SecFHIR: A Security Specification Model for Fast Healthcare Interoperability Resources

Ahmad Mousa Altamimi
Department of computer Science
Applied Science Private University
Amman, Jordan

Abstract*—Patients taking medical treatment in distinct healthcare institutions have their information deeply fragmented between very different locations. All this information --- probably with different formats --- may be used or exchanged to deliver professional healthcare services. As the exchange of information/ interoperability is a key requirement for the success of healthcare process, various predefined e-health standards have been developed. Such standards are designed to facilitate information interoperability in common formats. Fast Healthcare Interoperability Resources (FHIR) is a newly open healthcare data standard that aims to providing electronic healthcare interoperability. FHIR was coined in 2014 to address limitations caused by the ad-hoc implementation and the distributed nature of modern medical care information systems. Patient's data or resources are structured and standard in FHIR through a highly readable format such as XML or JSON. However, despite the unique features of FHIR, it is not a security protocol, nor does it provide any security-related functionality. In this paper, we propose a security specification model (SecFHIR) to support the development of intuitive policy schemes that are mapping directly to the healthcare environment. The formal semantics for SecFHIR are based on the well-established typing and the independent platform properties of XML. Specifically, patients' data are modeled in FHIR using XML documents. In our model, we assume that these XML resources are defined by a set of schemes. Since XML Schema is a well-formed XML document, the permission specification can be easily integrated to the schema itself, then the specified permissions are applied to instance objects without any change. In other words, our security model (SecFHIR) defines permissions on XML schemes level, which implicitly specify the permissions on XML resources. Using these schemes, SecFHIR can combine them to support complex constraints over XML resources. This will result in reusable permissions, which efficiently simplify the security administration and achieve fine-grained access control. We also discuss the core elements of the proposed model, as well as the integration with the FHIR framework.

Keywords—Healthcare; FHIR; Interoperability; Privacy preserving; Standards; XML schema

I. INTRODUCTION

Vast amount of information on health is created in the process of treatment in different medical centers such as hospitals, clinics, or other institutions. As a result, the information about patients becomes scattered over a large number of distinct health information systems [1]. To provide appropriate treatment, this information may need to be accessed and exchanged towards the successful execution of medical process. However, the variety of involved providers, as well as the diversity of health data, end up not being used due to the difficulties faced in exchanging and integrating issues. In fact, despite the recent research in the domain of information systems, it is still not completely settled out the exchanging and integrating challenges. This leads to inefficient treatment as the individuals' information may need to be accessed and processed at the right time and place [2].

The dis-connect in systems is due to two important reasons. First, the technological challenges for integrating heterogeneous legacy systems/information, which are almost impossible to directly include into any reasonable interoperable work-flow. Second, the lack of enforceable security policies for data interoperability and exchange as many healthcare applications operate in isolated mode that do not share information in an efficient manner. The solution for the former problem can be found in the frameworks that regulate how information is structuring and exchanging, while providing a security specification model can solve the latter problem.

We note that existing approaches have been proposed to achieve interchanging information (interoperability). Interoperability is widely acknowledged as a key requirement for the success of healthcare information systems. For example, the interoperability has a considerable benefits to national economies throughout the world, it has been estimated at USD\$77.8B [3] in the USA alone. Therefore, organizations proposed several approaches to provide interoperability. Health Level Seven (HL7) organization, as an example, developed a comprehensive framework and related standards for exchanging, integrating, and sharing healthcare data since 1987 to improve patient care [2, 4]. Recently, Fast Healthcare Interoperability Resources (FHIR) is introduced as a next generation standard to address fundamental limitations in HL7, and provide an implementable framework, supported by the prior experience, to apply best practices and avoid some of the pitfalls of previous work [5, 6].

FHIR attains interoperability in part by the use of consistent standards that define the syntactic and semantic meaning of information. Of course, using consistent standards reduce time, effort, and costs of health technology development projects. In practice, FHIR represents granular clinical concepts as a set of resources that may be addressed separately or in aggregating mode such as Patient, and Document. Its build upon the HL7-defined set of ``resources" aiming to support more modern approaches and be more developer-friendly for information

sharing, which includes documents, messages, services and RESTful interfaces. Some of these infrastructure resources define the standard itself --- i.e. what coding are used with them, what are resources' characteristics, etc. This set of resources is referred to as the "Conformance resources" [7,8,9].

