authority*. In each domain a set of resources (objects) are protected based on the domain's security policies. The duty of an authority in a domain is composing the security policies of the domain. It is worthwhile to note that domains may have got overlapped with each other. In this way, the owner or creator of each resource (object) can register with one or more domains and rely on their authorities' policies to secure his/her resource.

To state security policies by each domain's authority, we need to establish a policy language. MASL (multi-authority security policy language), which is presented in this paper, is a logical language that enables authorities to compose their security policy rules, and to infer implicit policy rules from the explicit ones. MASL is a kind of modal logic languages that is founded on the multi authorities version of deontic logic to specify *authorization* as well as *obligation* policies. Statements like it is obligatory that (denoted by `OB`), permissible that (denoted by `PE`), impermissible that (denoted by `IM`), and gratuitous that (denoted by `GR`) can be specified by this logic. Note that the paradoxes that are

mentioned for deontic logic in ethics and legal theory areas are the result of linguistic interpretation ambiguities. These problems are not considerable in our case [10].

### 3.1 Multi-Authority Security Policy Language (MASL)

To define the syntax of MASL logic and introducing the logical template of security policy rules, we need to define the alphabet and sentences (formula) of our logical language. The alphabet of MASL is as follows:

- a set of context and conditional propositions $x_0, x_1, ..., x_n$;
- a set of names (terms) $t_0, t_1, ...., t_l$;
- a set of context and conditional predicates $p_0^{i_0}, p_1^{i_1}, ..., p_m^{i_m}$ $(i_k > 0)$ in which each predicate $p_k^{i_k}$ is an $i_k$-ary relation on terms like $p_k^{i_k}(t_1, t_2, ..., t_{i_k})$;
- a definite set $AU$ of primitive authorities $u_0, u_1, ..., u_t$ and composite authorities resulted from composing the primitive ones (using the calculus of authorities, which will be introduced later);
- deontic statuses symbols including OB, PE, IM, and GR;
- the propositional primitive relaters $\wedge, \neg$ (We may use other abbreviations like $\vee$, $\rightarrow$, and $\leftrightarrow$);
- auxiliary symbols (, and ).

Using the above alphabets, the sentences of the language are formulas that are defined as follows.

**Definition 1 (Formula)** *A* formula *is defined inductively as follows:*

- *every proposition like $x_i$ is an atomic formula.*
- *if $t_1, t_2, ..., t_k$ are terms and p is a k-ary predicate, then $p(t_1, t_2, ..., t_k)$ is an atomic formula.*
- *if $\alpha_i$ and $\alpha_j$ are formulas then so are $\alpha_i \wedge \alpha_j$ and $\neg\alpha_i$ (and analogously $\alpha_i \vee \alpha_j$, $\alpha_i \rightarrow \alpha_j$, and $\alpha_i \leftrightarrow \alpha_j$)*
- *if ds is a deontic status, $u$ is an authority and $\alpha$ is a formula then $\mathtt{ds}_u\alpha$ is a formula.*

In this definition, $\mathtt{ds}_u\alpha$ intuitively means an authority $u$ declares that the status ds is established for $\alpha$. For example, $\mathtt{OB}_u\alpha$ means $u$ states that it is obligatory (necessary) to be the case $\alpha$.

Following the above language, if we suppose to have an access predicate, $\mathbf{do}(s, o, a)$ on terms of subjects (or access requesters denoted by $S$), objects (or resources denoted by $O$), and actions (denoted by $A$), security policy rules are defined as follows.

**Definition 2 (Security Policy Rule)** *A* security policy rule *(or in short* policy rule*) stated by an authority $u$ is a formula of the form:*

$$\alpha \rightarrow \mathtt{ds}_u\ \beta \text{ where } \beta = \mathbf{do}(s, o, a) \text{ or } \beta = \mathtt{ds}_v\beta$$

*where $\alpha$ is a formula, ds is a deontic status, $u$, and $v$ are authorities, $a \in A$ is an action term, $s \in S$ is a subject term, and $o \in O$ is an object term.*

Note that a security policy of the form $\mathtt{ds}_u\mathbf{do}(s, o, a)$ can be defined on the object term *o* by authority $u$, if objects described by the term *o* are registered in the security domain of authority $u$. In fact, MASL does not control administrative rights on specifying security policies and we should do the above control at the implementation level. Furthermore, as we describe later in this paper, a security policy like $\mathtt{ds}_u(\mathtt{ds}_v\mathbf{do}(s, o, a))$ is valid whenever an authority $u$ has administration delegation right of authority $v$.

There are two categories of policies in the system, *authorization policies* and *obligation policies*. The intuitive meaning of an *authorization policy* of the form $\alpha \rightarrow \mathtt{ds}_u\mathbf{do}(s, o, a)$ is as whenever the formula $\alpha$ is satisfied, a subject $s$ (a human user, an agent, or a service) is permitted to or forbidden to do the action $a$ on the object $o$. The intuitive meaning of an *obligation policy* is a subject $s$ is obliged to or not to do the action $a$ on the object $o$.

In specifying security policy rules by authorities using the above language, we have this assumption that each authority is not allowed to have inconsistent policy rules. The set of policy rules specified by authorities and the statements that are describing the context, construct the knowledge base (denoted by KB) in our model.

### 3.2 Proof Theory of MASL

The axioms (except the axioms of the composition of authorities) and inference rules of MASL are as follows.

