The 30th Annual Conference of the IEEE Industrial Electronics Society, November 2 - 6, 2004, Busan, Korea

Maintaining the Integrity of XML Signatures by using the Manifest element

Omar Khadeer Hussain and Ben Soh

Department Of Computer Science and Computer Engineering, La Trobe University
Bundoora, VIC, Australia 3083
Farookh.Hussain@cbs.curtin.edu.au

Abstract—One of the aims of providing 'security of data' in e-commerce transactions is making sure that the receiver receives the same data which the sender sends, that is the data has not been tampered in any way. To achieve this aim digital signatures are used. A digital signature helps in providing integrity, message authentication, and signer authentication for the signed data. An XML signature can contain or point to the data that is being signed. In this paper we discuss a possible solution of avoiding a signature from breaking when there is a change in the location of the document after it has been signed.

I. Introduction

As XML is becoming the de facto standard for communication of data over the Internet, the need for security of the data should not be left behind. In fact with out security the whole purpose of achieving data exchange will have no value as there will be no guarantee about the origin and the integrity of the data. Digital security has always been about the compromise between convenience and peace of mind [1]. This holds true for XML also. Different approaches have been proposed that address the problem of protecting information in a Web system. However, these approaches typically operate at the file-system level, independently of the data that have to be protected from unauthorized accesses. A lot of study is being done at proving security at the element level [2, 3, 4].

The two main issues that need to be addressed are:

- Restricting the access to an XML service only to authorized users.
- Maintaining the integrity and confidentiality of the information exchanged.

For this signatures are used. It provides integrity, message authentication, and signer authentication for data that is being included by the signature. A signature helps in keeping the data secure and helps in checking its integrity. The sender computes a hash value of the data which is to be signed and signs it with his/her private key and includes it in the signature. The receiver authenticates the signature and decrypts the hash value with the sender's public key. It then creates its own hash value of the data and checks it with the hash value that came along with the signature. If they both match then the integrity of the document is assured [5]. If any changes are made to the document after the sender signs the document then the hash value computed at either ends doesn't match and the signature is said to be broken i.e. the integrity of the message is not guaranteed

II. Problem Definition

An XML signature can contain or point to the data that is being signed and thus it relies on any source available via a URI in the digital signature. But consider a change in the URI location of the data. How will this affect the signature?

We will consider a scenario where the signature that is created points to the data through the Reference URI which has an id "change". Suppose we know that the location where the data is contained is going to change from http://www.change.com/old.xml to http://www.change.com/new.xml after the signature has been created, but the data signed at the original location is going to remain the same. Then when the signature is being verified, the computed hash value at the receiver's end will not match with the hash value which was sent by the sender due of the change in the URI location and thus the signature might break. So our problem is how can we create an XML signature that enables the validity of the signature not to break even in the change of the source location when the document signed is going to remain constant?

III. Proposed Solution

The signature is a collection of a number of elements according to a schema and each element has its own cardinality. Cardinality specifies the number of occurrence of each element in the signature. An approach to solve the above problem is to utilize such an optional element of the XML Signature, the Manifest element. The use of the Manifest element enables the application to reserve the reference validation semantics for itself [6].

We will discuss about the Manifest element, the process of Signature generation (core generation) and Signature authentication or validation (core validation), and the steps needed to achieve the solution for the above problem in the next sections. Core generation and core validation process are discussed as it is necessary to know how the signature is created, after the changes are made by using the Manifest element.

IV. The *Manifest* Element

The `Manifest` element is used to provide additional requirements, which are not directly addressed by the basic structure of the XML signature specification. The Manifest

0-7803-8730-9/04/$20.00 ©2004 IEEE

element, like the SignedInfo element is a collection of Reference elements. But the difference lies between the levels of importance of the Reference element each of them contain. The SignedInfo is an important element in the Signature specification as it contains the Reference element that's points to the data being signed, the digest value computed and other elements. The signature algorithm is applied to this element and it is verified during the signature verification process [6].

On the contrary the Manifest element too is a collection of Reference elements but it itself is a referenced element. If it is used, then it is referenced by the Reference subelement of the SignedInfo element. Figure 1 shows a Manifest element "Man" being referenced by the SignedInfo element. For simplicity only the structure of the Signature has been shown and the details are omitted.