That being said, while FHIR provides the most interesting standard since the original HL7, for exchanging health data in highly readable format such as XML, it does not define any kind of security related functionality [5]. In other words, the biggest concern is not just exchanging the patients' information but also preserve patients' privacy. The provided interoperability feature requires a proper level of awareness to provide the right security countermeasures. The security and privacy issues that arise within this context can then be handled with the appropriate policy specification that identifies and authenticates users in order to preserve the patients' privacy. For example, in a healthcare heterogeneous environment, an XML document generated according FHIR standards can consist of information about patients. In case of accessing this document by internal or external user, her access permissions should be validated according to security policies. The final decision which determines whither user can read or not is the result of the overall authorization constraints.

To control all accesses to XML instances in such environment without doing major changes to FHIR structuring, we propose a policy specification model (SecFHIR) that is based on XML Schema. SecFHIR relies on the concepts of XML Schema that supports complex constraints over XML elements in an XML instances. These instances can then be combined together to create more expressive policies. A primary objective of this approach is to allow policy designers to identify security/privacy constructs taking into account the requirements of healthcare environments such as the highly dynamic nature and the exceptional circumstances where privacy policy is override in emergencies cases. To support the evaluation of the proposed approach, we provide a prototype implementation developed specifically for this environment. The framework consists of three main components. The User Interface Tool allows end users to design policies either by using an XPath or the XQuery to be a XML query language with SQL-like syntax that directly interacts with the information repository. The Policy Repository, in turn, stores the policies generated by the interface tool. Finally, the Policy Manager retrieves policies from the repository and delivers them to the access control module for enforcement.

The remainder of the paper is organized as follows. An overview of related work is presented in Section II. Section III discusses the terminology relevant to the Fast Health Interoperable Resources and FHIR Information Modeling. The security specification model is then presented in detail in Section IV. The integration to FHIR framework discussed in Section V. Final conclusions and the future work are offered in Section VI.

II. Related Work

The need for developing standards for the exchanging of electronic health information has long been identified in the literature. In 1987, the Health Level Seven International (HL7) has founded as a non-profit organization dedicated to developing a comprehensive framework for the exchange and retrieval of electronic health information [2]. Recently, in 2014, the Fast Healthcare Interoperability Resources (FHIR) is introduced as a next generation standard to address fundamental limitations in HL7, where the patient's information or resources are structured and standard through a highly readable format such as XML or JSON [5]. However, a primary limitation of this approach is that it lacks for security countermeasures.

Because FHIR defines patient's resources as XML documents, some related works concerning XML security can be considered. Ponder2 as an example, which is an XML-based language that specifies security and management policies in a subject-action-target (SAT) format [10]. SecPAL credentials 30], on the other hand, are expressed using predicates defined by logical clauses, in the style of constraint logic programming. We note, however, that while these approaches are efficient when used on large-scale networks and distributed systems, they are not well-suited to healthcare information as they are often fragmented.

Creating secured views have also been considered, authors in [11] generated views for the required XML document using XSLT transformation technology. Typically, alternate views are defined for each distinct user or user group depending on their privileges. The end result is often the generation of a large number of such views, all of which must be maintained manually by the administrator. Clearly, this approach does not scale terribly well, and would be impractical in complex healthcare environment.

The encryption and key management technology have also used in this context. Authors of [12] provided a solution using encryption technique to secure delivery of XML documents, while authors of [27] encrypted their data before uploading them to cloud servers. This enables patients to create, modify, manage and control data in a centralized place from anywhere and at any time, as well as to share their data with wide range of users. However, for a well-designed encryption technique time-consuming brute force method and large computational resources are required to decrypt the data.

