A Publish/Subscribe Model for Secure Content Driven XML Dissemination

Mohammad Ashiqur Rahaman[1 2], Yves Roudier[1] and Andreas Schaad[3]

[1]EURECOM 2229 route des Cretes - B.P. 193, 06904 Sophia Antipolis Cedex, France, {*mohammad.rahaman|yves.roudier}@eurecom.fr*

[2]INRIA Domaine de Voluceau – Rocqencourt B.P. 105, 78153 Le Chesnay Cedex, France, *mohammad.rahaman@inria.fr*

[3]SAP Research, Vincenz-Priessnitz-Str. 1, 76131, Karlsruhe, Germany, *andreas.schaad@sap.com*

Abstract**:** Collaborating on complex XML data structures is a non-trivial task in domains such as the public sector, healthcare or engineering. Specifically, providing a distributed XML content dissemination services in a selective and secure fashion is a challenging task. This paper proposes a publish/subscribe infrastructure to disseminate enterprise XML content utilizing document semantics. Our approach relies on the dissemination of XML documents based on their content, as described by ontology concepts that form the basis for an interoperable description of XML documents. This infrastructure leverages our earlier encryption enabled document parsing [20] scheme for protecting the integrity and confidentiality of XML content during dissemination.

Keywords**:** XML, Ontology, Dissemination**.**

1. Introduction

Due to the rise of cross-organizational communication based on common XML processing standards such as XML schema, XSL, SOAP, WSDL or BPEL, an increasing number of business-related XML documents are exchanged through the internet. These documents may have a complex structure and rich semantics which we consider typical for enterprise applications such as enterprise resource planning (ERP) or supply chain management (SCM). We term such documents as 'Enterprise XML'. Today's cross-organizational communication mostly relies on a client-server interaction model which is not tailored for all business cases such as multiple agencies (e.g., police, custom, news agencies, hospitals, insurances) providing and consuming information anonymously. Certainly, a publish/subscribe interaction model is suitable for such cases considering that the number of publishers and subscribers may increase over time, subscribers need information whenever they require independently of publishers. Even though certain standards for publish/subscribe interaction exist (e.g., WS-Notification [3]), their adoption falls short.

Many organizations that participate in such processes develop proprietary XML schemas to address individual needs, for instance, a particular data model associated with a business process or an organizational structure. Such schemas may contain business critical information that needs to be protected. In addition, instances of these schemas (i.e., Enterprise XML) might be routed by untrusted intermediaries and through insecure communication channels which also asks for content confidentiality and integrity.

Regarding the actual service interface, communication parties need to agree on a certain data model (schema) which may evolve over time (e.g., due to changes in one party's organization, for instance after a merger); existing data exchanges with peers should however be maintained. We claim that, although local data models may differ from one organization to another or vary with time, the underlying semantics (represented by enterprise XML fragments) constitute a more stable and interoperable interface between organizations. Semantic web languages like RDF [2] and OWL [1] make it possible to share an ontology describing the business domain data model, independently from individual XML data models (i.e., schema) yet can still be mapped to instances of XML schemas. To address security requirements, authorization policies on the semantic level, i.e., ontology, should be supported. Such a secure exchange of documents can be achieved through the separate encryption of possibly each document node with a key associated with the semantics of the node (i.e., ontology concept) and computed in a distributed fashion by the publishers and subscribers. In this approach, an authorization on an ontology concept triggers a secret key computation resulting into granting authorizations to multiple XML documents or portions thereof.

In Section 2, a set of requirements is elicited based on an example scenario. Section 3 describes a brief solution and preliminaries of the publish/subscribe model which is described in detail in Section 4. This includes a family of protocols (i.e., publishing, (un)subscription, delivery) to selectively route and delivery of XML content from publishers to authorized subscribers. Subscriber-end processing upon receipt of such content is illustrated in Section 5. A comparison with the related work is provided in Section 6. Section 7 finally concludes the paper with future work.

2. An Example Scenario and Requirements

Previous research effort [6], [8]-[10], [12]-[18] targets some of these mentioned issues, namely confidentiality and integrity of documents in a client-server architecture. However, government or industry use cases imply that a separation of XML dissemination from its actual data

representation is required. The following example motivates this scenario.

2.1 A Cross Border Crime Scenario

1. Consider a car driver holding a license plate of EU country A while driving in a motorway of country B exceeds the speed limit and follows through a car accident.

2. The motorway police (MP) and community police (CP) of country B rush into the spot and find the driver badly injured.

3. The CP immediately calls an ambulance of a local hospital (LH) for emergency medical help.

4. Local news agencies (NA) rush there to cover news which will be then distributed to other news agencies including foreign agencies.

5. MP notifies the accident to the corresponding authority of country A (PA) and requests for more information of the driver.

6. PA looks up into its database and finds previous motor accident history of the driver that occurred in other countries and informs those to MP.

7. MP consolidates all accident histories and then files a case in the local court (LC).

8. To resolve cost claims by the driver, after few months of this accident, car and medical insurance people (IN) require information regarding car details and medical expenses that can be provided by the car seller and LH respectively.

9. Even after several months, the case will be in the court requiring all details that can be provided by MP, CP, NA and PA.

10. LC finds facts and evidence from those details for judicial proceeding.

Sensitive Information: Such cross border incidents also require secure information exchange among heterogeneous IT systems deployed in different EU countries as identified by the EU project R4eGov [7]. Parties providing information have individual XML schemas to comply with their organization and country specific policy, regulations which may not allow them to exchange the full documents or portions to everybody. For instance, CP may not disclose the driver's license plate, her social security number and insurance information to NA due to legal bindings or her bank information to IN as it can be misused. MP will not disclose driver's exceeded speed limit and her previous accident records to LH as those are not required for medical attention, however these could be important for court (LC) proceedings implying that the LC will be authorized to receive those.

Evolving Data Model: As can be imagined, the number of parties may increase over time. For instance, PA and other countries provide history of accident reports of the driver and intelligence agency (IA) may get involved in the case. As PA is from another country and IA is already protecting its data model they do not have any common data format with others. After a while, IA takes over the case from MP and CP as the alleged driver is considered to be a national threat. Proprietary data exchange formats with PA, other countries and due to the take over of IA, MP and CP (including IA) require restructuring their data models which in turn invalidate the existing data exchanges between providers and consumers. However, the information must be available whenever required for later court (LC) proceedings and police cases irrespective of its publication time.