When the signature verification takes place, only the integrity of the Reference elements inside SignedInfo element is verified and not of the Reference elements that are inside the Manifest element. But the digest over the Manifest itself will be checked by the core verification process. Thus the validation of the data inside the Manifest element is not done by the Signature verification process but is dependent on the application. We will be utilizing this feature of the Manifest element to propose a solution to the problem mentioned.

```
<Signature ID="Sig1">
        <SignedInfo>
        <Reference URI="Man" >
        <DigestMethod Algorithm=" "/>
        <DigestValue/>
        </Reference>
        </SignedInfo>
        <Object>
        <Manifest ID="Man"/>
        <Reference URI=" "/>
        <DigestMethord Algorithm=" "/>
        <DigestValue/>
        </Reference>
        </Object>
        <SignatureValue>
        <KeyInfo>
</Signature>
```

Fig. 1: A Manifest element "Man"

V. Core generation

Core generation is the process of creating a signature. It consists of two steps, Reference Generation and Signature Generation [7]. Reference Generation aims at creating the Reference element that have all their sub elements like transforms, attributes and digest values. There can be more than one reference element in a signature. The aim of signature generation is to create the SignatureValue element and complete the whole Signature element with all the attributes and elements.

The steps of Reference Generation are for each Reference element is:

- Apply the Transforms (optional)
- Create the Digest value of the resource
- Create and complete the Reference element.

The steps of Signature Generation are:

- Create the SignedInfo element by using the Reference elements created in Reference Generation.
- Apply the canonicalization algorithm and the signature algorithm to the SignedInfo element created.
- Create and complete the Signature element.

In Signature Generation the SignedInfo element is canonicalized before it is signed. Canonicalization ensures that the same octets are signed and compared with which will ensure the signature validity, and it makes sure that semantically meaningless alterations which may happen as the signature is passed through the XML processors of various application doesn't break the signature.

VI. Core validation

Core validation is the process of checking the validity of the signature. It is an inverse process of core generation. It too is composed of 2 steps, Reference Validation and Signature Validation [7]. Reference validation is used to check wether the data referenced by the Reference element has not been altered. Signature validation checks the digest value of SignedInfo element with the value stored in SignatureValue.

The steps of Reference Validation for each Reference element in SignedInfo are:

- Canonicalize the SignedInfo element and obtain the data being pointed by the URI attribute of each Reference element
- It is then digested according to the algorithm specified in its corresponding DigestMethord Algorithm element.
- The digest value computed is now checked with the value stored in the DigestValue element. If they match then reference validation is said to pass.

The steps of Signature Validation for the SignedInfo element are:

- Obtain the verification key from the KeyInfo element.

With the signature algorithm compute the signature value over the canonicalized SignedInfo element and compare it with the value inside the SignatureValue element. If they match then Signature Validation is said to pass.

VII. Utilizing the *Manifest* element

The use of the Manifest element to contain the Reference element is a process of defying the steps of core validation [6]. Each Reference element which anticipates a change in location must be placed in the Manifest element, which is referenced by the Reference element of the SignedInfo element as shown in figure 2. For simplicity the other elements of SignedInfo elements are omitted.

In order to achieve a solution by using the Manifest element, we have include the data that is going to change its location in the Manifest element and make some changes prior to executing the steps of core generation and core validation. These changes are done by XSLT transformation.

```
<Signature ID="Sig1">
  <SignedInfo>
        ................
        <Reference URI="Man">
        </Reference>
        ................
  </SignedInfo>
        <Object>
        <Manifest ID="Man">
        <Reference id="change"
       URI="http://www.change.com/old/xml/>
        <DigestMethord Algorithm=" ...."/>
        <DigestValue>........ </DigestValue>
        </Reference>
        </Manifest>
        </Object>
        <SignatureValue>...</SignatureValue>
        <KeyInfo>.......... </KeyInfo>
</Signature>
```

Fig. 2: SignedInfo element

Our aim is to create a transformation which finds the desired URI attribute with the particular id in the Reference sub-element of the Manifest element, and leave it out of the final data **before** the digest of the message is created i.e. before the steps of core generation are executed. So that the change in the URI location will have no effect during core verification, when the digest over the Manifest element is checked.

```
 Manifest ID="Man">
<Reference Id="change"
 URI="http://www.change.com/old.xml ">
 <DigestMethordAlgorithm="....."/>
 <DigestValue/>
 </Reference>
 </Manifest>
```

Fig. 3: "Manifest" Element before

The XSLT transform have to be created in such a way so that the <Manifest> element changes from as shown in figure 3 to the one shown in figure 4.

Once the transformation is done, the URI is omitted from being a part of the digest value. The process of core generation is done and the signature is created. Core validation passes as the URI is not included from being a part of the digest value, and as a result of that the SignatureValue remains the same at both ends and is impervious to changes in the location of the URI. Thus by applying the transformation and omitting the URI location of the document which is going to change, from being a part of the digest value, we are allowing the document to change its location with out affecting the signature.

```
<Manifest Id="Man">
<Reference Id="change ">
<DigestMethord Algorithm=" ....."/>
<DigestValue/>
</Reference>
</Manifest>
```

Fig. 4: "Manifest" Element after

In the core generation & validation process only the Reference elements that are inside the SignedInfo element are de-referenced, digested and validated, so the internal processing and validation of the Manifest element has to be application specific. The Manifest element enables the application to reserve the reference validation semantics for itself. Thus the signature that is created no longer guarantees the location that is contained by the Reference element of the Manifest element and it is the responsibility of the application to verify the validity of that data when the core validation process is being done.

