

# Mediating Between Strangers: A Trust Management Based Approach

Joachim Biskup   Yücel Karabulut

Fachbereich Informatik, Universität Dortmund, D-44221 Dortmund, Germany  
{biskup,karabulu}@ls6.informatik.uni-dortmund.de

## Abstract

Data sources in *i*-mediation, following property-based security policies, aim at supporting a wide range of potential clients, which are in general unknown in advance and may belong to heterogeneous and autonomous security domains. This raises the challenge how remote and autonomous entities can agree on a common understanding of certified properties, and other issues related to these properties (e.g. encoding formats). This paper proposes solutions that are based on secure *i*-mediation and a hybrid PKI model, which unifies X.509 and SPKI. We present a mediation functionality, called *f*-mediation. Secure *f*-mediation assists entities in finding partners for *i*-mediation and providing them with appropriate certificates and credentials. Thereby, among others, *f*-mediation deals with delegation and conversion of free properties into capability-like bound properties. An extension to the agent communication language KQML is used to implement the interactions among software agents in an instance of the *f*-mediation.

## 1 Introduction

In [3], we proposed a secure mediation approach considering the dynamics and conflicting interests of mediation participants. In a global computing infrastructure like the Internet, entities (strangers) need to reason about the trustworthiness of other entities in order to make autonomous security decisions. Based on this fact, in contrast to traditional (identity-based) access control mechanisms which operate under a closed world assumption, we followed a PKI-based approach in order to achieve the security goals with respect to confidentiality and authenticity. Our approach to secure mediation is based on evidences of clients' eligibility rather than user authentication and access control based on user identities. More precisely, we argued for basing the enforcement of these security goals on certified personal authorization attributes (e.g. organizational membership, security clearance) rather than on identification. In order to focus on the major mediation functionalities and to keep the design of secure mediation manageable, we assumed that the mediation participants agree on a *common*

*understanding* of personal authorization attributes, credential formats, and the certification policies under which the credentials are issued.

This paper is concerned with solutions to challenges which arise when we remove this assumption. We consider the following situation: Information sources may supply their information for purchase as well as for collaboration. While doing this, they may aim at determining a wide range of potential clients which could be interested in requesting specific services of the sources and which are qualified in terms of evidences of their eligibility.

*What is the problem?* Whatever data a source has to offer, it may aim at supporting a *wide range* of potential clients, which are in general unknown in advance and may belong to *heterogeneous* and *autonomous* security domains. *Why is the problem a problem?* The conceptual challenge arising in this situation concerns how remote and autonomous entities can agree on a common understanding of personal authorization attributes and other issues related to these attributes (e.g. credential formats). On the one hand, personal authorization attributes are assigned to clients in their autonomously operating security domains, in principle without knowing their later usage. On the other hand, sources independently define their security policies in terms of these attributes. *What is the solution?* In such situations the sources wish to be assisted to determine potentially eligible clients. To reach potentially eligible clients, a source could use a specific mediator which could mediate between a source's property-based security policy and clients' personal authorization attributes which have been asserted by some trusted parties. *Why is the solution a solution?* Electronic business transactions will involve asserted commitments, properties, etc. from many parties and the participants of such transactions will, in general, not be in a position to understand or manage everything that is involved. To reach potentially eligible clients, which might belong to remote security domains, the sources will need to *trust* mediating agents having the required domain expertise as well as the relationships with the potential clients.

As a concrete solution, we propose an additional mediation functionality, called *entity finding mediation*, *f*-mediation for short. *F*-mediation employs our hybrid PKI model [4] and our authorization model [3]. In order to prove the key ideas of *f*-mediation, we extended KQML<sup>1</sup> [12] and developed an agent-oriented and KQML-based prototype implementation [18, 19].

## 2 Secure I-Mediation

In mediated information systems [26], a client seeking information and various autonomous sources holding potentially useful data, are brought together by a third kind of independent components, called *mediators*. Mediation is required to deal with heterogeneity and the autonomy of the sources, not only from the functional point of view but also with respect to all aspects of security, such as confidentiality and authenticity. The design of our approach of secure mediation [3] is shortly outlined as follows: A client proves his eligibility

---

<sup>1</sup>KQML stands for Knowledge Query and Manipulation Language.

to see a piece of information by a collection of so-called personal authorization attributes assigned by appropriate trusted authorities. Such assignments are encoded within digitally signed digital credentials<sup>2</sup>. An information source always receives a mediated request to deliver some information together with a set of credentials stemming from the pertinent client. Then the source decides on the permission of the request by evaluating the credentials and the contained personal authorization attributes with respect to its *confidentiality* policy. In case of an allowance, the returned data is encrypted with the public keys found in the credentials on which the permission decision has been based. Thus the returned data can only be decrypted by that client who has proven his eligibility by showing an appropriate collection of personal authorization attributes.