The RBAC model has been utilized to provide XML security. For example [13] defined their own language to describe roles and permissions, which are kept in RBAC model for XML document stores. In fact, a number of researchers have looked at similar techniques, for example in [14], the RBAC model is extended to provide an access control in Dynamic XML-based Web-Services. Authors presented an XML-based RBAC (XRBAC) policy specification framework for enforcing access control in dynamic XML-based Web services. The DTDs are also used to represent authorizations such as [15], which used XML format authorization based on DTD, and embedded this with X.509 attribute certificate.

Ultimately, from industry, eXtensible Access Control Markup Language (XACML) [16], initially defined by OASIS (Organization for the Advancement of Structured Information Standards), is proposed as a declarative access control policy language that expresses policies in an XML format. SAML on the other hand, defines the sources of the security statements in XML. The XML statements are then used in the processes of

authentication and authorization. Still other works investigate the integration of security assertion with client side code, with protection provided by analyzing user's requests [17]. Such XML-based languages are particularly suitable to convey requirements related to authorization and privacy for web-based systems [28]. However, it can be more difficult to adapt them to other environments (i.e., FHIR). Moreover, policies expressed directly in XML are verbose and hard to read and write [29].

III. PRELIMINARIES

As healthcare data is increasingly becoming digitized and widespread over different places. The data must be interoperable (structured and standardized) before sharing or processing by doctors. In fact, numerous health standards have been developed to support interoperability such as Fast Health Interoperable Resources (FHIR). FHIR is the recently proposed standard from HL7 organization that enables exchanging health data in highly readable XML, RDF, and JSON format [5]. It supports a user friendly implementation, built-in clinical terminologies, and is based on widely-used web standards like HTTP. In this section FHIR is briefly described, from a technical, standard and developer perspective.

A. Fast Health Interoperable Resources

The basic building block in FHIR is a Resource. All exchangeable content is defined as a resource or entity, which has a known identity by which it can address itself as one of the resource types. Example resources include: Patient, Device, and Document. At the present time there are 32 resources defined with many more under consideration. Resources all share the following set of characteristics:

- A common way to define and represent them. In the regards, a set of data types are used to building and defining common reusable patterns of elements. In fact, there are two main categories of data types: simple and complex types. The simple types are those that define single elements (e.g., string, integer, url, date, ect.). The complex types, on the other hand are re-usable clusters of elements. These types are represented as XML elements with parent-child relationship and all the presented elements are defined by their names and types. Any of the XML elements may have an id attribute. Fig. 1 illustrates a simple example for patient resource in XML format taken from [18], where the "patient_example" element includes several sub-elements. Each sub-element is a simple type or complex type (e.g., table).

```
<Patient xmlns="http://hl7.org/fhir">
  <id value="example"/>
  <text><status value="generated"/>
   <div xmlns="http://www.w3.org/1999/xhtml">
<!-- use FHIR code system to present patient information -->
      <table><tbody>
         <tr> <td>Name</td> <td>Peter James</td> </tr>
         <tr> <td>Address</td> <td>534 Erewhon, London, UK,
                                  139</td> </tr>
         <tr> <td>Contacts</td> <td>Home:(01)55556473</td></tr>
         <tr> <td>Id</td> <td>MRN:135(Acme Healthcare)</td></tr>
        </tbody></table>
    </div>
  </text>
<!--    use FHIR code system for male / female    -->
  <gender value="male"/> <birthDate value="1974-12-25">
    <extension url="http://hl7.org/fhir/StructureDefinition/
                    patient-birthTime">
      <valueDateTime value="1974-12-25T14:35:45-05:00"/>
    </extension> </birthDate>
  <deceasedBoolean value="false"/>

<!--    MRN assigned by ACME healthcare on 6-May 2014    -->
  <identifier>
    <use value="usual"/>
     <type><coding><system value="http://hl7.org/fhir/v2/0203"/>
      <code value="MR"/>  </coding>  </type>
      <system value="urn:oid:1.2.36.146.595.217.0.1"/>
      <value value="12345"/>
      <period>  <start value="2041-05-06"/>  </period>
      <assigner> <display value="Acme Healthcare"/>  </assigner>
  </identifier>
  <active value="true"/>
<!--    Relationship contacts    -->
<contact>
    <relationship>  <coding>
        <system value="http://hl7.org/fhir/patient-contact-
                       relationship"/>
        <code value="partner"/> </coding>
    </relationship>
      <name> <family value="du">
      <extension url="http://hl7.org/fhir/StructureDefinition/
                      iso21090-EN-qualifier"></extension>
      </family> <family value="Marché"/>
                <given value="Bénédicte"/> </name>
      <telecom> <system value="phone"/>
              <value value="+1 (237) 998327"/> </telecom>
      <gender value="female"/>
      <period>  <start value="2012"/>  </period>
  </contact>
  <managingOrganization> <reference value="Organization/1"/>
  </managingOrganization>
<!-- ..... -->
</Patient>
```