**A1.** If $p$ is a tautology of propositional logic, then $\vdash p$ (TAUT)

**A2.** $\vdash \mathtt{OB}_u(p \rightarrow q) \rightarrow (\mathtt{OB}_u p \rightarrow \mathtt{OB}_u q)$ (OB-MK)

**A3.** $\vdash \mathtt{OB}_u p \rightarrow \neg\mathtt{OB}_u\neg p$ (OB-MD)

**A4.** $\vdash \mathtt{PE}_u p \leftrightarrow \neg\mathtt{OB}_u\neg p$ (PE-Def)

**A5.** $\vdash \mathtt{IM}_u p \leftrightarrow \mathtt{OB}_u\neg p$ (IM-Def)

**A6.** $\vdash \mathtt{GR}_u p \leftrightarrow \neg\mathtt{OB}_u p$ (GR-Def)

**R1.** If $\vdash p$ and $\vdash p \rightarrow q$, then $\vdash q$ (MP)

**R2.** If $\vdash p$ then $\vdash \mathtt{OB}_u p$ (OB-MO)

Axiom A1 (TAUT) subsumes propositional logic with in our logic. Axiom A2 (OB-MK) presents that if an authority like $u$ states that a condition is obligatory, and its antecedent is obligatory from $u$'s view point, then the consequence is obligatory from $u$'s view point as well. Axiom A3 (OB-MD) tells if $u$ states that $p$ is obligatory, then its negation is not obligatory from $u$'s view point. The axioms A4 to A6 define the deontic statuses PE, IM, and GR, based on the OB (obligation) status. We can easily take one of these four statuses as a primitive and define others. Inference rule R1 (MP) is Moduse-Ponens rule. Inference rule R2 (OB-MO) states that if something is a theorem, then its obligation from any other authority's point of view is also a theorem.

### 3.3 Semantics of MASL

To present the semantics of the proposed logic, we use a Kripke structure in which domain of objects and interpretation of names and predicates are included. Note that in this paper, quantifiers (like universal and existential ones) are from meta language and they do not belong to the MASL language.

A Kripke-style structure of the presented logic is a 5-tuple like $\mathcal{M} = \langle W, R, \Phi, \Delta, I \rangle$, where:

- $W$ is a non-empty set of possible worlds. Each world is a global authorized state in the environment.
- $R$ is an interpretation function that maps each *authority* to a binary relation on $W$. This mapping function for primitive authorities is defined as $r = AU \rightarrow \mathcal{P}(W \times W)$ and for composite authorities is defined as an extension on $r$ which is presented in section 4. Each relation $r(u)$ must be reflexive and serial, i.e., for all $u \in AU : \forall w \in W : w \in r(u)(w)$ [reflexiveness] and $\exists w' \in W, w' \in r(u)(w)$ [seriality]. However, reflexivity is enough, because, it redounds to seriality. Extension $R$ of function $r$ preserves these two properties.
- $\Phi$ is an interpretation function that maps each *formula* to a subset of possible worlds in which the formula is correct. This function for atomic formulas is defined as $\phi = Atomic\ Formulas \rightarrow \mathcal{P}(W)$ and for complicated formulas is defined inductively as follows:

$$\Phi(p) = \phi(p), \text{ if } p \text{ is an atomic formula}$$
$$\Phi(\neg\alpha) = W - \Phi(\alpha)$$
$$\Phi(\alpha \wedge \alpha') = \Phi(\alpha) \cap \Phi(\alpha')$$
$$\Phi(\mathtt{OB}_u\alpha) = \{w | R(u)(w) \subseteq \Phi(\alpha)\}$$
$$\Phi(\mathtt{IM}_u\alpha) = \{w | R(u)(w) \subseteq \Phi(\neg\alpha)\}$$
$$\Phi(\mathtt{PE}_u\alpha) = \{w | R(u)(w) \cap \Phi(\alpha) \neq \emptyset\}$$
$$\Phi(\mathtt{GR}_u\alpha) = \{w | R(u)(w) \cap \Phi(\neg\alpha) \neq \emptyset\}$$

- $\Delta$ is a non-empty set of objects in all worlds of $W$. We assume that all possible worlds have a shared domain of objects.
- $I$ is an interpretation function that in each world $w$ assigns to every *name (term)* $t$, a set of objects in $\Delta$, i.e., $I(w)(t) = t_w^I \subseteq \Delta$, to every predicate $p(t_1, ..., t_n)$, an n-ary relation on $\Delta^n$, i.e., $I(w)(p) = p_w^I \subseteq \Delta^n$ which is a set of n-tuples ${t_1}_w^I \times ... \times {t_n}_w^I$.

  This function must satisfy this limitation that the interpretation of a name (term) must be identical in all worlds, i.e., for all $w, w' \in W : I(w)(t) = I(w')(t)$.

The function $\Phi$ must satisfy this constraint that if $p$ is a predicate, $w \in \Phi(p)$ if and only if there exist a set of n-tuples in the world $w$ that the function $I$ assigns them to the predicate $p$.

**Definition 3 (Truth)** *A formula $\alpha$ in a model $\mathcal{M} = \langle W, R, \Phi, \Delta, I \rangle$ at a world $w \in W$ is true, denoted $\models_w^{\mathcal{M}} \alpha$, if and only if $w \in \Phi(\alpha)$. Analogously, $\alpha$ at a world $w$ is not true, denoted $\not\models_w^{\mathcal{M}} \alpha$ iff $w \notin \Phi(\alpha)$.*

Regarding to the above semantics and definition of truth, we can prove that the presented axiomatic system is sound, i.e., if $\vdash \alpha$ then $\models \alpha$. The soundness proof can be obtained by proving that the axioms are sound and inference rules preserve soundness. Due to space limitation we do not present the soundness proof of the proposed logic in this paper.

## 4 Calculus of Cooperative Administration

To enable authorities to enact, manage and enforce security policies cooperatively in their shared security domain, three styles of cooperative administration are introduced in this section including collaborative, disjunctive, and delegative administration. Before getting through the details of these three cooperative administration styles, we introduce a virtual authority, i.e., *composite authority*, for the domain which is managed by cooperative approach.

A composite authority is an authority which is obtained by composing of primitive authorities based on the one of the three aforementioned cooperative administration styles. Each shared domain has a composite authority (instead of a primitive authority) who is a representative of primitive authorities of the participating domains. The overall syntax of composite authority is defined as follows.