Neither point to point nor a centralized publish/subscribe architecture provides a scalable and highly available XML content distribution system. We propose a Publish/Subscribe based document dissemination system where document producers publish documents and subscribers consume those independently of each other. The dissemination system is an intermediate layer composed of distributed disseminators that selectively route XML content along the dissemination topology and perform selective delivery, i.e., filtering, to the authorized subscribers according to the authorization policy of the publishers.

2.2 Requirements

Important requirements we address in our system are:

1. Loosely Coupled Document Exchange: Actors being document providers or consumers must be able to exchange independently of each other.

This is addressed by first employing a publish/subscribe based dissemination network that forms a topology of disseminators and decouples publishers and subscribers. Then, by inducing a dissemination topology using a shared domain ontology that lets publishers' data models to evolve independently.

3. Confidentiality of schema information: An information schema (e.g., XML schema) represents a valuable asset in itself (e.g., information about organizational structures or business processes can be derived from the schema) and as such needs to be confidential.

This is fulfilled by using a domain ontology describing the semantics of data exchanged as the interface among organizations as opposed to sharing concrete schemas.

4. Confidentiality of information: Access to documents/document portions shall be limited to authorized communication partners, i.e., the respective publisher and authorized subscribers.

This is addressed by (a) the encryption of published XML nodes, supported by a distributed key management and (b) an ontology-based document authorization scheme that supports fine-grained

document access control and dissemination on semantic level.

5. Integrity of information: Documents must not be altered during transit.

This is achieved by a special XML encoding method based on our earlier work [20] for published document nodes that allows subscribers to verify integrity of the received documents.

Some of these requirements have been addressed in our previous work ([20], [21]): in [20], we developed a publish/subscribe based centralized XML content distribution technique relying on a that focuses on ontology-based authorization checking by leveraging the inference power of ontology and granting access of selective XML content to authorized users. There we also developed an encryption enabled XML parsing technique called *encrypted breadth first order labeling* (EBOL) to protect the confidentiality and integrity of the XML content and its semantics. The EBOL-based encoding annotates semantic information, such as, ontology concept, publisher id, document id, node id to the original XML document portions and attaches security metadata, such as, hash values of the XML content along with other cryptographic values. This annotation ensures that each encoded XML content is uniquely identifiable but the content is only readable to the respective publisher and authorized subscribers who do not need to know each other. Middleware reads the annotations and can extract and deliver appropriate encrypted XML content to legitimate peers. To compute a secret key associated with an ontology concept by the publishers and subscribers our distributed key management technique is leveraged [21]. This technique allows a group of users to compute a common key independently of each other which is used to encrypt the original encoded XML document portions. This paper focuses on selective delivery techniques of semantically related XML content through a publish/subscribe infrastructure of distributed XML content disseminators.

3. Solution Overview

3.1 System Overview

The dissemination system distinguishes three different actors (see Fig 1).

a. Document producers publish encrypted and encoded XML document portions that represent ontology concepts. Publishers take charge of individual XML document data models and policies over it. They also define a mapping relation of the ontology concepts into their individual data model as shown in Fig 2,

b. Document subscribers being the end users receive these XML document portions,

c. A disseminator is a piece of software running either in intra- or inter-enterprise boundaries and thus is distributed and manages subscriptions.

While access control is realized by encryption, disseminators check the authorization policies on behalf of the publishers and realize the actual XML content transmission from publishers to subscribers. Disseminators, however, should not be able to read document content.

We consider disseminators are honest but curious and as such malicious activities (i.e., integrity violation) may occur in a disseminator or in the communication channels between all actors (detailed in [20]).

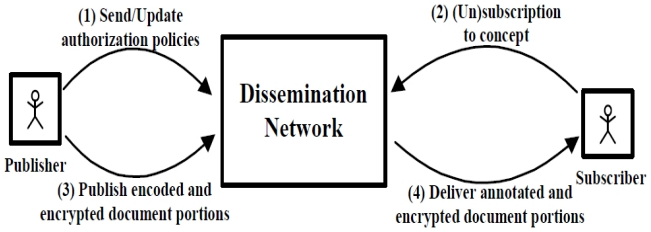

Figure 1. A publish/subscribe model of content-driven XML dissemination.

3.2 Interaction Overview

The definition (or nomination) of a domain ontology is the prerequisite for any interaction between the system actors. Such an ontology can be publicly shared for interested peers. The system actors interact as follows (see Fig 2):

1. Prior to the first document publication, a publisher needs to provide authorization policies that determine user authorizations and which will be enforced by the dissemination network. Publishers may also issue inference rules describing constraints (e.g., unsubscription) over their policies.

2. An end user sends a subscription request for a concept with valid credentials (e.g., public key certificate) to a disseminator which in turn checks associated policies (provided by the publishers) and trigger the computation of a secret key for every group of authorized subscribers to the same concept [21]. Unsubscription might occur at the user's request or be forced by the disseminator (e.g., if the user credentials expire or if authorization policies change).

3. The publisher of an XML document annotates XML document nodes with its conceptual information [20], encrypts the nodes in a stipulated granularity with the secret key computed associated with the concept, and sends the encrypted nodes along with their semantic annotations to the disseminators.

4. Disseminators follow a protocol (i.e., immediate delivery) triggered by publishing of documents in order to route the encrypted document portions selectively to other disseminators to the disseminators. Disseminators extract the relevant encrypted nodes by matching subscriber's authorized concepts with the annotations and thus cannot read the document content. Another protocol (i.e., catch-up delivery) triggered by subscription of concepts is