VIII. Conclusions

In this paper we discuss a possible solution of avoiding a signature from breaking when there is a change in the location of the document. Each URI which anticipates a change in its location must be placed in the <Manifest> element which enables the application to reserve the reference validation semantics for itself. We utilize this element to contain the data whose location is going to change and apply an XSLT transform to omit the URI attribute.

IX. References

[1]. Mark O' Neil, "XML Feature-XML and Security"

[2]. Ernesto Damiani, Sabrina De Capitani Di Vimercati, Stefano Paraboschi, Pierangela Samarati "A Fine-Grained Access Control System for XML Documents" TISSEC May 2002

[3]. Ernesto Damiani, Sabrina De Capitani di Vimercati, Pierangela Samarati, "Towards Securing XML Web Services" Proceedings of the 2002 ACM workshop on XML security2002

[4]. E. Damiani, S. De Capitani di Vimercati, S. Paraboschi and P. Samarati, "Securing SOAP e-services" International Journal of Information Security, February 2002

[5]. Ed Simon, Paul Madsen, Carlisle Adams, "An Introduction to XML Digital Signatures" http://www.xml.com/pub/a/2001/08/08/xmldsig.html

[6]. XML-Signature Syntax and Processing http://www.w3.org/TR/xmldsig-core/

[7]. Processing Rules: XML-Signature Syntax and Processing http://www.w3.org/TR/xmldsig-core/

Privacy and Security Shield for Health Information Systems (e-Health)

Mihaela Ulieru[1], Dan Ionescu[2]

[1] Director, Emergent Information Systems Laboratory, The University of Calgary, T2N 1N4 Alberta, Canada
Ulieru@ucalgary.ca
[2] Professor of Computer Science, School of Information Technologies (SITE), The University of Ottawa, Canada

Abstract—The objective of this work is to develop a platform supporting the secure and quick deployment of distributed medical applications creating an environment and associated tools for the usage of medical personnel in their interaction with patients. We adopted the Electronic Health Record (EHR) Architecture Blueprint as developed by Canada Health Infoway [1], which proposes Web service technology as an integration platform. We developed this environment for distributed and collaborative use by selected medical personnel using the combination of communication networks such as the Ca*net 4, ORION, NETERAnet and NCIT*net. This intelligent platform will enable the mining, retrieval, modification, management, and synchronization of various databases used by doctors in handling data in regards to patients and their illnesses, and last but not least, will examine and provide the security requirements associated with web services in the context of e-Health applications.

Keywords— e-Health; scalable, secure web-services; data mining, monitoring and management; privacy and security of the electronic health information

I. Introduction

In Health Care, there are many enterprise-oriented applications that have been used within a closed "circle-of-care". It has been widely recognized that an integration of these applications into a network-centric framework would likely result in significant service improvements and cost reductions [2].

A major challenge in this context is to develop scalable, secure web based services where the security and privacy framework is meant for the access to and the protection of sensitive information as it travels across the boundaries of individual organisations, in compliance with the Privacy of Information Act [3]. In this regard, e-health is one of the major blocks of the Secure Channel of the Canadian e-Government framework [4].

The backbone of our work is the secure web-data manipulation by medical specialists, while dealing with patients affected by diseases calling for a highly specialized knowledge and expertise. In this context a platform able to interact with a plethora of databases and other forms of information storage and retrieval methods is a must. The interaction of doctors with the information has to be secured through encryption and through a complex process of authentication and authorization. Some of the required security technologies have already been developed for other industries, e.g., in the area of electronic commerce.

Other technologies such as patient-consent dependent role-based access control and person-oriented audit trails are not ready available to date.

II. System Architecture

Figure 1 illustrates the system components, which will be described in the sequel.

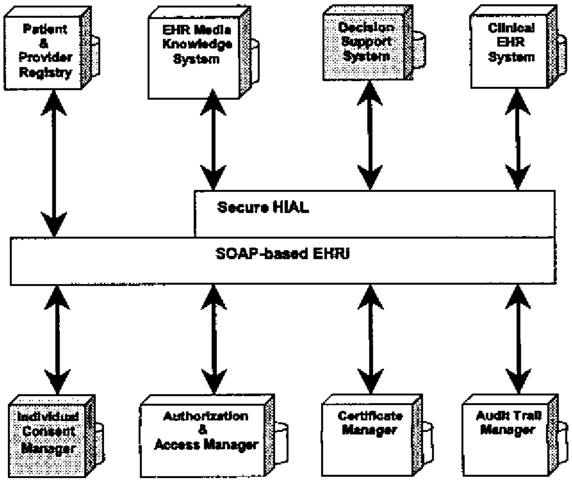
Patient & Provider Registry
EHR Media Knowledge System
Decision Support System
Clinical EHR System
Secure HIAL
SOAP-based EHRI
Individual Consent Manager
Authorization & Access Manager
Certificate Manager
Audit Trail Manager