**Definition 4 (Composite Authority)** *By having a set of primitive authorities, and $\&, \triangleright, |$ notions, and parenthesis as alphabets, a composite authority is defined inductively as follows:*

- *each primitive authority like $u$ is a composite authority.*
- *if $u_i$ and $u_j$ are composite authorities, then so are $u_i\&u_j$, $u_i \triangleright u_j$, and $u_i|u_j$.*
- *if $u$ is a composite authority, then $(u)$ is a composite authority as well.*

Adding composite authorities to the proposed security policy language, i.e., MASL, and promoting it with new axioms resulted from adding the concept of composite authorities, enables us to infer security policies specified by composite authorities on shared domains. Automation of policy derivation for shared domains based on the security policies of the participating individual domains is the main advantage of such a logic-based cooperative administration framework. In the rest of this section axioms, semantics and soundness proofs of proposed axiomatic system are presented separately for each cooperative administration style.

### 4.1 Collaborative Administration

There exist some situations in which two authorities require to collaborate in making an access decision. For example whenever two authorities have overlapped domains may need to establish security policy on the shared under protection objects in concert.

In order to support this kind of composite authority we add $u_i\&u_j$ notion to our calculus of authorities.

**Axioms:**

- $\vdash \mathtt{OB}_{u_i\&u_j}\alpha \leftrightarrow \mathtt{OB}_{u_i}\alpha \wedge \mathtt{OB}_{u_j}\alpha$ (CAU)
- $\vdash \mathtt{OB}_{u_i\&u_i}\alpha \leftrightarrow \mathtt{OB}_{u_i}\alpha$ (CAI)
- $\vdash \mathtt{OB}_{u_i\&u_j}\alpha \leftrightarrow \mathtt{OB}_{u_j\&u_i}\alpha$ (CAC)
- $\vdash \mathtt{OB}_{u_i\&(u_j\&u_k)}\alpha \leftrightarrow \mathtt{OB}_{(u_i\&u_j)\&u_k}\alpha$ (CAA)

Axiom CAU demonstrates that whenever two authorities make a statement collaboratively, means both of them have the same statement separately and they agree upon. The CAI, CAC, and CAA axioms show that collaborative operator (&) is idempotent, commutative, and associative over authorities.

**Semantics:** The extension of interpretation function $R$ in the proposed Kripke model for $\&$ notion is as follows: $R(u_i\&u_j) = R(u_i) \cup R(u_j)$

We may expect intersection instead of union in the interpretation of collaboration notion (&). Whereas, union of relations associated to the participating authorities results in decreasing the norms which are available in possible worlds from the resulted composite authority. In other worlds, if $u_i$ states in the world $w$ that $\alpha$ is obligatory, and $u_j$ states the inverse one (like $\mathtt{IM}_{u_j}\alpha$) or something else (like $\mathtt{GR}_{u_j}\alpha$), in collaborative combination of $u_i$ and $u_j$ we cannot infer the obligation of $\alpha$ in the world $w$ anymore. It is clear that in all the worlds resulted from the union of the possible worlds of $w$ from $u_i\&u_j$ point of view, $\alpha$ is not true, and hence $\mathtt{OB}_{u_i\&u_j}\alpha$ is not true. Figure 1 shows more precisely.

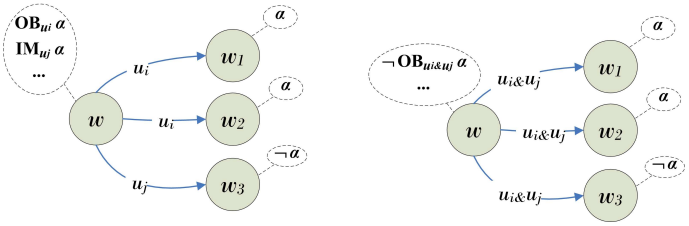


**Figure 1. Semantics of collaborative composition of authorities**

**Soundness:** The soundness of CAI, CAC, and CAA are easily proven by the idempotency, commutativity, and associativity of union operation on relation $R(u)$. The soundness proof of axiom CAU is presented in proposition 5.

**Proposition 5 (Soundness of CAU)** *Axiom CAU is sound.*

***Proof*** Suppose for some model $\mathcal{M} = \langle W, R, \phi, \Delta, I\rangle$ and some $w \in W$, we have $\models^{\mathcal{M}}_{w} \mathtt{OB}_{u_i\&u_j}\alpha$. Thus, $w \in \Phi(\mathtt{OB}_{u_i\&u_j})$ if and only if $w \in \Phi(\mathtt{OB}_{u_i}\alpha \wedge \mathtt{OB}_{u_j}\alpha)$, because:

$$\begin{aligned}
\Phi(\mathtt{OB}_{u_i\&u_j})\alpha &= \{w|R(u_i\&u_j)(w) \subseteq \Phi(\alpha)\}\\
&= \{w|(R(u_i) \cup R(u_j))(w) \subseteq \Phi(\alpha)\}\\
&= \{w|R(u_i)(w) \cup R(u_j)(w) \subseteq \Phi(\alpha)\}\\
&= \{w|R(u_i)(w) \subseteq \Phi(\alpha) \wedge R(u_j)(w) \subseteq \Phi(\alpha)\}\\
&= \{w|R(u_i)(w) \subseteq \Phi(\alpha)\} \cap \{w|R(u_j)(w) \subseteq \Phi(\alpha)\}\\
&= \Phi(\mathtt{OB}_{u_i}\alpha) \cap \Phi(\mathtt{OB}_{u_j}\alpha)\\
&= \Phi(\mathtt{OB}_{u_i}\alpha \wedge \mathtt{OB}_{u_j}\alpha)
\end{aligned}$$

Hence, $\models^{\mathcal{M}}_{w} \mathtt{OB}_{u_i\&u_j}\alpha \leftrightarrow \mathtt{OB}_{u_i}\alpha \wedge \mathtt{OB}_{u_j}\alpha$. ■

**Corollary 6** *Assume, $u_i$ and $u_j$ are authorities. Then:*

$$\begin{aligned}
&\vdash PE_{u_i\&u_j}\alpha \leftrightarrow PE_{u_i}\alpha \vee PE_{u_j}\alpha\\
&\vdash IM_{u_i\&u_j}\alpha \leftrightarrow IM_{u_i}\alpha \wedge IM_{u_j}\alpha\\
&\vdash GR_{u_i\&u_j}\alpha \leftrightarrow GR_{u_i}\alpha \vee GR_{u_j}\alpha
\end{aligned}$$

***Proof*** It is conveniently obtained by applying the definitions of modal statuses and axiom CAU. ■

The above corollary shows that for obtaining the collaborative obligation statement on something (e.g., doing an action on a object) we require to have each of authorities' obligation statement on it. However, for permission statement, having each of the participating authorities' permission statement is enough.