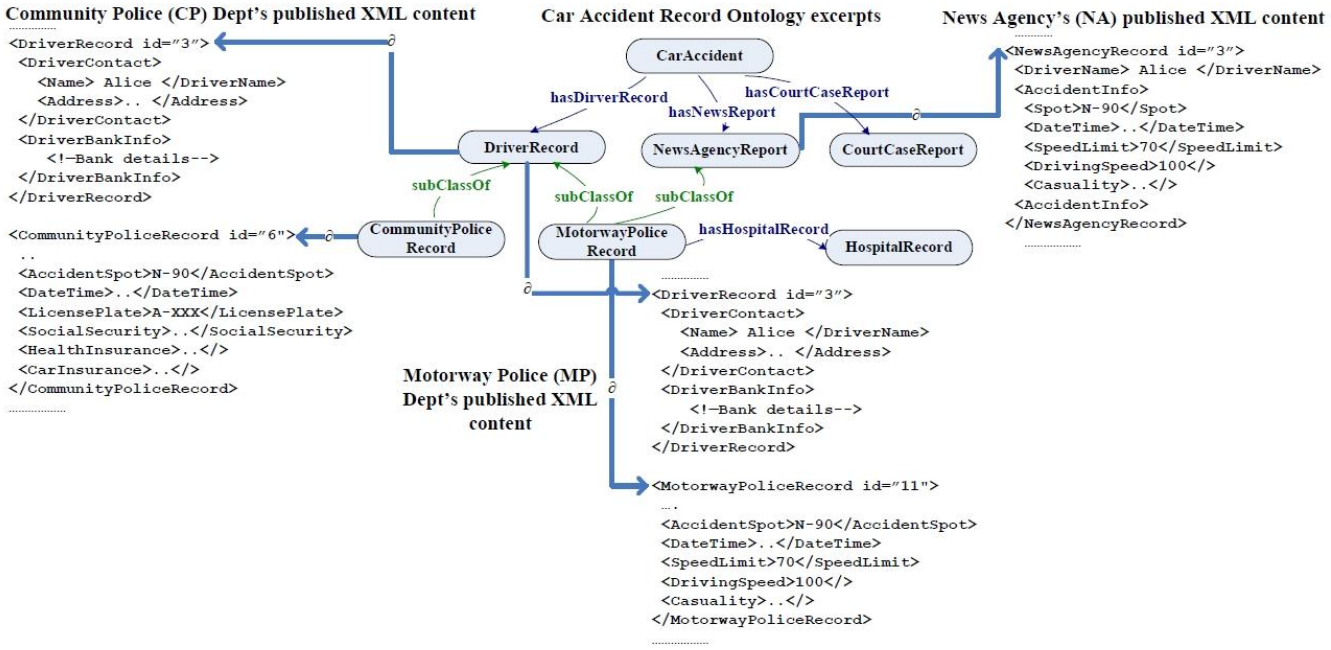

Figure 2. A car accident ontology excerpt. Community police (CP) and motorway police (MP) map `'DriverRecord'` concept to their individual document portion rooted at `<DriverRecord>`. The concepts 'CommunityPoliceRecord', 'MotorwayPoliceRecord' and 'NewsAgencyReport' are mapped to the corresponding XML data model excerpts (i.e., `<CommunityPoliceRecord>`, `<MotorwayPoliceRecord>`, `<NewsAgencyRecord>`) of CP, MP, and NA respectively.

designed to deliver the selective XML content to all authorized subscribers (Section 4).

5. The recipient can verify (not shown in the Fig 1) the received XML content by decoding the EBOL-based encoding, both semantically and structurally, in a verification phase which is detailed in [20].

The dissemination method leverages the publicly shared ontology that models all relevant business domain entities including their relationships. Fig 2 sketches a car accident ontology excerpt motivated by the example before. This method is designed to enable loosely document exchanges between publishers and subscribers. This is achieved by an ontology-based dissemination topology that utilizes the following notions (i.e., concept, *Concept Containment* and *Maximum Conceptual Block*) described below.

3.3 Preliminaries

A concept C_i is an abstraction of a physical or logical thing and can be communicated among peers. Ontology is a shared set of concepts of a domain and defined by the notions of class, subclass, properties representing concepts and their relationships using OWL [1] as illustrated in Fig 2. In our system ontology represents the semantics of the XML content that will be exchanged through multiple vocabularies for the same concept for instance.

Concept Containment is a succession of class relationships (i.e., sub class hierarchy or property) from a given concept C_i to C_j denoted as $C_i \prec C_j$. A *Maximum Conceptual Block* for a concept C_i is the set of all concepts that are reachable by *Concept Containment* from C_i.

Example: Fig 2 shows an ontology excerpt of a car accident $\boldsymbol{C}$ = {`CarAccident, DriverRecord, NewsAgencryReport, CourtCaseReport, CommunityPoliceRecord, MotorwayPoliceRecord, HospitalRecord`}. `DriverRecord` contains `CommunityPoliceRecord` and `MotorwayPoliceRecord`, i.e., `DriverRecord` $\prec$ `CommunityPoliceRecord`, `DriverRecord` $\prec$ `MotorwayPoliceRecord`.

Mapping to XML Data Model: A document publisher maps an ontology concept to its individual disjoint document portions as illustrated in Fig 2. Note that, Fig 2 shows corresponding mapping to instance level for illustration. However, mapping can be defined in the schema level also.

Lemma: Maximum conceptual blocks are always monotonically decreasing.

Proof: Let C_i and C_j be two concepts such that C_i contains C_j; let M_i^c and M_j^c be the *Maximum Conceptual Blocks* for C_i and C_j respectively. As C_i contains C_j the number of classes reachable from C_i is always more than that of C_j. So $M_j^c \subset M_i^c$. Transitively for any concept C_k such that $C_j \prec C_k$ then $M_k^c \subset M_j^c$.

3.3 Combining Documents upon Receipt

As different XML document portions can be associated with a concept, eventually an authorized subscriber of that concept may receive multiple document portions, possibly with different XML vocabularies. Essentially, the user needs an adaption mechanism possibly with its own XML data model to understand and process the received XML content. In this regard, the customized business rules (based on the shared ontology) depending on the recipient's necessity can also be used (see Section 5). This is illustrated below with respect to the cross border crime scenario.