Fig.1 Conceptual architecture

A. Authorization & Access Manager (AAM) using a Data Mining, Monitoring and Management Platform (DMMP)

The AAM uses the Certificate manager (CM) and the Individual Consent Manager (ICM) to control access to information content based on personal consent, the role of the information-accessing entity, and the type of information use. The DMMP component is the central control system of the data and metadata exploration, migration, unification, and management. It is meant to unify different data from different types from different databases used by different medical units, classifying them, storing the unified data in a database and allowing clients to retrieve the information with different degrees of flexibility. The platform is intrinsically providing the security infrastructure related to the management of authority certificates, the authentication and the encryption of data using standardized encryption algorithms.

0-7803-8730-9/04/$20.00 ©2004 IEEE

B. Decision Support System (DSS)

The DSS consists of two parts:

1) A Knowledge Sub-system
2) A Diagnostic Sub-system

The diagnostic Sub-system offers a primary use of personal health information for offering consultation while the Knowledge sub-system offers a secondary use of health information for training. The DSS is an excellent case study for analyzing and developing distributed security and privacy mechanisms that match a patient's given consent to the type of information use. Results will be applicable to a broad range of applications in health care.

For more details on the DSS, which is built as a dynamic service environment, using holonic multi-agent technology, please refer to [5].

C. Patient & Provider Registry (PPR)

The PPR is a directory service that contains information about patients and health care providers, such as their identification number, certification, domain expertise etc. The PPR is currently being developed in other projects funded by regional health authorities and the Canada Health Infoway.

D. Individual Consent Manager (ICM)

The ICM maintains and serves information about the consent given by individuals about the use of their personal health information. In addition to a Web service interface to be used by other framework components, it has a secure Web interface to enable individuals to review and adjust consent information.

E. Certificate Manager (CM)

The CM distributes and controls (potentially revokes) digitally signed trust certificates (X.509) of providers and patients. Trust certificates are essential for establishing authenticity as well as providing a basis for encryption based on a public key infrastructure (PKI). For DMMP see III.A.

F. Secure Health Information Access Layer (SHIAL)

Health Information Access Layer (HIAL) is a term defined in Canada Health Infoway's Electronic Health Record (EHRs) Blueprint Architecture.

The main purpose of this component is to leverage the value of existing heterogeneous medical applications and integrate them into a networked EHRs.

The SHIAL has to resolve heterogeneity on two levels, namely on the technological level and on the semantic level. From a technological point of view the HIAL provides a standardized way of accessing heterogeneous systems using Web service technology (SOAP, XML, and UDDI etc.).

From a semantic point of view, HIAL provides a conceptual mapping of data structures and terminologies used in the various heterogeneous medical systems to a standard ontology, based on the HL7 Reference Information Model (RIM).

In terms of security, SHIAL mediates between trust credentials on the inter-organizational network level and those trust credentials used within the enterprise. From an inter-organizational view, SHIAL security is based on functionality provided by AAM. Since the semantic mapping to RIM provides SHIAL with knowledge about the semantics of information accessed by network services, SHIAL can provide AAM with meta-data important for deciding whether to grant access for a particular use case.

G. Audit Trail Manager (ATM)

The ATM manages person-oriented audit trails for personal health information exchanged in the EHRs network.

Most currently available audit mechanisms are resource-based, rather than person-based. They log access operations to specific information resources such as data files, database tables, network objects etc.

In a network-centric architecture integrating many heterogeneous data sources, these mechanisms are too limited to provide answers to person-oriented auditing questions, such as "who has accessed what information about *me* and for what purpose?" Person-oriented audit-trails are difficult to achieve in heterogeneous environments because information content is structured in different ways, and, thus, it becomes problematic to distinguish anonymous information content from personal information. In our framework, this semantic heterogeneity problem is solved by the SHIAL, which maps heterogeneous concepts used at different organizations to a common RIM.

Consequently, the SHIAL will use the ATM to log all access to personal information on the network. In addition to logging these access patterns, the ATM has a role of resolving synonymous ways of identifying individuals (such as by name, by Personal Health Number - PHN etc.)

H. EHR Media Knowledge Base (MKB)

The availability of high bandwidth networks (such, as CA*Net4 [6] enable the use of multimedia and real-time collaborative web-services for pattern recognition in diagnostic images.