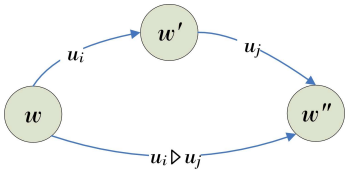


**Figure 2. Semantics of delegative authority with possibility relation composition.**

## 4.2 Delegative Administration

In this paper, by *delegation* we mean *administration delegation*. $u_i \triangleright u_j$ denotes $u_i$ on behalf of $u_j$ privileges can enact statements. For example, $\mathtt{OB}_{u_i \triangleright u_j}\alpha$ means $u_i$ states on behalf of $u_j$ that $\alpha$ is ought to be the case.

It is worthwhile to note that an authority for specifying statements on behalf of another authority needs to have a privilege. Using logic, abstracts us from these implementation details. However, we suppose that trust infrastructures like PKI [11] or delegation network which is proposed in [14] handles delegation details. In this paper, just specification of security policy rules with composite authorities and inference over them is important.

**Axioms:**

- $\vdash \mathtt{OB}_{u_i \triangleright u_j}\alpha \leftrightarrow \mathtt{OB}_{u_i}(\mathtt{OB}_{u_j}\alpha)$ (DAU)
- $\vdash \mathtt{OB}_{u_i \triangleright u_i}\alpha \leftrightarrow \mathtt{OB}_{u_i}\alpha$ (DAI)
- $\vdash \mathtt{OB}_{u_i \triangleright (u_j \triangleright u_k)}\alpha \leftrightarrow \mathtt{OB}_{(u_i \triangleright u_j) \triangleright u_k}\alpha$ (DAA)

Axiom DAU represents that in delegative administration of authority $u_i$ on behalf of $u_j$ rights, a case like $\alpha$ is obligatory from $u_i \triangleright u_j$ point of view if and only if in any state, $u_i$ accepts that $\alpha$ is obligatory from $u_j$'s point of view. Axioms DAI, and DAA impose that delegation operator ($\triangleright$) is idempotent, and associative. Idempotency of delegation compels the relations $R(u_i)$ in the proposed Kripke semantics to be reflexive.

**Semantics:** The extension of $R$ to include delegative composition of authorities is defined as $R(u_i \triangleright u_j) = R(u_i) \circ R(u_j)$.

The relation composition reflects this fact that from $u_i \triangleright u_j$ point of view, a world $w'$ is (normatively) possible from a world $w$ if and only if $w'$ is (normatively) possible from other world like $w$" in $u_j$'s point of view and $w$" is (normatively) possible from $w$ in $u_i$'s point of view (figure 2). Hence, $\mathtt{OB}_{u_i \triangleright u_j}\alpha$ is true at a world $w$ when $\alpha$ is true at every world possible from $w$ by the composite relation.

**Soundness:** The soundness proof of DAI and DAA are obtained by the idempotency, and associativity of relation composition on relation $R(u)$. The soundness proof of axiom DAU is presented in the following proposition.

**Proposition 7 (Soundness of DAU)** *Axiom DAU is sound.*

***Proof*** Suppose for some model $\mathcal{M} = \langle W, R, \phi, \Delta, I\rangle$ and some $w \in W$, we have $\models^{\mathcal{M}}_{w} \mathtt{OB}_{u_i \triangleright u_j}\alpha$. Thus, $w \in \Phi(\mathtt{OB}_{u_i \triangleright u_j})$ if and only if $w \in \Phi(\mathtt{OB}_{u_i}(\mathtt{OB}_{u_j}\alpha))$, because:

$$\begin{aligned}
\Phi(\mathtt{OB}_{u_i \triangleright u_j})\alpha &= \{w | R(u_i \triangleright u_j)(w) \subseteq \Phi(\alpha)\} \\
&= \{w | (R(u_i) \circ R(u_j))(w) \subseteq \Phi(\alpha)\} \\
&= \{w | \forall w', \text{if } (w, w') \in R(u_i) \text{ then, } R(u_j)(w') \subseteq \Phi(\alpha)\} \\
&= \{w | R(u_i)(w) \subseteq \{w' | R(u_j)(w') \subseteq \Phi(\alpha)\}\} \\
&= \{w | R(u_i)(w) \subseteq \Phi(\mathtt{OB}_{u_j}\alpha)\} \\
&= \Phi(\mathtt{OB}_{u_i}(\mathtt{OB}_{u_j}\alpha))
\end{aligned}$$

Hence, $\models^{\mathcal{M}}_{w} \mathtt{OB}_{u_i \triangleright u_j}\alpha \leftrightarrow \mathtt{OB}_{u_i}(\mathtt{OB}_{u_j}\alpha)$. ■

**Corollary 8** *Assume, $u_i$ and $u_j$ are authorities. Then:*

$$\begin{aligned}
&\vdash PE_{u_i \triangleright u_j}\alpha \leftrightarrow PE_{u_i}(PE_{u_j}\alpha) \\
&\vdash IM_{u_i \triangleright u_j}\alpha \leftrightarrow OB_{u_i}(IM_{u_j}\alpha) \\
&\vdash GR_{u_i \triangleright u_j}\alpha \leftrightarrow PE_{u_i}(GR_{u_j}\alpha)
\end{aligned}$$

***Proof*** It is conveniently obtained by applying the definitions of modal statuses and axiom DAU.

The above results show that we can infer that a case $\alpha$ is permissible or gratuitous from the view point of $u_i$ on behalf of $u_j$, if permissible or gratuitous statement of $u_j$ is permissible from $u_i$'s point of view. However, for obligatory or impermissible cases, we do more strict and obligatory statement of $u_i$ over the obligation or impermissible statement of $u_j$ is required.