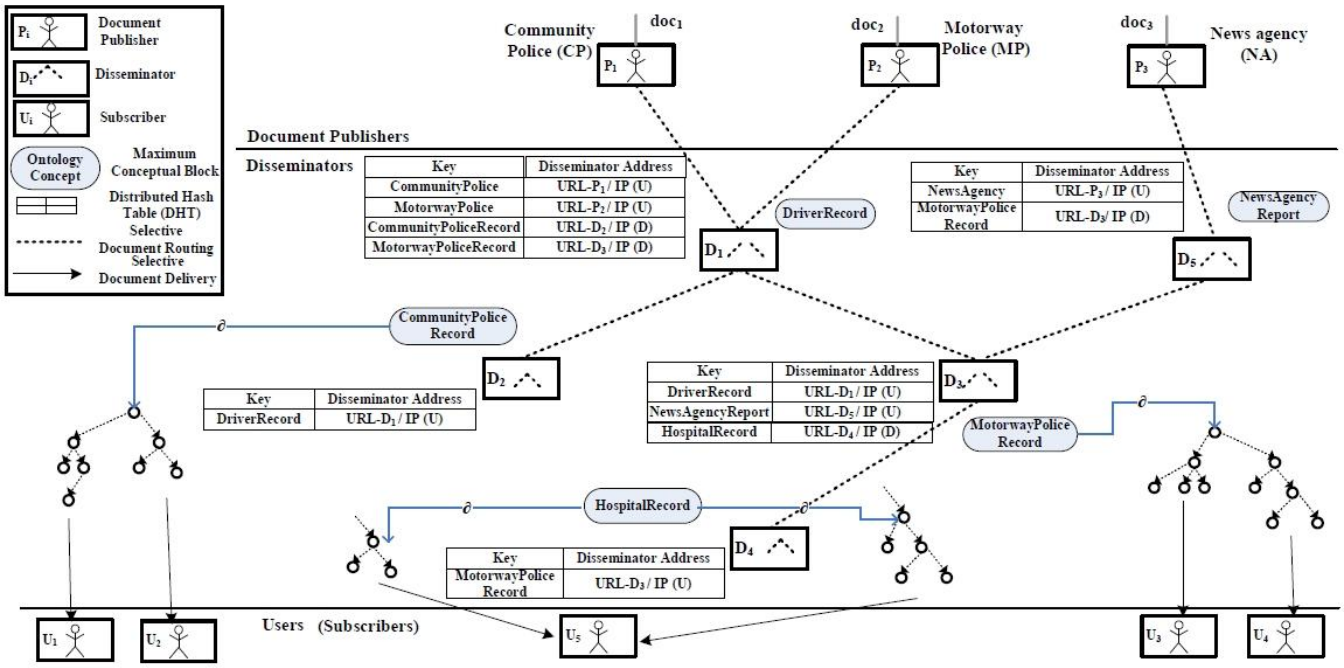

Figure 3. Publish/Subscribe infrastructure for XML content distribution.

LC receives various details related to the car accident in variety of formats and vocabularies from CP, MP, NAs, IA and possibly PA. Then LC determines facts and evidences from the received documents by correlating, combining, comparing different document portions based on some customized business rules. For instance, LC will consider document portions related to the car accident, i.e., police record of CP and MP, news reports of NA, accident history of PA; in order to give a verdict. One rule would be: if some driver with a license plate number X is found to drive over the speed limit on a motorway in the police record of MP with the same license plate number and an NA has a news report confirming the date and time of the incident, then the alleged driver is subject to punishment with a fine.

3.4 Ontology-based Authorization

A policy of a publisher is described over the shared ontology as illustrated in Fig 4 that shows two policies of the publishers, P_1 (i.e., CP) and P_2 (i.e., MP) of Fig 2. P_1's policy is to allow access of XML content associated to the concept `'DriverRecord'` to a user having credential 'Cred1'. R_1 and R_2 are two inference rules of P_1 and P_2 respectively, where R_1: an authorized user of the concept `'DriverRecord'` is also allowed to access all the contained concepts of `'DriverRecord'` and R_2: an authorized user can access the concept `'HospitalRecord'` until the end of judicial process (i.e., after that the user needs to be unsubscribed).

XML publisher	Subscriber_credentials	Concept, C_i	Rule, R
P_1	Cred1	DriverRecord	R_1
P_2	Cred2	HospitalRecord	R_2

Figure 4. Ontology-based authorization policy.

4. Publish/Subscribe Dissemination Model

This section describes the publish/subscribe scheme by elaborating the roles of actors, (un)subscription of concepts, publishing, selective delivery protocols of XML content, and ontology-based dissemination topology (see Fig 3).

4.1 Initialization

The intermediate layer of disseminators of Fig 3 is introduced to ensure loosely coupled document exchange in selective fashion between publishers and subscribers.

One *Maximum Conceptual Block* is assigned to a disseminator for routing and delivery of mapped XML content. Following the monotonically decreasing property of *Maximum Conceptual Blocks* of the concepts the responsibility of a disseminator for storing, routing and delivering of XML content can also be determined. Thus disseminators having more storage and computing capability can be assigned to disseminate more concepts than others. In case of equally able disseminators assignment can be random.

Each disseminator (including publishers) maintains a distributed hash table (DHT) where the key fields and the values are the concepts representing the *Maximum Conceptual Block* and references (i.e., URL/IP) of the disseminators respectively (see Fig 3). The ordering of the key fields is determined as follows:

1. We assign each *Maximum Conceptual Block* in the key fields in monotonically decreasing fashion.
2. We assign the reference addresses of the next disseminators in the value fields for each such key.

Let D_i, D_j be two disseminators that disseminate two *Maximum Conceptual Blocks* represented by concepts C_i, C_j respectively. We further define uplink disseminators and downlink disseminators as follows:

1. If $C_i \prec C_j$ holds then D_i is an uplink (U) disseminator of D_j and D_j is a downlink (D) disseminator of D_i.

2. D_i puts C_j as downlink such that $C_i \prec C_j$ and C_k as uplink such that $C_k \prec C_i$ as its key and corresponding references as value fields in its DHT respectively.

4.2 Dissemination Topology

The disseminators form a directed acyclic graph (DAG) topology based on *Concept Containment* where document publishers comprise multiple starting points (roots) in the dissemination. Fig 3 shows such a dissemination topology with three publishers (i.e., P_1, P_2 and P_3) as roots.

Let D_k be any disseminator reachable from D_i by following a dissemination path $D_i \rightarrow , ... , \rightarrow D_k$. C_i is the *Maximum Conceptual Block* at D_i if and only if D_i or any disseminator D_k has registered only the users who have authorizations to the concepts C_i or any of its contained concept C_i. Consequently D_i can deliver the encoded and encrypted XML nodes to a set of subscribers such that none of them has access or has subscribed to a concept $C_m \in \boldsymbol{C}$, where $C_m \prec C_i$. In effect, the disseminator D_i disseminates only the mapped XML nodes of C_i or any C_j such that $C_i \prec C_j$. In Fig 4, the disseminator D_3 has `'MotorwayPoliceRecord'` as the *Maximum Conceptual Block* for which user 3 and user 4 have collectively registered.