In such scenarios, the mediation participants appear in the following roles: A *client* is characterized by her properties (i.e. assigned by some trusted authorities) and seeks for information. Each heterogeneous and autonomous *source* offers data and follows a security policy expressed in terms of characterizing properties. A *mediator* has two major functions: *a)* From the functional point of view, the main role of a mediator is to retrieve, homogenize and assemble data from any sources the mediator may find worthwhile to contact. *b)* From the security point of view, the mediator contacts only the sources, whose security policies match a client's characterizing properties. Thus, seen from the client, a mediator acts as a kind of *filter* put in front of the sources. We call such a mediator an *information integrating mediator*, *i-mediator* for short, and the process of mediation *i-mediation* [3, 18]. The owner of an *i-mediator* may want to maintain data owned by the *i-mediator* himself. Thus, an *i-mediator* can nearly be treated like a source. As it is a source, an *i-mediator* may also want to protect its data with respect to confidentiality by following a property-based security policy.

### 3 Hybrid PKI Model Revisited

Most of the works, e.g., [21, 15, 27, 23, 7, 20], investigating the application of certificate/credential-based access control treat current PKI models [1, 11, 5, 6] as competing technologies, even some consider them as dueling theologies [2]. We take a different position. In many real-world scenarios, trust relationships consist of hierarchies, trust networks, and combinations of two. Therefore, we argue that a trust management infrastructure for a dynamic computing environment has to use and to link existing PKI models.

In a distributed system, neither the entities themselves nor their properties are directly visible to other entities. An entity uses the matching public parts of its key pairs as visible surrogates for itself. From the perspective of the visible virtual views, these surrogates are called *principals*. A property assignment to an entity in the (real) world is *presumably\_captured\_by* a *digital document* in the visible virtual world. Such a document is called a *certificate* or a *credential*,

---

<sup>2</sup>In the present paper, we would prefer to call the document a certificate in the sense of Section 3.

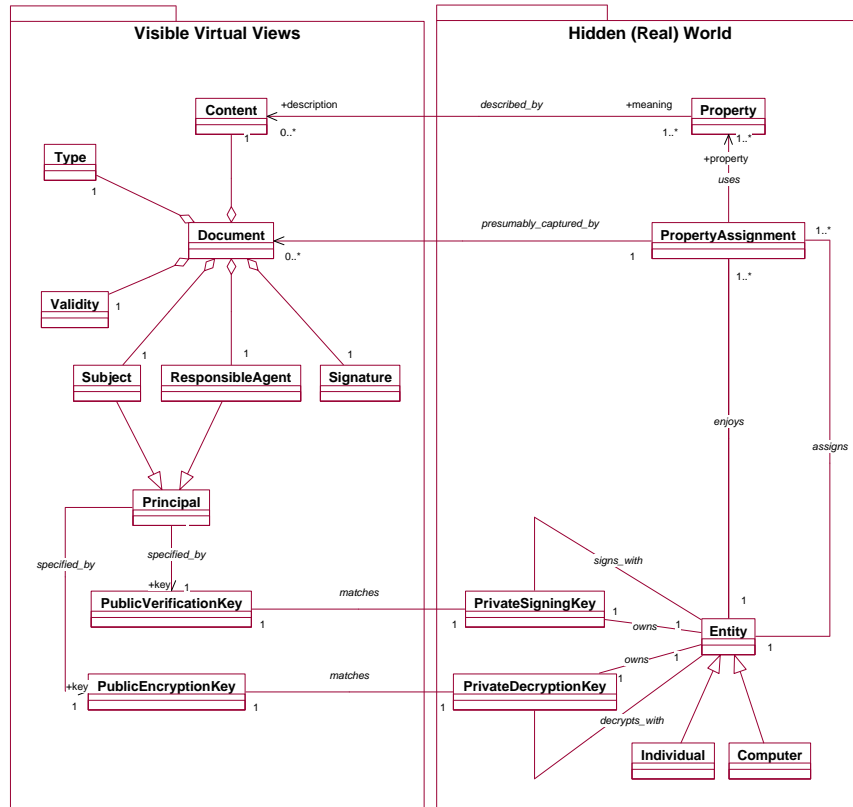


Figure 1: Property assignment: hidden (real) world and visible virtual views

depending on the details explained below. As a consequence, security policies and permission decisions of an entity as a resource owner are solely grounded on the locally available visible view on the global (real) world. This sketched exposition is visualized by Figure 1.

We distinguish two kinds of characterizing properties. A *free property* is intended to express some feature of an entity by itself (e.g. personal data, a technical detail, a skill, an ability). A *bound property* is intended to express some relationship between a client entity and another entity which might act as a server (e.g. a ticket, a capability, a role). While enjoying a free property usually does not entail a guarantee to get the permission for a specific service, enjoying a bound property entails the promise to get a specific service as expressed in the relationship. The assignment of a characterizing property to entities is regulated by corresponding *administrative* properties (e.g. delegatee, licensor) for the characterizing property which must be hold by the entities as responsible issuers.

Following and extending the basic approach of X.509 [1], free properties and the corresponding certificates<sup>3</sup> are handled by trusted authorities using licencing (as shown on the left side of the upper part of Figure 2). The crucial point here is that in general the issuer and the holder of the certificate are different from the entity which afterwards inspects the certificate.