## 4.3 Disjunctive Administration

Disjunctive administration can be employed when two authorities like to manage the underlying shared domain objects based on the administration opinion of each of them. In this way, each authority can make a statement instead of the whole authorities who are participated in the disjunctive administration of a domain (or subdomain).

In the proposed calculus of authorities, $u_i|u_j$ denotes $u_i$ and $u_j$ enact disjunctively for their underlying domains.

**Axioms:**

- $\vdash \mathtt{OB}_{u_i}\alpha \vee \mathtt{OB}_{u_j}\alpha \rightarrow \mathtt{OB}_{u_i|u_j}\alpha$ (JAU)
- $\vdash \mathtt{OB}_{u_i|u_i}\alpha \leftrightarrow \mathtt{OB}_{u_i}\alpha$ (JAI)
- $\vdash \mathtt{OB}_{u_i|u_j}\alpha \leftrightarrow \mathtt{OB}_{u_j|u_i}\alpha$ (JAC)
- $\vdash \mathtt{OB}_{u_i|(u_j|u_k)}\alpha \leftrightarrow \mathtt{OB}_{(u_i|u_j)|u_k}\alpha$ (JAA)

In disjunctive administration, an obligation statement is derived when at least one of the participating authorities enact the obligation statement. This principle is appeared as

axiom JAU in the proposed calculus. Idempotency, commutativity, and associativity of disjunction operation (|) is added by the JAI, JAC, and JAA axioms respectively.

**Semantics:** The disjunction operation is interpreted as an intersection operation in possibility relation $R(u)$ in the proposed Kripke structure as $R(u_i|u_j) = R(u_i) \cap R(u_j)$.

There is a justification similar to collaborative composition for having intersection of relations $R(u_i)$ and $R(u_j)$ instead of union for the semantics of disjunctive composition of authorities $u_i$ and $u_j$.

**Soundness:** For soundness, we just need to prove the soundness of axiom JAU. Soundness proofs of others are easy.

**Proposition 9 (Soundness of JAU)** *Axiom JAU is sound.*

***Proof*** Suppose for some model $\mathcal{M} = \langle W, R, \phi, \Delta, I\rangle$ and some $w \in W$, we have $\models_w^{\mathcal{M}} \mathtt{OB}_{u_i}\alpha \vee \mathtt{OB}_{u_j}\alpha$. Then:

$$
\begin{aligned}
& w \in \Phi(\mathtt{OB}_{u_i}\alpha \vee \mathtt{OB}_{u_j}\alpha) \textbf{ iff } w \in \Phi(\mathtt{OB}_{u_i}\alpha) \cup \Phi(\mathtt{OB}_{u_j}\alpha)) \\
\textbf{iff } & w \in \{w|\forall w', (w,w') \in R(u_i) \rightarrow w' \in \Phi(\alpha)\} \cup \{w|\forall w', (w,a') \in R(u_j) \rightarrow w' \in \Phi(\alpha)\} \\
\textbf{if } & w \in \{w|\forall w', (w,w') \in R(u_i) \cap R(u_j) \rightarrow w' \in \Phi(\alpha)\} \cup \\
& \quad \{w|\forall w', (w,a') \in R(u_j) \cap R(u_i) \rightarrow w' \in \Phi(\alpha)\} \\
\textbf{iff } & w \in \{w|(w,w') \in R(u_i|u_j) \rightarrow w' \in \Phi(\alpha)\} \textbf{ iff } w \in \Phi(\mathtt{OB}_{u_i|u_j}\alpha)
\end{aligned}
$$

Thus, $\models_w^{\mathcal{M}} \mathtt{OB}_{u_i|u_j}\alpha$. Hence, the axiom is sound. ■

**Corollary 10** *Assume, $u_i$ and $u_j$ are authorities. Then:*

$$
\begin{aligned}
& \vdash \mathit{PE}_{u_i|u_j}\alpha \rightarrow \mathit{PE}_{u_i}\alpha \wedge \mathit{PE}_{u_j}\alpha \\
& \vdash \mathit{IM}_{u_i}\alpha \vee \mathit{IM}_{u_j}\alpha \rightarrow \mathit{IM}_{u_i|u_j}\alpha \\
& \vdash \mathit{GR}_{u_i|u_j}\alpha \rightarrow \mathit{GR}_{u_i}\alpha \wedge \mathit{GR}_{u_j}\alpha \\
& \vdash \mathit{OB}_{u_i|u_j}\alpha \rightarrow \mathit{PE}_{u_i}\alpha \wedge \mathit{PE}_{u_j}\alpha \\
& \vdash \mathit{IM}_{u_i|u_j}\alpha \rightarrow \mathit{GR}_{u_i}\alpha \wedge \mathit{GR}_{u_j}\alpha
\end{aligned}
$$

***Proof*** It is conveniently obtained by applying the definitions of modal statuses, and the OB-MD and JAU axioms. ■

### 4.4 Hybrid Administration

We can have the different combination of the three aforementioned administration styles together. Complicated composition of authorities results in some axioms which are presented in the following.

**Axioms:**

- $\vdash \mathtt{OB}_{u_i \triangleright (u_j \& u_k)}\alpha \leftrightarrow \mathtt{OB}_{(u_i \triangleright u_j)\&(u_i \triangleright u_k)}\alpha$ (DDC)
- $\vdash \mathtt{OB}_{u_i \triangleright (u_j | u_k)}\alpha \leftrightarrow \mathtt{OB}_{(u_i \triangleright u_j)|(u_i \triangleright u_k)}\alpha$ (DDJ)

Axioms DDC and DDJ show the distribution of delegation operation ($\triangleright$)on collaboration (&) and disjunction (|) operations respectively.

**Soundness:** The soundness proof of axioms DDC and DDJ are similar to each other. In the following, the soundness proof of DDC is given.