In the following, we elaborate on the protocols which make use of an annotation element *'content signature*' [20] and two functions. A *'content signature'* is comprised of XML node's structural and conceptual information. The function *servedlist(d)* returns the set of concepts represented by the *Maximum Conceptual Block* in the DHT of the associated disseminator *d*. The function *authlist(u)* returns a set of *'content signatures'* which is used by a subscriber *u* as a means to verify the received XML content.

4.3 Publishing

For a new instance of a document, a publisher annotates and encrypts the mapped document portions and finally sends those to its downlink disseminators.

For selective routing of annotated and encrypted XML nodes, a disseminator D_r filters by matching conceptual annotations of the received content with its DHT's *Maximum Conceptual Block* assignment (see Fig 3).

1. ***Determine served content:*** Upon receipt of annotated and encrypted XML content associated to a set of concepts (e.g., C_i), D_r first determines if some or all content are already added to its *servedlist(*D_r*)*, i.e., $\exists C_i \in$ *servedlist(*D_r*)*. If some are not added then it determines the content that it can serve from the rest by matching conceptual annotations of the content with the concepts of its assigned *Maximum Conceptual Block*. The determined XML content are then added to its *servedlist(*D_r*)*.

2. ***Filter content:*** D_r separates XML content that are not added in step 1 associated to the concepts not in its *Maximum Conceptual Block.*

3. ***Route content:*** D_r then sends XML content of step 2 to either downlink or uplink disseminators. The content associated to concepts C_k such that $C_k \prec C_i$ are sent to uplink disseminators and concepts C_j such that $C_i \prec C_j$ are sent to downlink disseminators.

The end result of the publishing is that the published annotated and encrypted document nodes are selectively routed to disseminators who store and can route further or deliver those to other disseminators and authorized subscribers.

4.4 Subscription

1. ***Subscribe concepts:*** User *u* sends a subscription request (together with its credentials) for a set of concepts to a disseminator D_r. Upon receipt of such a request, D_r determines the authorizations of the user (*authlist(u)*) based on the publishers' policy as described in Section 3.

2. ***Register:*** If all authorized concepts of *authlist(u)* are contained in the list of served concepts of D_r, (i.e., *authlist(u)* $\in$ *servedlist(*D_r*)*, then D_r registers the user, *u*, successfully as an authorized subscriber by sending the associated *'content signatures'* and the protocol ends.

3. ***Forward subscription request:*** Otherwise concepts of step 1 include at least one concept, $C_k \in$ *authlist(u)* such that $C_k \notin$ *servedlist(*D_r*)*. If C_k contains D_r 's served concepts, i.e., $C_k \prec \forall C_i \in$ *servedlist(*D_r*)*, then D_r sends the request to the uplink disseminators. Otherwise, D_r sends the request to the downlink disseminators.

4. ***Route content signatures:*** After receiving a request for C_k from D_r, a disseminator D_m checks if there exists either a $C_k \in$ *servedlist(*D_m*)* of step 3 or a concept containment relation ($C_m \in$ *servedlist(*D_m*)*) $\prec C_k$. If so, D_m returns the associated *'content signatures'* of C_k with success as a response to D_r, else D_m recursively performs the same step 3 for other disseminators in its DHT.

5. ***Update and register:*** After receiving the responses possibly from several disseminators, D_r selects a sending disseminator using a selection policy described below, updates its list of served *'content signature'* by adding the new one and notifies the disseminators accordingly. Now, the disseminator D_r is able to register the user and sends a response by sending the *'content signatures'* to it.

Selection policy: A selection policy is based on the notion of *'concept distance'* aiming at minimizing the hops required to route the XML content: Let C_i, C_j be two concepts identified by $O_1.C_i$, $O_2.C_j$, where $O_i \in [1,2]$ are two path expressions and $|O_i|$ denotes the number of hops required as entailed by *Concept Containment*. Then concept distance between C_i and C_j is defined as $||O_i|-|O_j||$. The receiving

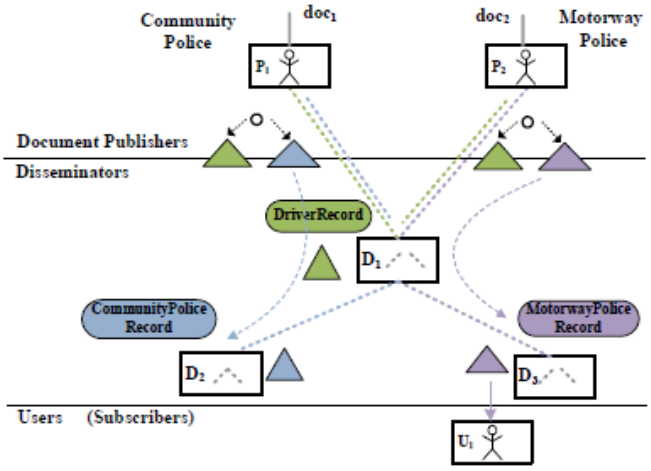

Figure 5. **Publishing** - CP and MP both publish individual document portions associated to the concept `'DriverRecord'` of which only one copy is stored in the disseminator D_1. Published document portions associated to the concepts `'CommunityPolicRecord'` and `'MotorwayPoliceRecord'` are filtered selectively to D_2 and D_3 respectively according to their *Maximum Conceptual Block* assignment. **Immediate delivery** - D_3 immediately delivers the document portion associated to the concept `'MotorwayPoliceRecord'` to the authorized subscriber U_1.

disseminator chooses the sending disseminator with the smallest *'concept distance'* from itself.

4.5 Selective Delivery

Delivery of selected encrypted XML content by a disseminator D_r occurs after either a newly published document or a successful subscription. The method enables avoiding multiple routing and delivery of documents.

Immediate delivery after publishing: For each subscribed user, u , D_r performs the following steps in order (Fig 5).

1. ***Separate allowed concepts:*** find all $C_i \in$ *authlist(u)* such that $C_i \in$ *servedlist*(D_r).
2. ***Determine allowed nodes:*** match concepts of *authlist(u)* with annotation of the encrypted content.
3. ***Extract content:*** extract associated encrypted and encoded XML nodes.
4. ***Deliver content:*** Finally, send the XML nodes extracted in step 3 to the user u.