Naturally, it should be possible for any entity, as owner of its resources, to define his own vocabulary for bound properties, to grant corresponding digital documents, and even, to express his trust in delegates, each of which is entitled to assign a specific bound property (defined in the owner's vocabulary) to other entities. Following and extending the basic approach of SPKI [11], bound properties and the corresponding credentials are handled by owners of services using delegation (as shown on the right side of the lower part of Figure 2).

We are mainly interested in the analysis of how an entity can exploit its free properties in order to acquire bound properties or administrative properties for bound properties. Our hybrid PKI model suggests protocols which follow a *property conversion policy*. A property conversion policy specifies *which set of free properties an entity has to enjoy in order to obtain a bound property or a corresponding administrative property assignment*.

The middle part of Figure 2 visualizes the situation. The entity on the right is the grantor following a property conversion policy. The entity in the center requests a promise for a permission, i.e., a bound property. The grantor, after verifying the submitted free property-certificates with the supporting licences, applies his conversion policy on the free properties extracted from the submitted certificates, and finally, if all checks have been successfully completed, grants a bound property-credential where the subject (grantee) is the same as in the submitted free property-certificates. Figure 2 visualizes an instance of the hybrid PKI model linking previous PKI models.

## 4 Secure F-Mediation

### 4.1 Requirements

As stated in Section 1, information sources may supply their information for purchase as well as for collaboration. In the former case, the motivation of a source might be to broaden its potential customer base. In the latter case, a source's main motivation might be to broaden its potential collaborators base by sharing its resources with the *locally eligible* users of the potential remote partner-organizations. Then, a source's security policy can be based on the clients' characterizing properties which are related to the clients' organizational activities and responsibilities (e.g. organizational role membership or group membership) within the corresponding organizations to which the clients belong. The intended high level functionality common to both cases is *finding* potential clients and *stimulating* them to access resources. Accordingly, a mediator assisting the

---

<sup>3</sup>In contrast to X.509 attribute certificates, a certificate in our model is not associated with any identity certificate.

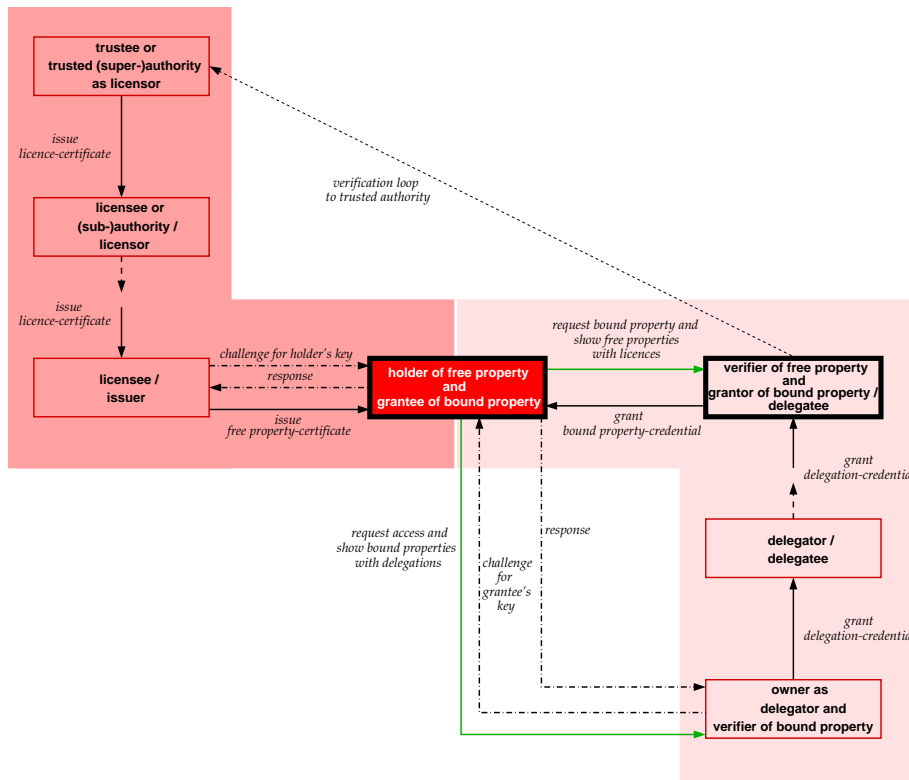


Figure 2: Outline of an instance of the hybrid model for a PKI

sources for this functionality has the following major functions: *a)* From the functional point of view, the main role of a mediator is to seek out an entity *B* for another entity *D* and stimulate *B* to contact *D*. *b)* From the security point of view, the mediator acts as a broker between independently operating security domains of the mediation participants by mapping the properties of an entity *B* on the property vocabulary and the security requirements of the other entity *D*. We call such a mediator an *entity finding mediator*, *f-mediator* for short. The process of mediation using *f*-mediators is called *f-mediation*.