**Proposition 11 (Soundness of DDC)** *Axiom DDC is sound.*

***Proof*** Suppose for some model $\mathcal{M} = \langle W, R, \phi, \Delta, I\rangle$ and some $w \in W$, we have $\models_w^{\mathcal{M}} \mathtt{OB}_{u_i \triangleright (u_j \& u_k)}\alpha$. Then:

$$
\begin{aligned}
& w \in \Phi(\mathtt{OB}_{u_i \triangleright (u_j \& u_k)}) \\
\textbf{iff } & \forall w', [(w,w') \in R(u_i) \circ R(u_j \& u_k)] \rightarrow w' \in \Phi(\alpha) \\
\textbf{iff } & \forall w', [(w,w') \in R(u_i) \circ (R(u_j) \cup R(u_k))] \rightarrow w' \in \Phi(\alpha) \\
\textbf{iff } & \forall w', [\exists w", (w,w") \in R(u_i) \wedge ((w",w') \in R(u_j) \vee \\
& \quad (w",w') \in R(u_k))] \rightarrow w' \in \Phi(\alpha) \\
\textbf{iff } & \forall w', [(\exists w", (w,w") \in R(u_i) \wedge (w",w') \in R(u_j)) \vee \\
& \quad (\exists w", (w,w") \in R(u_i) \wedge (w",w') \in R(u_k))] \\
& \quad \rightarrow w' \in \Phi(\alpha) \\
\textbf{iff } & \forall w', [(w,w') \in R(u_i) \circ R(u_j) \vee (w,w') \in R(u_i) \circ R(u_k)] \\
& \quad \rightarrow w' \in \Phi(\alpha) \\
\textbf{iff } & \forall w', [(w,w') \in R(u_i \triangleright u_j) \vee (w,w') \in R(u_i \triangleright u_k)] \\
& \quad \rightarrow w' \in \Phi(\alpha) \\
\textbf{iff } & \forall w', (w,w') \in R(u_i \triangleright u_j) \cup R(u_i \triangleright u_k) \rightarrow w' \in \Phi(\alpha) \\
\textbf{iff } & \forall w', (w,w') \in R((u_i \triangleright u_j)\&(u_i \triangleright u_k)) \rightarrow w' \in \Phi(\alpha) \\
\textbf{iff } & w \in \Phi(\mathtt{OB}_{(u_i \triangleright u_j)\&(u_j \triangleright u_k)}\alpha)
\end{aligned}
$$

Thus, $\models_w^{\mathcal{M}} \mathtt{OB}_{(u_i \triangleright u_j)\&(u_j \triangleright u_k)}\alpha$. Hence, the axiom is sound. ■

## 5 Policy Enforcement

Enforcement of policies in the security domains are done by two subsystems. *Obligation enforcement* subsystem which is used to enforce the actions which must be done in a specific situation. *Access control* subsystem which is used to apply access control policies on the access request which are sent by subjects to access the objects or resources in the environment.

### 5.1 Access Control

The centric security mechanism in each system is an access control subsystem. By receiving an access request in such a system, we need to make a decision whether to permit the requested access or not.

Access control in separated domains is easily done based on the security policy rules specified by their authorities.

However, for the shared domains, administration or policy enforcement strategy might be different based on the agreement made between the rightful authorities. In previous sections, cooperative administration is modeled by the concept of composite authority. Hence, for access control, we just need to infer implicit policy rules composed by composite authorities in the shared domains. The way in which a composite authority is determined (specified) is depend on the cooperative administration strategy which we intend for that shared domain. For example, if an authority $u_i$ makes agreement with $u_j$ on having collaborative administration strategy in their shred domain, we must infer implicit policy rules composed by collaborative authority $u_i \& u_j$.

Based on the above description, and the logic-based model which was presented, the access decision making functions are as follow. In these functions, the argument $u \in AU, s \in S, o \in O, a \in A$.

$$\text{BADF}(u,s,o,a) = \begin{cases} Deny & \text{, if KB} \vdash \texttt{IM}_u do(s,o,a) \\ Grant & \text{, if KB} \vdash \texttt{PE}_u do(s,o,a) \\ Don'tCare & \text{, otherwise} \end{cases}$$

$$\text{FADF}(u,s,o,a) = \begin{cases} Deny & \text{, if [BADF}(u,s,o,a) = Deny\text{]} \\ & \quad \text{or [BADF}(u,s,o,a) = Don'tCare \\ & \quad\quad \wedge\ \text{DefSt} = \texttt{IM}\text{]} \\ Grant & \text{, otherwise} \end{cases}$$

BADF (Basic Access Decision Function) goes through the existing policy rules for permissible or impermissible ones and FADF (Final Access Decision Function) makes the final decision. In the definition of FADF, DefSt returns to the default access right which might be set to one of IM, PE, or GR deontic statuses. Thus, if there is no policy rule for a request, the default strategy determines the final decision for the requested access. It is worthwhile to note that in case of existing contradictory policies (from different authorities on the same resource), we may not be able to infer an access decision. In this case, DefSt solves our problem and resolves the conflicts by enforcing the default decision.

### 5.2 Obligation Enforcement

The issues related to the inference of obligation policies are similar to the inference of authorization ones. We follow the same approach for having cooperative administration of security obligations here. However, enforcing obligations is different from the authorizations and requires its enforcement mechanisms and services apart from the access control services.

Monitoring and enforcing obligations (or its restricted type, provisions) is one of the open problems in security management. Since, we concentrated on the specification and inference of security policies including obligation policies in this paper, we do not scrutiny the details of obligation enforcement in this paper. Some approaches for monitoring and enforcement of obligations and provisions are addressed in [7, 9, 19, 20].

## 6 Case Study

For representing the applicability of the proposed logical model and cooperative administration styles, we reveal a small case study based on the collaborative region concept which is introduced in the MIT Oxygen project[1].

**Collaborative Meeting Region:**

In the Oxygen project, a collaborative meeting region is defined as an area with a set of devices for meeting. This region has a set of trust and authorization rules that specify what happens during a meeting [1]. For security management of collaborative region in Oxygen, we suggest applying our proposed model. For this purpose, we define the following scenario.