For a successful subscription, if a disseminator does not host XML content for subscribed concepts, needs to fetch relevant content.

Catch-up delivery after subscription: If a subscribed concept C_k of a user is part of *servedlist*(D_r) then D_r can execute same steps of delivery after publishing as it already possesses the respective content. Otherwise the subscribed concept C_k is not in its *servedlist*(D_r), then it needs to fetch the content from other disseminators. Now, D_r can fetch the XML content directly from the disseminator D_k selected by the selection policy during subscription by requesting

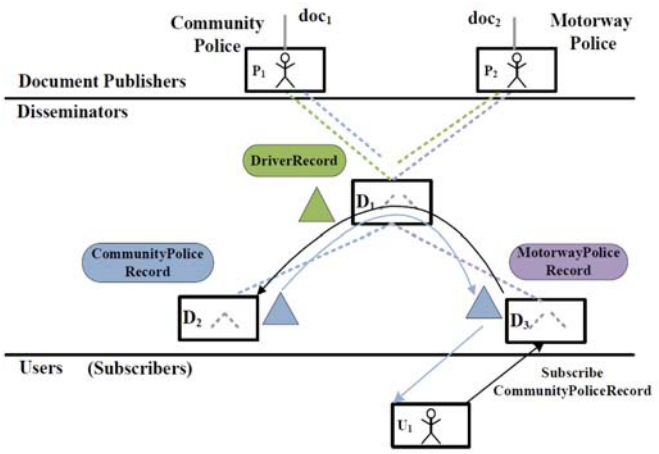

Figure 6. D_3 does not host the associated document portion of the concept `'CommunityPoliceRecord'` subscribed by the user U_1. D_3 then fetches those from D_2 through D_1 which in turn updates its DHT by adding the concept `'MotorwayPoliceRecord'` and the reference of D_3 for later potential routing. D_3 then delivers the content to U_1.

desired concepts. In case D_k does not host the content, a fetching protocol similar to the steps 3-5 of subscription is followed (Fig 6). The fetching protocol differs from the subscription in that it fetches the actual content and updates its DHT whereas the other only retrieves the *'content signatures'*.

1. ***Forward subscription request:*** If subscribed concept C_k contains D_r's served concepts, i.e., $C_k \prec \forall C_i \in$ *servedlist*(D_r), then D_r sends the request to its uplink disseminators. Otherwise, D_r sends the request to its downlink disseminators.

2. ***Update DHT and route content:*** After receiving a request for C_k from D_r, a disseminator D_m checks if there exists either a $C_k \in$ *servedlist*(D_r) or a concept containment relation $C_m \in$ *servedlist*(D_m)) $\prec C_k$. If so, D_m adds D_r's reference as the value field in its DHT for the concept C_k and returns the mapped encoded and encrypted XML nodes of C_k with success as a response to D_r, else D_m recursively performs the same step 1 for other disseminators in its DHT. The added reference allows, D_m to route later published XML content associated to the concept C_k to D_r.

3. ***Update and deliver:*** After receiving the first response from a disseminator, the disseminator D_r updates its *servedlist*(D_r) by adding the newly received content and adds the corresponding disseminator reference as the value field of its DHT for the concept C_k. Now, the disseminator D_r is able to deliver the subscribed content to the user u.

4.6 Unsubscription

As mentioned before, we rely on our distributed key agreement scheme [21] that is required to be executed by a group of subscribers in a subscription phase in order to compute the shared key and thus to protect the confidentiality of the XML content between the publishers and subscribers. While a new secret key should be