Fig. 1. A patient resource in XML format

- A common set of metadata that facilitate the technical and design context for the resources. We note that, the metadata items are optional, however some of them may be required in some implementations.
- A human readable part that contains a brief description of the resource. This part is used to represent the content of the resource to a user. It provides all the information needed for a user to understand the provided clinical and business information.

B. FHIR Information Modeling

FHIR aims to present the common patients' use cases. With FHIR, the patients' information are often modeled based on the composition approach. It structures the defined information contents by combining a set of resources together through their references. This ends with set of information that can be shared by the majority of the healthcare systems. Definitely, a single resource can be used by itself for particular use cases, however in most cases resources are combined and tailored to meet common use cases with specific requirements.

IV. The Security Specification Model

As mentioned in the previous section, patient's data/resources are modeled in FHIR using XML documents. In our model, we assume that these XML resources are defined by a set of schemes. Since XML Schema is a well-formed XML document, the permission specification can be easily integrated to the schema itself, then the specified permissions are applied to instance objects without any change. In other words, our security model (SecFHIR) defines permissions on schema objects, which implicitly specify the permissions on XML resources. Using these schemes SecFHIR can combine them to support complex constraints over XML resources.

That is, a FHIR resource may be generated with varying security requirements. The user, on the other side, may have a permission to access only particular parts of such instance. So users permissions can be defined using XML Schema for the corresponding FHIR resource. After that, the permissions are transported to all XML instances specified by this schema, then it will be transparently and consistently propagated to all relevant resources specified by this specific schema before sharing it with different users. Schemes elements can be further combined together to create more expressive policies. At the same time, because of the rich relationships between schema elements or between schemes themselves the generated complex permissions can be reused. Based on these unique features, SecFHIR would express efficient permissions that can in turn simplify the security administration.

A. Proposed Architecture

In fact, the SecFHIR architecture consists of four major components: Policies Repository, Security Schemes, XML Parser, and Security Specification Engine. Figure 2 provides an illustration of the architecture itself. In the following subsections, we discuss these components and provide a brief description of their Functionalities.

a) Policies Repository: In SecFHIR, administrators are responsible for defining security policies. The defined policies consists of a series of conditions (restrictions and/or permissions) identified by users credentials. These policies are stored in the policies repository and used by the Security Engine to generate Security Schemes for the requested XML resources. Policies are stored as a set of tables (users, permissions, and data elements) in the repository that collectively represent the security meta data. Ultimately, administrators interact with the policy repository using front-end tools that is typically a graphical, interactive interface. The important of the GUI is to hide the unnecessary details and facilitate the using of resources elements to be protected in a simple way.

b) Security Schemes: The second and more significant component is the Security Schemes. Security schemes are created in SecFHIR to provide templates, or blueprints to define the security permissions for FHIR resources. At the present time there are 32 resources defined in FHIR include: Patient, Device, Document, etc. So, an equivalent number of security schemes have been created, each of which defines the permissions and access types for each resource.