We consider the following example scenario motivating our design and implementation of the *f-mediation* functionality. Autonomously operating forensic institutions of European countries offer anonymous forensic data about sex offenders, and a fictitious *i*-mediator conducted by the European Union is specialized in European forensic institutions. Besides serving spontaneous users, the *i*-mediator may also want to share some part of his data with a discretionary determined kind of users, e.g., the researchers working in US forensic institutions. For this purpose, the *i*-mediator may follow a security policy specifying that *an entity can be granted the local role "visiting researcher", if she is a "psy-*

*choanalyst*” and working in a US forensic institute as a “principal investigator”. We can hardly assume that the *i*-mediator, as well as the forensic institutions abroad, can reach a common understanding about personal attributes and organizational roles implicitly. To reach his potential clients, the *i*-mediator could contact the head of a corresponding FBI unit having the required expertise about mapping between the properties used in the *i*-mediator’s security policy and the properties used in the security domains of the appropriate US forensic institutions.

The other way round, *f*-mediators can also be utilized by the clients. In this case, a client’s motivation might be to determine which *qualified* sources are willing to offer him a specific service and then request them. In such scenarios, the function of a *f*-mediator is to find the most appropriate sources which, on the one hand, are qualified according to the client’s requirements and, on the other hand, may accept clients as potentially eligible entities based on their asserted characterizing properties. For the sake of simple exposition, in this paper, we assume that the *f*-mediators are only utilized by the sources. However, our design and implementation of *f*-mediation is flexible enough to realize other recasted scenarios.

Based on the requirements discussed above and in Section 1, we present a general design for secure *f*-mediation in the following sections.

## 4.2 Design

### 4.2.1 Security Requirement

The following fundamental security requirement is considered:

Any *authorizer* autonomously follows a security policy which ensures that *requested information* is delivered only to appropriate *requestors*. In order to achieve this goal, requestors have to provide *evidence* that they are eligible for requested information, and authorizers have to maintain mechanisms to inspect such evidence and to decide whether and which information is returned. Furthermore, an authorizer has to ensure that information is actually usable to only that requestor which provided the inspected evidence.

### 4.2.2 Informational Environment

We assume that there exists a trust management infrastructure based on our hybrid PKI model (see Section 3). This infrastructure provides the basic PKI functionalities required for the design of the secure *f*-mediation. As indicated in Section 4.2.1, the fundamental security requirement considers entities acting in two modes, as authorizer and requestor, respectively. However, all entities should be able to act in any of these modes during their lifetimes. For the sake of conciseness, for a particular instance of *f*-mediation, we only consider the specific entities *resource owner*, *f-mediator*, and *client* acting as follows: the *resource owner* only acts as authorizer, the *f-mediator* might act in both modes, and the *client* only acts as requestor.

### 4.2.3 Interactions and Basic Protocols

The fundamental security requirement (see Section 4.2.1) has several specific interpretations, each of which result from (a) appropriately replacing the mentioned modes (e.g. authorizer and requestor) by two of the three entities in an instance of  $f$ -mediation (e.g. resource owner,  $f$ -mediator, client) or in an instance of  $i$ -mediation (e.g. resource owner,  $i$ -mediator, client), respectively, and (b) specifying the kind of requested information and the kind of needed evidence. Tables 1 and 2 summarize the most important interpretations for  $i$ -mediation and  $f$ -mediation, respectively. These interpretations result from the concrete interactions among the entities involved.

Inter-action	<i>authorizer</i>	<i>requested information</i>	<i>requestor</i>	<i>needed evidence</i>	<i>security policy</i>
I	resource owner	source-specific service	client	free properties	confidentiality policy
II	resource owner	bound properties	client	free properties	property conversion policy

Table 1: Instantiations of the fundamental security requirement for direct contact and  $i$ -mediation

Inter-action	<i>authorizer</i>	<i>requested information</i>	<i>requestor</i>	<i>needed evidence</i>	<i>security policy</i>
III	resource owner	administrative property for bound property	$f$ -mediator	free properties	delegation policy
IV	$f$ -mediator	bound properties	client	free properties	property conversion policy
V	resource owner	source-specific service	client	bound properties	reconfirmation policy

Table 2: Instantiations of the fundamental security requirement for  $f$ -mediation

In the simplest case, a client contacts directly the source and shows his certified free properties. In secure  $i$ -mediation, as designed in [3] and outlined in Section 2, an  $i$ -mediator is involved. However, concerning the current point of interest, the  $i$ -mediator basically only forwards requests and responses. So we can consider both situations together.

**Interaction I:** A client acts as a requestor (possibly assisted by an  $i$ -mediator). A resource owner follows his confidentiality policy (as security policy) to allow or deny access to content information (source-specific service) based on shown personal authorization attributes (as evidence in form of specific free properties).

**Interaction II:** In a degenerated form, a resource owner may be able to deal with converting free properties into bound properties on his own. Accordingly, the client (possibly assisted by an  $i$ -mediator) applies for a bound property with respect to a source-specific service directly from the resource owner. The source,

after verifying the shown free properties, follows his property conversion policy (as security policy aiming at confidentiality). In the positive case, the resource owner assigns a bound property to the client and grants a corresponding bound property-credential, i.e. to express a possibly conditional permission to access a service.

In more advanced cases, additionally, a *f-mediator* is involved. The trustworthiness of the *f-mediator* may be determined based on the previous direct organizational or business relationships or on the recommendations or on the evidences of *f-mediator*'s eligibility in terms of certified free properties (e.g. personal authorization attributes). In the context of this paper, we focus on the last case, as examined by the following interaction.