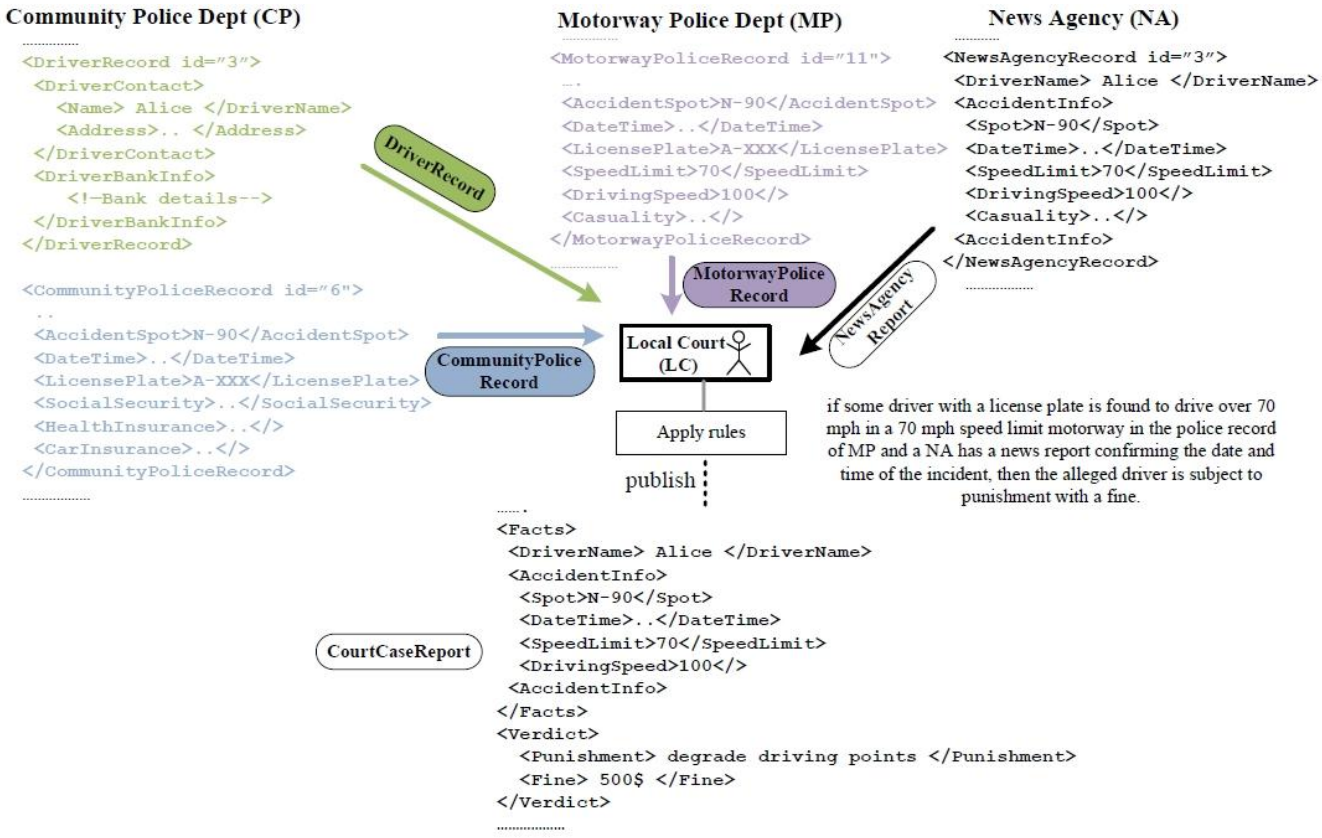

Figure 7. Applying customizing rules in order to process various document portions

computed by a group of subscribers in the event of a new subscription for the same concept, the existing secret key can be used in case of a unsubscription of an existing user of the same group. This is because for a successful unsubscription the responsible disseminator simply stops sending the associated XML content to that user. For an unsubscription of a concept C_i of a user *u*, the disseminator D_r performs the following steps:

1. ***Determine content:*** Determines the authorized XML content based on the authorizations of *u* for the concept C_i.

2. ***Unregister and stop delivery:*** Sends a successful unsubscription response to *u* and stops sending encoded and encrypted XML content of step 1 to *u*. Then it checks whether any other authorized user has currently subscribed for the same concept C_i. If no then D_r also forwards the unsubscription request for the concept C_i to other disseminators in its DHT.

3. ***Unregister and stop routing:*** Upon receipt of an unsubscription request for concept C_i from another disseminator, i.e., D_r, D_i sends a response back to D_r stating that unsubscription is successful and stops routing encoded and encrypted XML content associated to C_i to D_r. D_i further checks whether any other authorized user or disseminator has currently subscribed for the same concept C_i. If no, then it performs similar steps as in step 2.

5. Subscriber-end Document Processing

Upon receipt of various encoded and encrypted XML nodes, an authorized subscriber decrypts and then decodes those to check precise integrity violation for instance any node deletion, node swapping, order change, node update [20]. The subscriber then processes those contents depending on its role in the dissemination. For instance, the local court (LC) subscribes concepts `'MotorwayPoliceRecord'`, `'CommunityPoliceRecord'` and `'NewsAgencyReport'` to receive published content of MP, CP and NA respectively to determine facts and give a verdict. LC may also publish its verdict later on. In the following, we illustrate on such three scenarios:

5.1 Sharing Partial XML Schema Model

A publisher and a subscriber may share the same XML schema model, for instance CP and MP departments have a bilateral agreement to have the same partial schema for a `<DriverRecord>`. As such, the MP receiving multiple instances of CP `<DriverRecord>` with different `id` represents multiple driver information and vice versa.

5.2 Authorizing Partial Schema Mapping

This scenario is similar to the previous one except there is no bilateral agreement between any two parties. However, a publisher can also publish the mapping from a concept to its schema elements as part of XML node's encoding so as to be accessible by the authorized subscriber only. Such a

mapping of each node of a document portion accompanied by a conceptual annotation provides a clear understanding of semantics and structure of the document portions received.

5.3 Applying Customized Rules

We illustrate this scenario in Fig 7. Let LC be the authorized subscriber of the content associated to the concepts `'DriverRecord'`, `'CommunityPoliceRecord'`, `'MotorwayPoliceRecord'` and `'NewsAgencyReport'`. Considering no integrity violation occurs, LC applies the following general steps:

1. **Document portion identification rule:** is the initial step performed over the encoding of document nodes (i.e., *'node identifier'* of an XML node [20]) to identify the desired nodes for further processing. The rule is: determine all nodes annotated with desired concepts and possibly filter those further by separating nodes for desired publishers.

2. **Business rules:** is a set of rules applied over the document nodes from step 1 in order to perform an application specific processing.

LC identifies document portions associated to the concepts `'DriverRecord'`, `'CommunityPoliceRecord'`, `'MotorwayPoliceRecord'` and `'NewsAgencyReport'`. Further, it separates the `<DriverRecord>` and `<CommunityPoliceRecord>` of CP, `<MotorwayPoliceRecord>` of MP and `<NewsAgencyRecord>` of NA. LC has two business rules for: ‘document correlation’ and ‘document composition’ where the first determines the related document instances and the other builds LC's own document portion associated to the concept `'CourtCaseReport'` that it may publish. In particular:

1. Document correlation: find some driver name having same license plate no in some CP and MP record that states a driving speed > 70 mph in the motorway N-90 on the same date and time. If the NA report also confirms the same driver then LC considers those as facts and gives its verdict: degrading driving points with 500 USD fine.
2. A simplified document composition rule: build a document with `<Facts>` and `<Verdict>` accordingly.

If none of the mentioned scenarios applies, the subscriber-end processing looses precise semantics of different XML vocabularies despite each received document portion is annotated with a concept. In such cases, schema matching solutions [5], [11], [22] can be used for mostly structural matching given that full schemas of all parties are known (i.e., against our motivation). As such, we suggest a technique to build up a customized recipient schema based on shared ontology that can be used for further processing (detailed in [19]).

6. Related Work

There has been remarkable progress in recent years to address access control issues focusing on XML structure [6],[8]-[10],[15]-[17]. The basic model of this work is a typical request response paradigm in a client-server architecture. Instead, this paper proposes a publish/subscribe model for semantic based fine-grained document dissemination.

The work of [13], [14] specifies policy on XML data structure and focuses on dissemination of XML data exploiting XML data structural properties. However, our approach is fundamentally different as policy specification is on domain concepts and selective dissemination is performed based on the semantics captured as ontology concepts. If the local XML structure changes, solutions of [13] and [14] require associated routing topology to be changed. For the same reason, a subscriber needs to have prior knowledge of the routing structure as the router can not fetch any content which is not hosted currently. In contrast, our disseminator topology is based on domain ontology concepts relationships which are independent of any XML structural change. Moreover, subscribers in our settings do not need to have any prior knowledge of disseminator's capability as disseminator can fetch the desired content for an authorized subscriber.