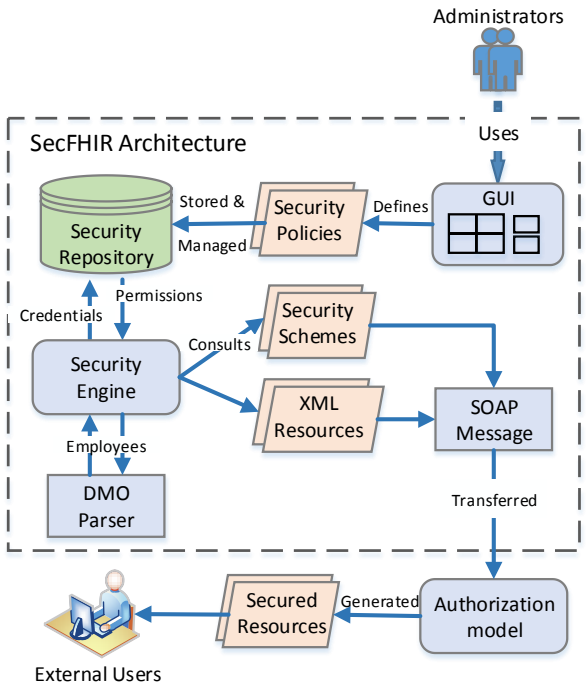

Fig. 2. The SecFHIR Architecture

For example, the core Schema in SecFHIR is the Patient schema where it is used to set the access constraints on patient's resource. Figure 3 illustrates the corresponding security schema for the XML instance shown in Figure 1. Due to the space limitation, we just provide the patient's schema, however the same principle can be applied to the remained resources.

The schema begins with the XML declaration followed by the root element. All elements may be properly nested and each element must be closed. In this schema, each component is followed by one or more < permission > tags. For example: Because XML Schema is XML like document, it can be validated and used after parsing it just like normal XML instance. In addition, the sender describes the data as the receiver will understand. Based on these features, languages like XPath and XQuery can be used to make data contained in the instance available to other applications.

c) Security Specification Engine: The Security Specification Engine is a bidirectional component. As mentioned previously, while security policies are defined and stored in the policies repository, the engine in turn, contacts the repository and retrieves the applicable policies, which are then integrated to the schema that defines the FHIR resource to be secured. Protecting FHIR resource is based on the three common access types: Read, Update and Delete.