In a collaborative meeting room, all meeting members (who are participated in a meeting) are authorities of the meeting room (with all resources which are available there). Each meeting member has his/her own security rules (obligation as well as authorization rules). In our case, Bob and Alice are meeting members with the following security policy rules. We suppose that the meeting room has a security agent that we call it MSA in short. MSA is another authority of the room. We will see the role of MSA in this case later.

**Alice's Security Policy Rules:**

Alice likes to delegate the administration to the meeting room's security agent (MSA) by transferring her security policy rules to it. She does not allow printing the confidential documents (in her domain) in any case. She allows meeting members to read the confidential documents when they are in the meeting room for a meeting. She also gives the read-only permission on customers' information to a meeting member when the location of meeting is outside their company. The logical representation of Alice's security policy rules are as follow. Note that in the following policy rules, each statement placed in the parentheses in the left side of a rule, is a proposition.

**[AP1]** (Alice's location is meeting room) $\wedge$ (time is meeting time) $\rightarrow \texttt{OB}_{\text{Alice}}$ **do**(AliceSecAgent, MeetSecPolicies, transferTo(MSA))

**[AP2]** True $\rightarrow \texttt{IM}_{\text{Alice}}$ **do**(Any, ConfDocs, print)

**[AP3]** (requester's location is meeting room) $\wedge$ (time is meeting time) $\rightarrow \texttt{PE}_{\text{Alice}}$ **do**(MeetMember, ConfDocs, read)

**[AP4]** ¬ (requester's location is company) ∧ (location is meeting room) → $\texttt{PE}_{\text{Alice}}$ **do**(MeetMember, CustInfo, read)

**Bob's Security Rules:**

Bob similar to Alice likes to delegate the administration to the meeting room's security agent (MSA). He allows printing any documents to meeting members when they are in the meeting room. However, he does not allow anybody who is outside the company to write or update customers' information.

**[BP1]** (Bob's location is meeting room) ∧ (time is meeting time) → $\texttt{OB}_{\text{Alice}}$ **do**(BobSecAgent, MeetSecPolicies, transferTo(MSA))

**[BP2]** (requester's location is meeting room) → $\texttt{PE}_{\text{Bob}}$ **do**(MeetMember, Docs, print)

**[BP3]** True → $\texttt{PE}_{\text{Bob}}$ **do**(MeetMember, CustInfo, read)

**[BP4]** ¬ (requester's location is company) → $\texttt{IM}_{\text{Bob}}$ **do**(Any, CustInfo, write)

By entering Alice and Bob into the meeting room, following their first obligation policy, their agent (which is installed on their PDAs) must send all the security policies to the meeting security agent (MSA). MSA establishes its security policies based on the received security policies as follows. In this way, MSA obligates every obligation policy specified by Alice or Bob and makes permissible every authorization policy specified by them.

**[MP1]** True → $\texttt{OB}_{\text{MSA}}(\texttt{IM}_{\text{Alice}}$ **do**(Any, ConfDocs, print))

**[MP2]** (requester's location is meeting room) ∧ (time is meeting time) → $\texttt{PE}_{\text{MSA}}(\texttt{PE}_{\text{Alice}}$ **do**(MeetMember, ConfDocs, read))

**[MP3]** ¬ (requester's location is company) ∧ (location is meeting room) → $\texttt{PE}_{\text{MSA}}(\texttt{PE}_{\text{Alice}}$ **do**(MeetMember, CustInfo, read))

**[MP4]** (requester's location is meeting room) → $\texttt{PE}_{\text{MSA}}(\texttt{PE}_{\text{Bob}}$ **do**(MeetMember, Docs, print))

**[MP5]** True → $\texttt{PE}_{\text{MSA}}(\texttt{PE}_{\text{Bob}}$ **do**(MeetMember, CustInfo, read))

**[MP6]** ¬ (requester's location is company) → $\texttt{OB}_{\text{MSA}}(\texttt{IM}_{\text{Bob}}$ **do**(Any, CustInfo, write))

We set the security enforcement system for the collaborative meeting room to enforce the security policies based on collaborative administration of MSA on behalf of each meeting member. In our case, this means the authority which is resulted from the collaborative composition of MSA on behalf of Alice and MSA on behalf of Bob, is agreed for administration of the meeting room. Formally, the desired composite authority is (MSA ▷ Alice)&(MSA ▷ Bob).

Suppose that in the meeting of Alice and Bob, Alice request to update a customer information through her PDA device. By her request, she sends a certificate that she is a meeting member. The security enforcement system by receiving this access request, calls FADF for checking the read and write access of Alice (as a meeting member) to customers information (CustInfo). Note that update operation is interpreted to read and write operations. The arguments of FADF in this case is like the following.

$$\text{FADF}\big((\text{MSA} \triangleright \text{Alice})\&(\text{MSA} \triangleright \text{Bob}), \text{MeetMember, CustInfo, read}\big)$$
$$\text{FADF}\big((\text{MSA} \triangleright \text{Alice})\&(\text{MSA} \triangleright \text{Bob}), \text{MeetMember, CustInfo, write}\big)$$

Following MP3, and MP4 policy rules, BADF infers the following statement for read operation, and so returns $Grant$.

$\texttt{PE}_{(\text{MSA}\triangleright\text{Alice})\&(\text{MSA}\triangleright\text{Bob})}$**do**(MeetMember, CustInfo, read)

However, for write operation, since, it cannot infer neither PE nor IM, returns $Don'tCare$. Therefore, in this situation, FADF grants read operation and denies write operation following the default strategy for access control (Suppose DefSt=IM).

## 7 Conclusions

In pervasive or ubiquitous computing, everything in our environment is supposed to be integrated in information processing. Wide distribution of computational devices in such environments motivates special security management styles. Policy-based approaches in security management of pervasive environments enable dynamic and distributed administration of security in them.

The proposed security model in this paper, is based on splitting the whole environment into some security domains. In each domain, a security administrator (we call it authority) is responsible of specifying security policies of resources registered themselves in the domain. The logical policy specification language, MASL, proposed in this paper, enables authorities in each domain to specify their domain's security policies in an abstract manner and independent of the implementation details in such a heterogeneous computational model. MASL is founded on deontic logic which is enables the specification of obligations as well as authorizations away from any potential conflicts between them.