In [20], we proposed a centralized publish/subscribe middleware which is able to perform selective XML content delivery based on a shared ontology. There we suggest semantic queries over domain concepts to compute a set of candidate concepts and over publishers' policies to check evolving policies for a subscriber. These queries can also be used for similar purpose in each disseminator in our proposed distributed publish/subscribe network.

The work in [12], [18] proposes an ontology based access control for XML documents having variant schemas and semantically related documents respectively. However, both consider issues related neither to dissemination of semantically related documents nor to integrity and confidentiality of documents at all.

7. Conclusion and Future Work

This paper shows how business domain ontology can be used as a stable interface among interacting peers while ensuring individual confidential and evolving data model. It introduces a publish/subscribe model for a loosely coupled dissemination of semantically equivalent XML content to the authorized subscribers. This model describes an ontology-based dissemination topology and a family of protocols for selective XML dissemination. This also illustrates document processing scenarios on the subscriber end. While this model relies on a set of disseminators to check the access control policies, the confidentiality and integrity of the disseminated content is assured by a secret key computed in distributed fashion and special encoding method respectively.

We are currently investigating how to extend semantic based selective document dissemination to a workflow context in

which a document receipt may trigger processing of tasks and the generation of additional documents.

References

[1] OWL Web Ontology Language Overview, http://www.w3.org/tr/owl-features/.

[2] Resource Description Framework (RDF), http://www.w3.org/rdf/.

[3] Web Services Notification, http://www.oasis-open.org/committees/tchome.php? wgabbrev=wsn

[4] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order preserving encryption for numeric data. In SIGMOD '04: Proceedings of the 2004 ACM SIGMOD interna tional conference on Management of data, pages 563-574, New York, NY, USA, 2004. ACM.

[5] Y. An, A. Borgida, and J. Mylopoulos. Constructing Complex Semantic Mappings Between XML Data and Ontologies. In The Semantic Web ISWC 2005, pages 563-574. Springer Berlin / Heidelberg, 2005

[6] W.-C. L. Bo Luo, Dongwon Lee and P. Liu. A Flexible Framework for Architecting XML Access Control Enforcement Mechanisms, volume Volume 3178/2004 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, December 2004.

[7] A. D. A. Boujraf and M. Noble. Towards e-Administration in the Large (R4eGov). Deliverable WP3-D7, 2007. EU IP R4eGov.

[8] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. Fine Grained Access Control for Soap E-services. In WWW '01: Proceedings of the 10th interna tional conference on World Wide Web, pages 504-513, New York, NY, USA, 2001. ACM.

[9] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. A Fine grained Access Control System for XML Documents. ACM Trans. Inf. Syst. Secur 5(2):169-202, 2002.

[10] W. Fan, C.-Y. Chan, and M. Garofalakis. Secure XML Querying With Security Views. In SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data, pages 587-598, New York, NY, USA, 2004. ACM Press.

[11] M. Ferdinand, C. Zirpins, and D. Trastour. Lifting XML Schema to OWL. In Web Engineering, pages 563-574. Springer Berlin / Heidelberg, 2004.

[12] A. Jain, D. Wijesekera, A. Singhal, and B. Thuraisingham. Semantic-Aware Data Protection in Web Services, Proceedings of IEEE Workshop on Web Services Security held in Berkeley, CA, May 2006, 2006.

[13] A. Kundu and E. Bertino. A new model for secure dissemination of xml content. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 38(3):292-301, May 2008.

[14] A. Kundu and B. Elisa. Secure Dissemination of XML Content Using Structure- based Routing. In EDOC '06: Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference, pages 153-164, Washington, DC, USA, 2006. IEEE Computer Society.

[15] G. Kuper, F. Massacci, and N. Rassadko. Generalized XML Security Views. In SACMAT '05: Proceedings of the tenth ACM symposium on Access control models and technologies, pages 77-84, New York, NY, USA, 2005. ACM Press.

[16] Miklau and D. Suciu. Controlling Access to Published Data Using Cryptogra phy. In VLDB, pages 898-909, 2003.

[17] M. Murata, A. Tozawa, M. Kudo, and S. Hada. XML Access Control Using Static Analysis. In CCS '03: Proceedings of the 10th ACM conference on Computer and communications security, pages 73-84, New York, NY, USA, 2003. ACM Press.

[18] V. Parmar, H. Shi, and S.-S. Chen. XML Access Control for Semantically Related XML Documents. Proceedings of the 36th Annual Hawaii International Conference on System Sciences, pages 10 pp. Jan. 2003.

[19] M. A. Rahaman, H. Plate, Y. Roudier, and A. Schaad. Content Driven Secure and Selective XML Dissemination. Technical Report RR-09-219, Eurecom, 05 2009.

[20] M. A. Rahaman, Y. Roudier, P. Miseldine, and A. Schaad. Ontology-based Secure XML Content Distribution. In IFIP SEC 2009, 24th International Information Security Conference, May 18-20, 2009, Pafos, Cyprus, May 2009.

[21] M. A. Rahaman, Y. Roudier, and A. Schaad. Distributed Access Control For XML Document Centric Collaborations. In IEEE, editor, The 12th IEEE Enterprise Computing Conference (EDOC 2008), September 2008.

[22] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema match ing. VLDB Journal: Very Large Data Bases, 10(4):334-350, 2001.

[23] M. R. X. F. Wang. Defending Against Denial-of-service Attacks with Puzzle Auctions. In proceeding Symposium on of Security and Privacy, USA, 2003.

Author Biographies

Mohammad Ashiqur Rahaman has been working for INRIA-Paris as a R&D Engineer since September, 2009, in the ARLES project team. He previously worked for SAP as a Research Associate. He is a PhD candidate at EURECOM Institute and will defend his thesis in early 2010. His research interests span SOA-based workflow technologies and he currently is working on modeling and designing an architecture for IT and embedded systems to enable business workflow executions.

Yves Roudier is an Assistant Professor in the Network and Security department of EURECOM Institute. His main research interests are secure P2P storage, ITS security, ubiquitous computing security, SOA security, and research on the introduction of security through middleware, model-driven engineering, or aspect-oriented design.

Andreas Schaad, PhD, is a Senior Researcher at SAP Research. His research interests are organizational structures and behavior, security and business process management. He is a certified information systems security professional (CISSP).