**Interaction III:** A *f-mediator* acts as a requestor. The *f-mediator* claims to be a qualified entity (e.g. the head of a specific FBI unit). He shows his pertinent certified free properties and requests an administrative property for a bound property from a resource owner. The source, as the owner of a specific service and a vocabulary for service-specific bound properties, after verifying the shown free properties, follows his delegation policy as a special property conversion policy (as security policy). In case of allowance, the resource owner assigns an administrative property to the *f-mediator* and grants a corresponding delegation credential, the content of which roughly means "*can speak for the owner*" (in the sense of [17]) to assign a specific bound property, i.e. to grant corresponding bound property-credentials. *F*-mediation can be transitively organized by using redelegation of received authorities. A *f-mediator* (as authorizer) can express his trust in another *f-mediator* (as requestor) to speak for the former *f-mediator* in turn. For the sake of simplicity, we don't consider the possible interactions among *f-mediators*.

**Interaction IV:** A client shows his certified free properties and applies for a bound property from a *f-mediator* who is acting as an authorizer on behalf of and in explicit delegation of a resource owner. In order to assign a bound property and a corresponding credential, the *f-mediator* performs the steps carried out by the source during the interaction II.

**Interaction V:** A resource owner is contacted by a client who requests a service. The resource owner, after verifying the submitted bound property-credentials with the supporting delegations, follows something like his "reconfirmation policy" for bound properties (as security policy aiming at confidentiality) to allow or deny access to content information (as source-specific service) based on shown bound properties.

The security policies applied during the interactions II, III and IV are based on a property conversion policy of our hybrid PKI model, as outlined in Section 3. The high level functionality common to these interactions is *showing* free-property certificates to an authorizer in order to *acquire* appropriate credentials. In the following we sketch a **protocol for secure credential acquisition** recasting our *protocol for secure query answering* [3, 18].

We distinguish a preparatory phase and an acquisition phase. In the *preparatory phase*, requestors and authorizers do not interact yet. A requestor, wishing to acquire a credential later on, collects his free property-certificates. On de-

mand and by interaction, the requestor also gets the issue requirements for the properties to be acquired, i.e. the requestor can ask to be informed about which free properties are likely to be sufficient to acquire which (bound) (administrative) properties. And an authorizer, entitled to assign (bound) (administrative) properties later on, defines an appropriate security policy which relates free properties to the amounts of (bound) (administrative) properties allowed for assignment. The set of free property-certificates, accepted by a security policy as input, must belong to a unique requestor or at least a group of requestors which consciously cooperate. It is not necessary for the authorizer to know the identity of that requestor.

The *acquisition phase* is outlined as follows:

1. The requestor sends a request (return information, requested (bound) (administrative) properties, set of free property-certificates) to the authorizer.
2. The authorizer verifies each free property-certificate and determines the associated free properties.
3. The authorizer evaluates the request by following the pertinent property conversion policy. The resulting set of properties is the intersection of the set of requested properties and the largest permitted set of properties (computed on the basis of the associated free properties).
4. For each of the resulting (bound) (administrative) properties, the authorizer grants a corresponding credential.
5. The signed credentials are sent back following the directions given by the return information.

By making some minor modifications to the protocol sketched above, we get a further protocol fulfilling the functionality outlined in the interaction **V**. For *i*-mediation [3], we presented a similar protocol employing free properties instead of bound properties, as outlined by the interaction **I**.

### 4.3 Implementation

We have developed an agent-oriented prototype implementation (which constitutes a testbed) for demonstrating the basic functionality of secure *f*-mediation in combination with *i*-mediation. In order to focus on the implementation of the basic *f*-mediation concepts, so far we limited the functionality of an *i*-mediator and data sources to an owner's functionality (see the model of owners and delegations in Section 3 and [4]). Figure 3 shows the security architecture of the implementation, and the structure of the common agent core (see Section 4.3.2).

We have implemented software agents of four kinds according to the activities of the entities involved in such a composed mediation scenario (see Figure 2): *trusted authority agents* representing issuers of free properties and corresponding administrative properties as trusted authorities and licencees; *f-mediator agents* representing verifiers of free properties and grantors of bound properties as delegates; *user agents* representing holders of free properties and grantees of bound properties; *i-mediator agents* and *data source agents* representing grantors of administrative properties for bound properties, and grantors and verifiers of bound properties as delegators and owners. In Figure 2, *licensees* as well *delegates* are