1) Read: the most frequent operation by external users such as hospitals or healthcare centers users. A user in such institute is authorized to read the patient's info upon the permissions given to his/her roles.

```
<xs:schema attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="http://
www.w3.org/1999/xhtml" xmlns:xs="http://www.w3.org/2001/
XMLSchema">
 <xs:element name="div">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="table">
     <xs:annotation>
        <xs:documentation> use FHIR code system to
                           present patient information
        </xs:documentation>
     </xs:annotation>
    <xs:complexType>
     <xs:sequence>
      <xs:element name="tbody">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="tr" maxOccurs="unbounded"
                          minOccurs="0">
            <xs:permission>
                <xs:PermissionType = "read",
                                  value = "True"/>
                <xs:PermissionType = "update",
                                  value = "True"/>
                <xs:PermissionType = "delete",
                                  value = "False"/>
                </xs:permission>
        <xs:complexType>
          <xs:sequence>
            <xs:element type="xs:string"
                        name="td" maxOccurs="unbounded"
                        minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
            </xs:element>
       </xs:sequence>
      </xs:complexType>
     </xs:element>
    </xs:sequence>
    </xs:complexType>
   </xs:element>
  </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Fig. 3. The SecFHIR Architecture

2) Update: is used to modify the content of a resource. The updating permission value should be set by the resource's creator on the targeted XML schema before sharing its resource.

3) Delete: this type is used to remove a resource or some elements, including the elements and contents. Also the deleting permission value should be set by the resource's creator before sharing the XML resource.

For example, refereeing to the Fig. 3 deleting the patient's information is restricted because the delete permission value is set to False, while it is authorized to read or update the information.

a) XML Parser: To properly validate and utilize the FHIR resource, it must be first parsed and decomposed into its basic elements. In fact, the DOM parser utility [19, 20] is utilized by the Security Specification Engine to produce a DOM tree that represents the raw contents for both the schema and resource. Here, the parser first verifies that the resource has valid syntax corresponding to the provided schema, and then builds the parse tree. After that the security engine is set the values for the three access types: Read, Update and Delete according to the permissions stored in the security repository.

b) Authorization Module: While the proposed model provides a security mechanism for securing the patient's information defined by FHIR, it is important to note that without an appropriate user-side authorization module to permit or deny the user access, based on the user's privileges, the process of securing patient's information is not possible. Many authorization modules have been proposed in the literature such as [21, 22, 23, 24]. For example, the authors in [24] provided an authorization model based on RBAC and XML Schema. The model can be integrated and used by user-side machines to authenticate and authorize FHIR resources accessing. At this point, one can assume that such module exists, however in the future work, we aim to design a module which satisfies the required security measures for FHIR resources.

V. Integration to FHIR Framework

FHIR resources can be exchanged between internal and external users. While internal users can use any proper application to use the available local information, external users have to exchange patients resources over the network. This is done by using Hypertext Transfer Protocol (HTTP), which is the best way to communicate between applications because it is supported by all Internet browsers and servers, and SOAP (Simple Object Access Protocol) message [25], which is a messaging protocol used to encapsulate data as messages and exchange them via HTTP [26].

That is, a SOAP message is an ordinary XML document contains a differentiate element that identifies the resource, a header and body elements, which contain the requested information. SecFHIR exploits SOAP messaging for sending security schemes to different users. These messages were designed so that they can be tunneled over HTTP. This would and did help in its rapid adoption. Because the infrastructure of HTTP is already in-place, users would not have to spend extra money for another kind of implementation. Instead they can expose and access web services using technology already deployed.

In addition, the integration allows to easily transferring the healthcare information in secured mode, which provides a way to communicate between different applications with different technologies running on different operating systems. The success of this integration demonstrates that the general principles behind our model are broadly applicable to any standards FHIR resource, regardless to the content of the resource.

VI. Conclusion and Future Work