The gist which is not considered in the previous relative researches is cooperative or composite administration. We proposed three styles of cooperative administration in this

paper; namely collaborative, disjunctive, and delegative administration. Collaborative administration enables enforcing the obligation policies which are specified by all the authorities which are participated in the shared domain's administration. In disjunctive administration, we proposed a less strict approach in which we infer an obligation statement whenever at least one of the authorities states such an obligation. Delegative administration is used to infer policy rules which are stated by an authority on behalf of another authority.

In future steps of our research we are about to take the logical model of domains into account. In this paper we left this concept informally and just supposed that each authority is responsible of one or more domains. Implementing the proposed logical model using logical programming languages and getting through the details of enforcing obligation policies, construct our future steps toward completing this research.

## References


[1] MIT project Oxygenom, 2004. http://oxygen.csail.mit.edu/, Accessed in Nov 2007.

[2] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Programming Languages and Systems*, 15(4):706–734, September 1993.

[3] R. Au, M. Looi, P. Ashley, and L. T. Seet. Secure authorisation agent for cross-domain access control in a mobile computing environment, 2001.

[4] S. Barker, M. Leuschel, and M. Varea. Efficient and flexible access control via logic program specialisation. In *ACM SIGPLAN symposium on Partial evaluation and semantics-based program manipulation (PEPM'04)*, pages 190–199, Verona, Italy, 2004. ACM Press.

[5] S. Barker and P. J. Stuckey. Flexible access control policy specification with constraint logic programming. *ACM Transactions on Information and System Security (TISSEC)*, 6(4):501–546, 2003.

[6] F. Berkes. New and not-so-new directions in the use of the commons: Co-management. *The Common Property Resoruce Digest*, 42:5–7, 1997.

[7] C. Bettini, S. Jajodia, X. S. Wang, and D. Wijesekera. Provisions and obligations in policy rule management. *Journal of Network and Systems Management*, 11(3):351–372, 2003.

[8] F. Cuppens and R. Demolombe. A deontic logic for reasoning about confidentiality. In *3rd International Workshop on Deontic Logic in Computer Science*, pages 66–79, Sesimbra, Portugal, 1996.

[9] P. Gama and P. Ferreira. Obligation policies: An enforcement platform. In *The 6th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05)*, pages 203–212, Stockholm, Sweden, 2005.

[10] J. I. Glasgow, G. H. MacEwen, and P. Panangaden. Reasoning about knowledge and permission in secure distributed systems. In *First IEEE Computer Security Foundations Workshop (CSFW'88)*, pages 139–146, Franconia, New Hampshire, USA, 1988. MITRE Corporation Press.

[11] ISO/IEC:9594-8. ITU-T recommendation X.509: Infomation technology - open systems interconnection - the directory : Public-key and attribute certificate frameworks. Technical report, ITU-T, 2001.

[12] S. Jajodia, P. Samarati, M. L. Sapino, and V. S. Subrahmanian. Flexible support for multiple access control policies. *ACM Transaction on Database Systems*, 26(2):214–260, 2001.

[13] S. Jajodia, P. Samarati, and V. S. Subrahmanian. A logical language for expressing authorizations. In *IEEE Symposium on Security and Privacy*, pages 31–42, Oakland, CA, USA, 1997.

[14] A. Josang, D. Gollmann, and R. Au. A method for access authorisation through delegation networks. In *The 2006 Australasian Workshops on Grid Computing and E-Research (ACSW Frontiers '06)*, pages 165–174, Hobart, Tasmania, Australia, 2006. Australian Computer Society, Inc.

[15] J. B. Joshi, R. Bhatti, E. Bertino, and A. Ghafoor. Access-control language for multidomain environments. *IEEE Internet Computing*, 8(6):40–50, 2004.

[16] L. Kagal, T. Finin, and A. Joshi. Trust-based security in pervasive computing environments. *IEEE Computer*, 34(12):154–157, 2001.

[17] L. Kagal, T. Finin, and A. Joshi. A policy-based approach to security for the semantic web. In *2nd International Semantic Web Conference (ISWC03)*, Sanibel Island, Florida, USA, Oct 2003.

[18] S. Kaushik, D. Wijesekera, and P. Ammann. Policy-based dissemination of partial web-ontologies. In *the 2005 Workshop on Secure Web Services (SWS '05)*, pages 43–52, Fairfax, VA, USA, 2005. ACM Press.

[19] Z. Liu, A. Ranganathan, and A. Riabov. Specifying and enforcing high-level semantic obligation policies. In *The 8th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'07)*, pages 119–128, Bologna, Italy, 2007.

[20] M. C. Mont and R. Thyne. A systemic approach to automate privacy policy enforcement in enterprises. In *6th Workshop on Privacy Enhancing Technologies*, volume 4258 of *Lecture Notes in Computer Science (LNCS)*, pages 118–134, Cambridge, UK, 2006. Springer.

[21] S. Piromruen and J. B. D. Joshi. An RBAC framework for time constrained secure interoperation in multi-domain environments. In *the 10th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS05)*, pages 36–48, Sedona, Arizona, USA, 2005. IEEE Computer Society.

[22] B. Shafiq, J. B. D. Joshi, E. Bertino, and A. Ghafoor. Secure interoperation in a multidomain environment employing rbac policies. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1557–1577, 2005.

[23] Y. Tang, S. Zhang, and L. Li. A mobile access control architecture for multiple security domains environment. In *the International Conference on High Performance Computing and Applications*, pages 457–461, Shanghai, P.R. China, 2004. Springer.

[24] Z. Tang, R. Li, and Z. Lu. A request-driven role mapping for secure interoperation in multi-domain environment. In *The IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007)*, pages 83–90, Dalian, China, 2007.

[25] T. Y. C. Woo and S. S. Lam. Authorizations in distributed systems: A new approach. *Journal of Computer Security*, 2(2-3):107–136, 1993.