We extend the concept of a knowledge base with textual data to include diagnostic images and the expert system (part of the DSS) will be extended to offer consultation on graphical images, generated by the medical machines.

The MKB is based on off-the-shelf components. Diagnostic Imaging repositories are currently being developed in projects funded by Infoway and health authorities.

I. Clinical EHR System (C-EHR)

The C-EHR component in our EHRs stands for potentially many different clinical information systems in Vision Care to be integrated in the network-centric framework.

For the purpose of this project, we use VRIS (Vision Rehabilitation Information System), an in-kind contribution made by Jackson Willms Medical Services Inc.

III. Data Mining, Monitoring and Management Platform (DMMP) for E-Health Privacy Enhancement

A. Overview

The DMMP (Fig. 2), [7]is seen as the control layer of the entire system. The key problems to be addressed by the DMMP are related to the general architecture of the data indexing for unifying diverse data from different doctor's files about the patient history and diagnostic made as well as the data exploration, synchronization and reporting. These issues are to be investigated in the context of the SHIAL architecture. The mapping between SHIAL and DMMP has to be devised and implemented as the SHIAL is providing functions for the domain data understanding while the DMMP is providing the connection between data stored in data bases, file systems, emails, web-servers, and in general any form of data storing and manipulation.

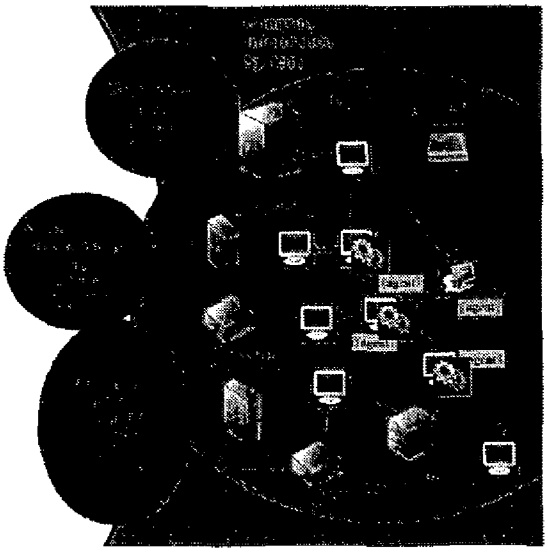

Fig.2 Overview of the data mining, monitoring, and management platform

The DMMP is capable of exploring and unifying information from any source such as:

- databases: Oracle, DB2, MSSQL(Microsoft SQL Server) , MySQL, Informix, Lotus and others Relational databases;
- Web sites: http and https; iii) File –systems;
- FTP(File Transfer Protocol)
- e-mails: IMAP(Internet Message Access Protocol) and Microsoft Exchange;
- JNDI (JAVA NAMING AND DIRECTORY INTERFACE);
- SAP

and it enables the automated categorization of information.

The destination can be:

- . databases: Oracle, DB2, MSSQL(Microsoft SQL Server), MySQL, Informix, Lotus and others Relational databases;
- File –systems;
- FTP(File Transfer Protocol);
- DMMF-finder, which offers highly sophisticated information retrieval that, is not restricted solely to search processes.

Through DMMP, the information can be accessed from practically any source. The accessed material is qualified and integrated into other applications. A manager of the platform (DMMP-manager) allows the administrator of the system to manipulate data among different databases by simple iconic interactions. The DMMP -manager automatically weeds out any irrelevant information, incorporating only relevant material in the data flow. The DMMP provides for fast, well-founded decision making and effective information management. DMMP contains also tools for the manipulation of data and metadata and thus for managing its infrastructure as well. The DMMP manager can perform:

- simple definition and adjustment of qualifying processes using an iconic programming language;
- the central control of DMMP-manager components;
- provision for automated categorization of information;
- the optimization of system's performance

The DMMP-manager is based on the DMMP -finder. The DMMP-finder allows DMMP-manager to understand information, thus adding a quality dimension never seen before in information retrieval systems. DMMP -manager ensures central access to a host of different data sources. With the help of an intelligent agent system (Fig. 2) [8], any changes can be identified and conveyed to the central system. This means, users are kept up to speed with the latest information at all times. The DMMP is compatible with portals and other systems. This degree of interconnectivity enables it to deliver qualified information from one or more data sources to any users or systems within the loop. DMMP is transparent to the hardware and the operating system used.

The DMMP has a layered architecture (Fig. 3). Functionality in the lower layers is used by the upper layers, while the user interacts with the highest level layers.