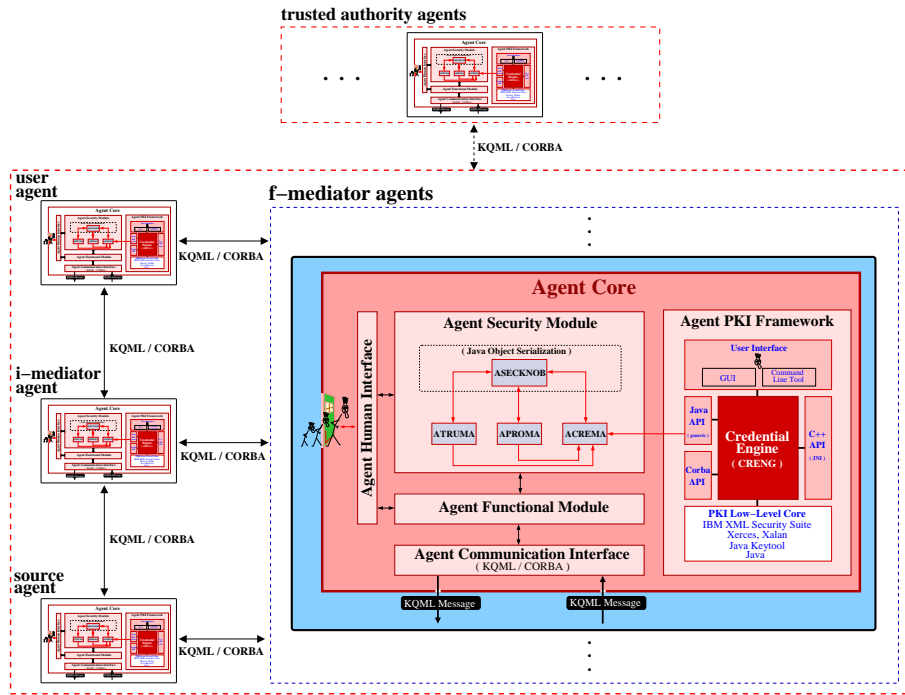


Figure 3: Security architecture

organized transitively, but not further considered in this paper.

For our prototype implementation we primarily focused on three aspects. The first aspect handles the internal authorization model needed for granting privileges including credentials to appropriately represented grantees, and for deciding on the service requests of the requestors. The second aspect is concerned with the structure of the software agents. And the third aspect is related to the KQML-based communication between the agents. We deal with each of these aspects in the following sections.

#### 4.3.1 Authorization Model

As common base for all interactions of *f*-mediation as well as direct contacts and *i*-mediation, we need an internal authorization model that provides syntactic means for (a) expressions over free properties as *grantees* (see interactions **I**, **II**, **III**, and **IV**), (b) expressions over bound properties as *grantees* (see interaction **V**), (c) expressions over bound properties as *privileges* (see interactions **II** and **IV**), (d) administrative property for a bound property as *privileges* (see interaction **III**), and (e) source-specific services, interpreted as access allowances, as *privileges* (see interactions **I** and **V**). For this purpose, we employed an extension of our authorization model designed for *i*-mediation [3, 18], where an authorization is an aggregation including a privilege and a grantee.

### 4.3.2 The Agent Core

In general, all agents should be able to perform as any of the actors (e.g. grantor, issuer, etc.) during their lifetime (see Figure 2). To achieve this goal, we implemented an *agent core* providing a common *core functionality* which is available to all agents. The main features and the structure of the agent core are reported in [19] and sketched in the following (see Figure 3).

The agent core [18, 19] is implemented in Java and on a Solaris platform. The agent core consists of following five main modules: The **agent security module** has four components. The *agent security knowledge base* ASECKNOB maintains a property database, a trust relationships database containing the public keys of the trusted agents, the certificates and credentials, and a (Horn clause) rule base that specifies implications among properties. The *agent credential manager* ACREMA is programmed to perform the tasks related to issuing and evaluating XML-encoded certificates and credentials. The *agent property manager* APROMA implements a property conversion policy (see Section 3). The *agent trust manager* ATRUMA implements the operations needed to store and to retrieve the information (e.g. public keys) about the trusted agents. The **agent functional module** is a kind of scheduler analysing the incoming KQML performatives (see Section 4.3.3) and scheduling the protocol steps to be executed. The **agent human interface** is designed as an interface for the administrators of the agents to use and set up the corresponding agents. The **agent communication interface** implements classes and functions for sending and the reception of CORBA messages wrapping the KQML performatives. The **agent PKI framework** is a collection of tools providing basic PKI services.

### 4.3.3 KQML Extensions

The Knowledge and Query Manipulation Language, KQML [12], is a language that is designed to support interaction among intelligent software agents. In a KQML-based agent architecture, the agents communicate by sending certain kinds of messages, called *performatives*, to each other. For example, a KQML performative is the high-level “ask”, which demands the recipient for a query operation on a knowledge base. KQML is complementary to approaches to distributed computing, like CORBA, which focus on the transport level.

In the original version of the KQML [12], security issues were not taken into consideration. The works in [22] and [14] made some changes with respect to secure communications and PKI related communications. We followed and extended the approaches of [22, 14] such that they satisfy the requirements of the composed secure mediation. Thereby, we proposed a new KQML ontology and recasted some performatives from [22, 14] and added new ones [18]. The new ontology is called secure mediation PKI, *smpki* for short, and enables the agents to know that the KQML performative they received concerns interactions involving in an instance of the composed secure mediation. We use XML as encoding format for the data communicated through performatives. The recasted

and added performatives are outlined below.