The big revolution in healthcare is supported by an increasing number of modern technologies such as medical devices that help in collecting or sensing continuously patients' information. Providing efficient healthcare is highly dependent on the availability of the collected information. However, to accomplish the aforementioned goal, the data should be interoperable (able to exchange) between healthcare providers. The biggest concern is not just exchanging the patients' information but also preserve patients' privacy. Specifically, many standards have been proposed to accomplish interoperability principle such as FHIR, which is comprehensive standards for exchanging and sharing healthcare information. That is being said, FHIR does not provide any type of security nor a security protocol for transferring information. Therefore, in this paper we present a security specification model that defines security policies on the level of XML schemes. These policies are implicitly mapping to the FHIR resources and can then be combined together to support complex constraints over resources. This will result in reusable policies, which efficiently simplify the security administration and achieve fine-grained access control. In the future work, we aim to design an Authorization module that can be plugged at client side, which satisfies the required security measures for FHIR resources.

Acknowledgment

The authors are grateful to the Applied Science Private University, Amman-Jordan, for the full financial support granted to cover the publication fee of this research article.

References

[1] Rostad, Lillian. Access control in healthcare information systems. Diss. Norwegian University of Science and Technology, 2008.

[2] Benson, Tim. Principles of health interoperability HL7 and SNOMED. Springer Science \& Business Media, 2012.

[3] T. M. Alenazi and A. A. Alhamed, "A Middleware to Support HL7 Standards for the Integration between Healthcare Applications," Healthcare Informatics (ICHI), 2015 International Conference on, Dallas, TX, 2015, pp. 509-512.

[4] N. Beredimas, V. Kilintzis, I. Chouvarda and N. Maglaveras, "A reusable ontology for primitive and complex HL7 FHIR data types," 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, 2015, pp. 2547-2550.

[5] Franz, Barbara, Andreas Schuler, and O. Kraus. "Applying FHIR in an Integrated Health Monitoring System." EJBI 11.2 (2015).

[6] U. Pervez, O. Hasan, K. Latif, S. Tahar, A. Gawanmeh and M. S. Hamdi, "Formal reliability analysis of a typical FHIR standard based e-Health system using PRISM," e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on, Natal, 2014, pp. 43-48.

[7] D. Bender and K. Sartipi, "HL7 FHIR: An Agile and RESTful approach to healthcare information exchange," Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems, Porto, 2013, pp. 326-331.

[8] E. Osorio, L. Ferreira, R. Abreu and F. Sousa, "Interoperability in Ambient Assisted Living using OpenEHR," e-Health Networking, Applications \& Services (Healthcom), 2013 IEEE 15th International Conference on, Lisbon, 2013, pp. 394-398.

[9] G. C. Lamprinakos et al., "Using FHIR to develop a healthcare mobile application," Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on, Athens, 2014, pp. 132-135.

[10] Twidle, K., Dulay, N., Lupu, E., Sloman, M.: Ponder2: A Policy System for Autonomous Pervasive Environments. In: IEEE Workshop on Policies for Distributed Systems and Networks, pp. 330-335. IEEE Computer Society, Washington (2009).

[11] A.Gabillon, E.Bruno, Regulating Access to XML Documents, 15th IFIP WG 11.3 Working Conference on Database Security, 2001.

[12] E.Bertino, B.Carminati and E.Ferrari, A Temporal Key Management Scheme for Secure Broadcasting of XML Documents, In Proc. 9th ACM Computer and Communications Security Conference, 2002.

[13] M.Hitchens, and V.Varadharajan, RBAC for XML Document Stores, International conference on Information and Communications security, 2001.

[14] R. Bhatti, J. B. D. Joshi, E. Bertino, and A. Ghafoor, "Access Control in Dynamic XML-based Web-Services with X-RBAC," 2002.

[15] D.W.Chadwick, O.Otenko, The PERMIS X.509 Role Based Privilege Management Infrastructure, In Proc. of ACM Symposium on Access Control Models and Technologies, 2002.

[16] OASIS. eXtensible Access Control Markup Language (XACML) Version 3.0, Jan 2013. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

[17] Wang, Hongbing, et al. "Web services: problems and future directions." Web Semantics: Science, Services and Agents on the World Wide Web 1.3 (2004): 309-320.

[18] FHIR. https://www.hl7.org/fhir/

[19] S. Kabisch, D. Peintner, J. Heuer and H. Kosch, "Optimized XML-based Web service generation for service communication in restricted embedded environments," Emerging Technologies \& Factory Automation (ETFA), 2011 IEEE 16th Conference on, Toulouse, 2011, pp. 1-8.

[20] Wang, Fangju, Jing Li, and Hooman Homayounfar. "A space efficient XML DOM parser." Data \& Knowledge Engineering 60.1 (2007): 185-207.

[21] Gajanayake, Randike, et al. "Accountable-eHealth systems: The next step forward for privacy." Electronic Journal of Health Informatics 8.2 (2013): 11.

[22] Al-Hamdani, Wasim A. "Cryptography based access control in healthcare web systems." 2010 Information Security Curriculum Development Conference. ACM, 2010.

[23] Leyla, Nazia, and Wendy MacCaull. "A Personalized Access Control Framework for Workflow-Based Health Care Information." Business Process Management Workshops. Springer Berlin Heidelberg, 2011.

[24] Zhang, Xinwen, Jaehong Park, and Ravi Sandhu. "Schema based XML security: RBAC approach." Data and Applications Security XVII. Springer US, 2004. 330-343.

[25] Box, Don, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, and Dave Winer. "Simple object access protocol (SOAP) 1.1." (2000).

[26] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1999). Hypertext transfer protocol--HTTP/1.1. Chicago.

[27] Li, Ming, Yu, Shucheng, Ren, Kui, and Lou, Wenjing. "Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings". Security and Privacy in Communication Networks: 6th Iternational ICST Conference, SecureComm 2010, Singapore, September 7-9, 2010. 89-106.

[28] WS-Policy.Web services policy framework (WS-Policy) Version 1.5, Nov 2010. http://www.ibm.com/developerworks/library/j-jws18/.

[29] Moritz Y. Becker, C´edric Fournet, and Andrew D. Gordon. Secpal: "Design and semantics of a decentralized authorization language". Journal of Computer Security,18(4):619–665, December 2010.

[30] Becker, Moritz Y., Cédric Fournet, and Andrew D. Gordon. "SecPAL: Design and semantics of a decentralized authorization language." Journal of Computer Security 18.4 (2010): 619-665.