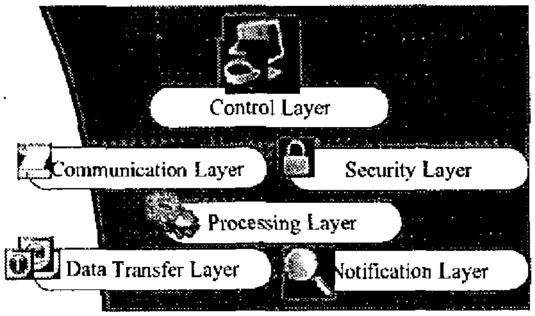

Fig. 3: DMMP architecture

- ❖ Data Transfer Layer. Provides uniform interfaces for reading and writing to data repositories. All information repositories are uniformly accessed through the data transfer layer.

- Communication Layer. Facilitates communication between DMMP modules.
- Security Layer. Handles credentials for accessing sensitive repository data.Based on roles and permissions and authorization tickets it authenticates users and authorizes all operations.
- Notification Layer. Monitors the source data repositories for new information. The notification layer monitors the data repositories and when content is created or updated, other data management components are notified.
- Processing Layer. Coordinates the data collecting, processing and distribution operations. Data is processed through processing scripts, this layer handles data translation, mapping and structuring.
- Control Layer. Responsible with the DMMP modules administration. Provides a complete management system for data management framework components, components can be remotely installed, configured and managed.

B. The Security Layer

DMMP access is granted only to authorized users, which can enter the platform only through the DMMP Security Layer (Fig. 4). Based on the user credentials, the security layer determines the user roles and permissions, and the data that can be available for the user. The key security concept that lay the foundation of the DMMP functionality are:

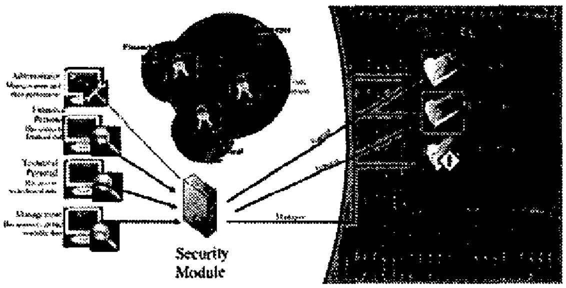

Fig. 4: DMMP secure access workflow

Permissions. A permission specifies the right of performing an operation or accessing a resource.

Roles. Roles represent collections of permissions.

Authorization Tickets. The authorization ticket represents the information used to securely identify the session with the client. Based on the authorization ticket that has been obtained as a result of the login operation, the client identifies its session using this authorization ticket. The authorization ticket carries information related to the user session and it is signed with the security module's secret key in order to avoid malicious usage. The authorization ticket has a very short validity period (some minutes) calculated from the last usage time of the ticket.

Security Session. Once the client authenticates himself to the system a security session is created for him on the security module server side. Each security session has associated an authorization ticket, which is passed to the client. The authorization ticket is the identifier of the session. The security session maintains the security connection context for the user that has been authenticated in the system.

User Security Context. Every time a user logs in, a user security context is created (if it was not created before - a user may login in the system more times through different modules). There is a connection between the user security context and the authorization ticket. The user security context maintains secure information related to the user (user credentials, user principal, etc.).

Authentication Ticket. The authentication ticket represents a substitute for the credentials that the users must provide to authenticate themselves to the system.

Heartbeat. When the authorization ticket is not used in some security operation for a longer period of time, the client may loose the session. In order to avoid this situation, the client may send heartbeat signals to the security layer, maintaining the session alive.

Supported encryption algorithms:

- symmetric key based encryption (DES, DESede, Blowfish)
- asymmetric key based encryption (DSA, RSA)

IV. SECURE-HIAL (HEALTH INFORMATION ACCESS LAYER) AND AUDIT TRAIL MANAGER

The two previously mentioned key problems to be addressed in the SHIAL (Fig. 5), namely resolution of technological and respectively semantic heterogeneity, are addressed on two separate layers in the conceptual SHIAL architecture.

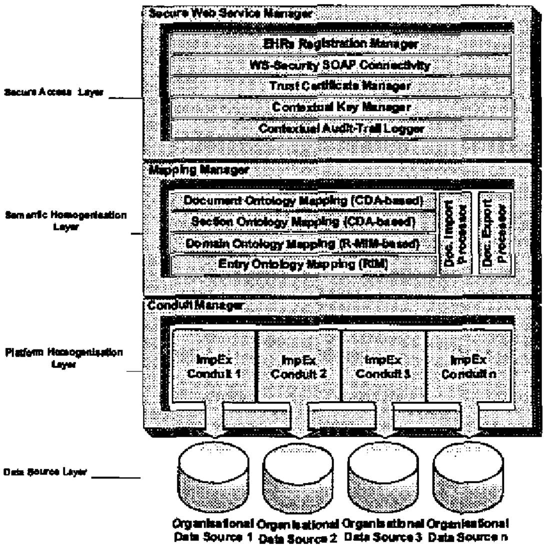
Secure Web Service Manager
EHRs Registration Manager
WS-Security SOAP Connectivity
Trust Certificate Manager
Contextual Key Manager
Contextual Audit-Trail Logger
Secure Access Layer
Mapping Manager
Document Ontology Mapping (CDA-based)
Section Ontology Mapping (CDA-based)
Domain Ontology Mapping (R-MIM-based)
Entry Ontology Mapping (RIM)
Doc. Import Processor
Doc. Export Processor
Semantic Homogenisation Layer
Conduit Manager
ImpEx Conduit 1
ImpEx Conduit 2
ImpEx Conduit 3
ImpEx Conduit n
Platform Homogenisation Layer
Data Source Layer
Organisational Data Source 1
Organisational Data Source 2
Organisational Data Source 3
Organisational Data Source n