**applyDocument** This performative has a dual usage. It is used for applying for free property-certificates from trusted authority agents<sup>4</sup> as well as for applying for bound property-credentials (see interactions **II** and **IV**) and for delegation credentials (see interaction **III**). A requestor agent sends the following recasted performative to a corresponding authorizer agent:

```
applyDocument  :language XML
                :ontology smpki
                :content <requested properties and principal>
                [:certificateChains <free property-certificates and
                corresponding licences>]
```

**authChallenge and authResponse:** These recasted performatives can be employed in all interactions discussed in 4.2.3. For the sake of simple exposition in that section, we omitted the steps corresponding to these performatives. The performative **authChallenge** is used by the agents acting as authorizer. Before issuing a certificate or granting a credential, the issuer or grantor, respectively, has to *challenge*<sup>5</sup> the claiming requestor to prove that she holds the matching private key. In our prototype, the proof is accomplished by an appropriate *response* (i.e. encoded in a corresponding **authResponse** performative) which is generated with the matching private key.

An issuer or a grantor sends the following performative to the requestor agent acting on behalf of the claiming entity:

```
authChallenge  :language XML
                :ontology smpki
                :content <nonce>
```

The content of **content** in the **authChallenge** performative contains a randomly generated string which is to be signed by the claiming entity. The claiming entity signs the string received and sends the following performative including the signed string to the corresponding agent:

```
authResponse   :language XML
                :ontology smpki
                :content <signed nonce>
```

Obviously, also more advanced challenge-response procedures could be exploited.

**issueCredential:** This recasted performative is used by the agents to issue a certificate or a credential. The agent can send the following performative to other agents which have previously applied for a certificate or a credential by using the **applyDocument** performative:

<sup>4</sup>We omitted this interaction in Section 4.2.3

<sup>5</sup>In some cases, it might be sufficient for a grantor to evaluate solely the certificate chains sent by an agent as requestor before granting a credential, since the grantor may only want to gain assurance, whether the property encoded in the “main document” of a certificate chain is bound to the public key included in this document. In such cases, a grantor might not need to apply a challenge-response protocol.

```

issueCredential  :language XML
                 :ontology smpki
                 :content <issued certificate or granted credential
                 with the corresponding chain of supporting documents>

```

In addition to using the recasted performatives and standard KQML performatives (e.g. `ask_all`, `tell`, etc.), we designed the following new performative.

**reduceCredential:** The (so far limited) functionality of an *i*-mediator agent and data source agents is to handle the incoming `reduceCredential` performatives. This performative is designed to be used by user agents in order to apply for a reduced credential. A user agent may send the following performative to some receiver:

```

reduceCredential :language XML
                 :ontology smpki
                 :content <chain of credentials>

```

The parameter `content` contains a chain of credentials which has been previously gathered from appropriate agents.

The receiver can immediately reduce the chain and sign the resulting reduced credential, if he himself is the origin of the chain. Otherwise, the receiver could still perform the reduction [9], but he cannot properly sign the result. In that case, the receiver forwards the chain to the origin, who in turn sends back the properly signed reduction result via the receiver to the user. Though the receiver, for instance an *i*-mediator, may not be able to properly sign a reduction result, he can nevertheless base his own access decisions on it.

## 5 Comparison and Conclusions

Existing works, which are related to the challenges we tackled in this paper, are rooted in three research areas: secure mediation, certificate/credential-based access control, and the employment of KQML for implementing PKI-based security architectures. Contributions to secure *i*-mediation [8, 25, 10] employ either identity-based or security clearance-based authentication and authorization approaches which appear to be less useful for *i*-mediation scenarios which we consider. To our knowledge, no credential-based secure *f*-mediation approach has been proposed to date for establishing interoperability between the entities of two heterogeneous and autonomous security domains. The traditional way of accomplishing this task is to build coalitions between these security domains by committing coalition agreements (e.g. [13]). Such agreements aim at building common vocabularies and a contractually involved cross-certification [24]. The main problem opposing the approach of cross-certification, is that ad-hoc cross-certification between commercial and organizational PKIs is difficult to achieve due to heterogeneous certification policies. Most of the works, e.g., [21, 15, 16, 27, 23, 7, 20, 13], investigating the application of certificate/credential-based access control treat previous PKI models (discussed in Section 3) as competing approaches and base their work on a sin-

gle PKI model. Even some of these works abstract from any particular PKI model (e.g. [13]). In contrast to these works, our proposal is based on a hybrid PKI model. The KQML extensions [22, 14] propose performatives with respect to secure communications and PKI related communications. As demonstrated in Section 4.3.3, our work defines a new ontology, recasts some of their KQML performatives and add new ones.

There are various topics for future research and development. For instance, we would like to integrate distributed role-based access control concepts [20], or negotiations [27, 7] into our approach. Further on, we plan to implement an advanced prototype which allows us to evaluate the actual performance of our approach in terms of effectiveness and efficiency.