Figure 5: Secure-HIAL Architecture

The first problem is addressed in the Platform Homogenization Layer, which hosts an extensible plug-in architecture of so-called ImpEx Conduits (import/export adapters). Each such conduit interfaces to a different type of physical data repository and provides a canonical, XML-based data access layer to be used by the next SHIAL architectural level, the Semantic Homogenization Layer

(SHL). The SHL provides a semantic mapping of the intra-organizational meta-data to standardized terminologies provided by HL7 domain ontologies [9]. This complex function is broken down in four sub-functions:

- ❖ Information units exchanged between SHIAL and the EHRs network are called documents in our terminology (which adopts the HL7 Clinical Document (CD) Architecture. The Document-Level Ontology Mapping provides a coarse-grained association between document types in the CDA and information units to be exchanged with the SHIAL to be deployed in a particular organization. Moreover, it defines how to generate and process the contextual meta-data that has to be associated with all CDA documents (such as confidentiality, author, subject etc.).
- ❖ The Section-Level Ontology Mapping provides a more fine-grained association between different semantic sections in each document type with section-types defined in the HL7 CD architecture.
- ❖ The Entry-Level Ontology Mapping derives the semantics of the intra-organizational meta-data from information types defined in the HL7 Reference Information Model (RIM).
- ❖ The Domain-Level Ontology Mapping provides a further detailed association between entries in document sections and organizational meta-data, based on a domain-specific R-MIM. (HL7 standard [9] defines an R-MIM as an "Information structure that represents the requirements for a set of messages. A constrained subset of the RIM which may contain additional classes that are cloned from RIM classes".

On the top of the SHIAL architecture stack is the Secure Access Layer (SAL), which implements advanced security services and Web-service connectivity. Of particular importance is the Contextual Audit-Trail Logger, a component that scans the contextual meta-data of all in- and outgoing documents and logs consistent audit trails using the ATM. The Contextual Key Manager securely hosts a key-chain of keys for accessing information in documents, depending on the parameters of the individual usage context (such as role of the organization, purpose of use, identity of person, etc.). The Trust Certificate Manager holds a set of electronic trust certificates issued by the ATM. The top two components in the SAL are responsible for providing standards compliant Web service interface (SOAP/WS-Security) for document exchange, and for registering the SHIAL information services with EHRs registries.

V. AUTHORIZATION AND ACCESS MANAGER (AAM) AND INDIVIDUAL CONSENT MANAGER (ICM)

The AAM's main function is to answer the question "Does prospective user X have the right to perform function Y on data set Z." X might be an individual or a system. Y might be, for example, display, print, copy, email, or import into a clinical system. Z is often characterized as being about a person and of a particular information type. The ICM's main purpose is to provide the systematic capability for individuals to author and record their preferences (policies, rules) for the dissemination and use of their private health information (medical patient records), and then to serve that information to the AAM. The two components work together to effect individual privacy preferences in practice at the point of use of patient health records. They must be able to handle cases where the records are in a structured database or in document form (a human-readable file such as a word processor file) independent of any identified database or document repository. CANARIE recently supported the Policy and Peer Permission (PPP) system project involving RightsMarket:

(http://www.rightsmarket.com), University of Calgary Telehealth (http://www.fp.ucalgary.ca/telehealth/), and the Ottawa Heart Institute. The AAM to a large extent, and ICM to a significant extent, are derived from the deliverables of that successful project and integrated with the DMMP component.

VI. APPLICATIONS

The project is focused on a specific EHR solution, in the context of a decision support system (DSS) for Glaucoma Progression Monitoring, Fig. 6 [10]. The application domain is the area of health science applications, namely the research of using web-services in assisting doctors for the investigation of the progression process along a patient's lifetime in the case of glaucoma monitoring and treatment.

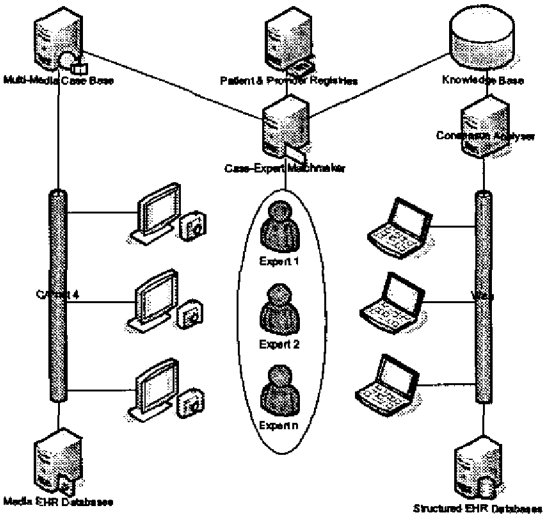

Figure 6: Solution Domain Architecture

The DSS will support the doctors in the diagnostic, treatment and supervision processes of the evolution of a glaucoma patient based on the exploration of all data pertinent to the case and on the scientific data contained in various professional databases [12]. The DSS consists of an Education&Consultation System [11] to provide evidence-based guidelines of care to clinicians, and a Consensus Analyzer [13] to constantly update and refine these guidelines based on patient encounters and expert opinions. Users can access the DSS directly via a Web-based user interface, or indirectly by using their clinical EHR system, which integrates with the DSS. The DSS is also integrated with other EHR infrastructure services, such as patient and provider registries. In addition to structured data, the Glaucoma monitoring system uses high-resolution diagnostic imaging supported by CA*net 4, ORION, NETERAnet etc. The medical specialist interacts in real-time with the various data collected, unified, and explored through the DMMP and provided to the DSS components. With respect to the HL7 e-Health communication standard, we are working on the creation of an R-MIM for Glaucoma diagnosis and guidelines, based on our previous work on the development of e-Health ontologies [8].

VII. Conclusions

The main contributions of our work are:

(1) a reference model for secure web-services as a refinement of Infoway's EHR Blueprint with respect to aspects of security,

(2) specification and design of an integrated environment and tools for supporting the activity of the medical specialist while curing patients with glaucoma

(3) implementation of these components and their integration with components developed by other EHR initiatives (e.g., patient and provider registry), and

(4) their evaluation with a specific net-centric pilot application: glaucoma progression monitoring.

VIII. Acknowledgements

The contribution of Professor Jens Jahnke from the University of Victoria, Canada to the integration of this work with Canada Health Infoway's SHIAL as well as the visionary contribution of Dr. Maike Miller from National Capital Institute of Telecommunications (NCIT) Ottawa to integrate our work within the CANARIE research, are gratefully acknowledged.

IX. References

[1] http://infranet.uwaterloo.ca/infranet/s20030618.htm

[2] http://www.hc-sc.gc.ca/ohih-bsi/chics/achi_fpt_ccis_e.html

[3] http://www.epic.org/

[4] http://www.comnet-it.org/egovernment/cdnexperience.pdf..

[5] Ulieru, M. "Internet-Enabled Soft Computing Holarchies for e-Health Applications", in *New Directions in Enhancing the Power of the Internet, (L.A. Zadeh and M. Nikravesh – Editors)*, pp. 131-166, Springer Verlag, Berlin, 2003.

[6] http://www.canarie.ca/

[7] http://www.artisinc.com/about_artis/artisactivity.htm

[8] Ulieru, M., Maja Hajdec and Elizabeth Chang] Ontology-Based Holonic Diagnosis System for the Research and Control of Unknown Diseases, *3^{rd} IASTED International Conference on Biomedical Engineering (BioMed 2005)*, Innsbruck, Austria, February 16-18, 2005.

[9] http://www.hl7.org/

[10] Ulieru, M., Soft Computing Agents for e-Health, *Proceedings of NAFIPS 2004*, June 28-30, 2004, Banff, Canada, pp. 116-121

[11] Ulieru, M., Andrew C S Crichton, M. Rizzi and Cynthia Karanicolas "Using Soft Computing to Define Standards of Care in Glaucoma Monitoring", *International Journal of Soft Computing: A Fusion of Foundations, Methodologies and Applications (Springer)* ISSN 1432-7643, 2003 (available on Springer's website, Journal currently in print).

[12] Ulieru, M. and Alexander Grabelkovsky, "Telehealth Approach to Glaucoma Progression Monitoring", *International Journal of Information Theories and Applications 10(3)*, 2003, ISSN 1310-0513, pp. 326-330.

[13] Ulieru, M. and Rizzi, M. A Cooperative Approach to the Development of Expert Knowledge Bases Applied to Define Standard of Care in Glaucoma, *Proceedings of CoopIS 2003*, Catania, Sicily, November 3-7, 2003, pp. 235-243, Springer Verlag Lecture Notes in Computer Science LNCS 2888