## References

- [1] ITU-T recommendation X.509: The directory - public-key and attribute certificate frameworks, 2000.
- [2] Dueling theologies. In *1st Annual PKI Research Workshop*, Gaithersburg, Maryland, USA, Apr. 2002.
- [3] C. Altenschmidt, J. Biskup, U. Flegel, and Y. Karabulut. Secure mediation: Requirements, design and architecture. *Journal of Computer Security*. To appear.
- [4] J. Biskup and Y. Karabulut. A hybrid PKI model with an application for secure mediation. In *16th Annual IFIP WG 11.3 Working Conference on Data and Application Security*, Cambridge, England, July 2002. To appear.
- [5] M. Blaze, J. Feigenbaum, and A. Keromytis. The KeyNote trust management system version 2. RFC 2704, IETF, Sept. 1999.
- [6] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *17th IEEE Symposium on Security and Privacy*, pages 164–173, Los Alamitos, 1996.
- [7] P. Bonatti and P. Samarati. Regulating service access and information release on the web. In *Proceedings of the 7th ACM Conference on Computer and Communication Security*, pages 134–143, Athens, Greece, Nov. 2000.
- [8] K. S. Candan, S. Jajodia, and V. S. Subrahmanian. Secure mediated databases. In S. Y. W. Su, editor, *12th*, pages 28–37, New Orleans, Louisiana, USA, Feb. - Mar. 1996. IEEE, IEEE Computer Society Press.
- [9] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest. Certificate chain discovery in SPKI/SDSI. *Journal of Computer Security*, 9(4):285–322, 2001.
- [10] S. Dawson, S. Qian, and P. Samarati. Providing security and interoperation of heterogeneous systems. *Distributed and Parallel Databases*, 8(1):119–145, Jan. 2000.
- [11] C. Ellison. SPKI/SDSI certificates. <http://world.std.com/~cme/html/spki.html>, Aug. 2001.
- [12] T. Finin, Y. Labrou, and J. Mayfield. KQML as an agent communication language. In J. M. Bradshaw, editor, *Software Agents*. MIT Press, Cambridge, 1997. <http://www.cs.umbc.edu/kqml/papers/>.
- [13] H. M. Gladney. Safe deals between strangers. Technical report, IBM Research Report RJ 10155, July 1999. <http://xxx.lanl.gov/ftp/cs/papers/9908/9908012.pdf>.
- [14] Q. He, K. P. Sycara, and T. Finin. Personal Security Agent: KQML-Based PKI. In *Proceedings of the 2nd International Conference on Autonomous Agents*, pages 377–384. ACM Press, 1998.

- [15] A. Herzberg and Y. Mass. Relying party credentials framework. In D. Naccache, editor, *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference*, LNCS 2020, pages 328–343, San Francisco, CA, 2001.
- [16] A. Herzberg, J. Mihaeli, Y. Mass, D. Naor, and Y. Ravid. Access control meets public key infrastructure, or: Assigning roles to strangers. In *IEEE Symposium on Security and Privacy*, Oakland, USA, May 2000.
- [17] J. Howell and D. Kotz. A formal semantics for SPKI. In *Proceedings of the 6th European Symposium on Research in Computer Security (ESORICS 2000)*, LNCS 1895, pages 140–158, Toulouse, France, Oct. 2000. Springer-Verlag.
- [18] Y. Karabulut. *Secure Mediation Between Strangers in Cyberspace*. PhD thesis, University of Dortmund, Sept. 2002.
- [19] Y. Karabulut. Implementation of an agent-oriented trust management infrastructure based on a hybrid PKI model. In *1st International Conference on Trust Management*, Heraklion, Crete, Greece, May 2003. To appear.
- [20] N. Li, J. C. Mitchell, and W. H. Winsborough. Design of a role-based trust-management framework. In *IEEE Symposium on Security and Privacy*, pages 114–130, Berkeley, California, USA, May 2002.
- [21] P. Nikander. *An Architecture for Authorization and Delegation in Distributed Object-Oriented Agent Systems*. PhD thesis, Helsinki University of Technology, Mar. 1999.
- [22] C. Thirunavukkarasu, T. Finin, and J. Mayfield. Secret agents - a security architecture for the KQML agent communication language. In *4th International Conference on Information and Knowledge Management - Workshop on Intelligent Information Agents*, Baltimore, Maryland, USA, Dec. 1995.
- [23] W. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson, and A. Essiari. Certificate-based access control for widely distributed resources. In *Proceedings of the 8th USENIX Security Symposium*, Washington D.C., Aug. 1999.
- [24] J. Tumbull. Cross-certification and PKI policy networking. [http://www.entrust.com/resources/pdf/cross\\_certification.pdf](http://www.entrust.com/resources/pdf/cross_certification.pdf), Aug. 2000.
- [25] G. Wiederhold, M. Bilello, and C. Donahue. Web implementation of a security mediator for medical databases. In T. Y. Lin and S. Qian, editors, *Database Security, XI: Status and Prospects, Proceedings of the 11th Annual IFIP WG 11.3 Working Conference on Database Security*, pages 60–72, Lake Tahoe, California, 1998. IFIP, Chapman & Hall.
- [26] G. Wiederhold and M. Genesereth. The conceptual basis for mediation. *IEEE Expert, Intelligent Systems and their Applications*, 12(5):38–47, Sept.-Oct. 1997.
- [27] T. Yu, M. Winslett, and K. Seamons. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. *ACM Transactions on Information and System Security*, 6(1), Feb. 2003.