An Ontology of a Computing Device Secured
with Trusted Platform Module 2.0
Jiun Yi Yap and Allan Tomlinson

Technical Report
RHUL–ISG–2016–1
27 Jan 2016

Information Security Group
Royal Holloway University of London
Egham, Surrey, TW20 0EX
United Kingdom
Abstract. The ability of an ontology to describe trustworthy computing devices with a standard vocabulary enables experts to share a common understanding of such technologies with developers. To this end, we research on and contribute an ontology for describing a computing device secured with Trusted Platform Module (TPM) version 2.0. This ontology represents information about the capability of the computing device, TPM 2.0 and notions of security and trust. They are characterized at different abstract level. It is based on the Web Ontology Language (OWL) and recorded in Resource Description Framework (RDF) / eXtensible Markup Language (XML) format. A developer can review the ontology based description by asking competency questions. Finally, we demonstrate how to translate the competency questions into SPARQL queries.

1 Introduction

An ontology defines a common vocabulary for researchers who need to share information in a domain [1]. It includes machine-interpretable definitions of basic concepts in the domain and the relations among them. In recent years, the ontological approach has been used to build a common understanding between experts and developers of information security technologies. M. Karyada et. al. propose to use ontology to capture and depict security experts' knowledge [2]. In this way developers can exploit security expertise in order to make design choices that will help them fulfill security requirements more effectively.

In this paper, we will use an ontological approach to characterize security rooted in hardware. The aim is to achieve a common understanding between security experts who established such technologies and the developers who make use of these technologies. As the field of trustworthy computing covers numerous security hardware technologies, we framed our research effort around the capabilities of a computing device secured with Trusted Platform Module. This is to ensure meaningful results can be obtained with reasonable amount of resources. The TPM is a device that is specified by the Trusted Computing Group (TCG). The TPM is designed to improve the trust in computing devices by offering certain functionalities such as cryptographic engine, secure storage, digital certification and trusted reporting of the identity and state of its host computing device. TPM 2.0 is the latest specification from TCG [3].

The outcome of our research is an ontology that characterize the capabilities of a computing device secured with TPM 2.0. The main feature of this ontology is that it can be used to describe, in a way that is easily comprehended and machine readable, the capabilities of such a computing device at various levels of abstraction. In addition, a developer can review the ontology by asking competency questions and these questions can be translated into SPARQL queries.

The rest of this paper is organized as follows. Section Two of this paper gives the background to the use of ontologies. This is followed by Section Three that describes the ontology development and explains the structure. Section Four
demonstrates how the ontology can be queried. Section Five reviews the related works and the conclusion to this paper is in Section Six.

2 Background

2.1 Characterizing the capabilities

On how to characterize the capabilities of a computing device secured with TPM 2.0, we referred to the technical models described in the National Institute of Standards and Technology Special Publication 800-33 [4]. These models encompass information at several levels. They range from high level security objectives to low level specific technical details. For example, the low level technical primitive of cryptographic key management is described as enabling the capability of access control enforcement which in turn supports the security objective of confidentiality. The SP800-33 publication is intended to provide a description of the technical foundations that underlie security capabilities. Since our intentions are broadly aligned, we will frame our ontology of a computing device secured with TPM 2.0 on the structure of the technical models described in this publication.

2.2 Purpose of ontology

Ontology development involves giving explicit formal specifications of the terms in a domain and the relations among them [5]. Plainly, an ontology structures the information into classes and instances while their relationships are termed as properties. Classes refer to general concepts while an instance denotes a specific case. For example, cryptographic algorithm is a class and asymmetric key algorithm is a subclass. Then, RSA is a subclass of asymmetric key algorithm and that RSA with key size 2048 is an instance of this subclass. In an ontology, a property refers to the semantic relationships among the classes and instances. For example, the ability of a computing device to produce a digital certificate is enabled by the asymmetric engine of TPM 2.0. Therefore, "is enabled by" is the property that describes the semantic relationship between the production of digital certificate and asymmetric engine.

Kim et al. [6] explain that there is a difference between an ontology and a taxonomy. A taxonomy can only describe and classify concepts based on hierarchical relationships while an ontology can specify semantic relationships among the classes. Referring to the same example of describing a digital certificate is enabled by TPM 2.0 asymmetric engine. There is no simple way for the hierarchical and classification based taxonomy to express this statement. Thus, it follows that an ontology is a more relevant approach because it can describe such statement in a semantic manner. Moreover, the ontological approach would also have contextualized the information. This enables reasoning and deduction at the level of detail which taxonomies cannot provide. At the mean time, we consider this ontological approach as complimentary to formal models such as [7]. An ontology based description benefits at a higher level by providing expressive,
semantic meanings to relationships while a formal model acts at a finer level by giving unambiguous descriptions to these relationships. Furthermore, it is easier to obtain contextualized information from an ontology. This will be discussed in Section Five.

Hence, we can envisage that this approach will produce an ontology of the capabilities of a computing device secured with TPM 2.0. This ontology will have a class structure that depicts computer capabilities by classifying concepts under different classes. This class structure also allows us to describe the concepts at different abstract levels. In addition, the properties of these classes define the relationships among themselves.

2.3 Web Ontology Language

The Web Ontology Language (OWL) is specified by the World Wide Web Consortium (W3C) in 2004 [8]. The language is based on the Resource Description Framework (RDF) [9]. RDF is a language for encoding information on Web pages to assist computers searching for a particular information. An RDF statement revolves around the triple model and it contains a subject, a predicate and an object. Consider the example RDF statement "TPM is specified by TCG". The RDF components for this statement are: the subject is "TPM", the predicate is the words "is specified by" and the object is "TCG". This ability of describing the semantics of data enables RDF to support better discovery of resources and provides more meaningful description of content. OWL extends the RDF schema by providing a more expressive vocabulary and improving on the inference capabilities. Detailed description of OWL can be obtained from [10].

While there are other ontology languages such as the DARPA Agent Markup Language [11] and the Ontology Interface Layer [12], they have been superseded by OWL. Hence we reviewed OWL and found that the RDF triple model is especially useful for representing the classes and their relationship in an uncomplicated manner. In addition, OWL is designed to be compatible with the eXtensible Markup Language (XML) and this allows an OWL based ontology to be processed by the wide range of XML and RDF tools that are already available. This ability can be leveraged upon when we wish to query an ontology.

3 An Ontological Approach

3.1 Developing an ontology

We followed the methodology described in [13] to create this ontology. This methodology ensures that the developed ontology is fit for use in its intended application. In this work, we plan to use an ontology to characterize the capabilities of a computing device. This ontology is examined by a developer to determine if a design of computing device meets certain trustworthiness requirements. Thus, this understanding is crucial to guiding the decision making during the ontology development.
The first step is to define the domain and scope in which this ontology will be used. We aim to represent at different levels of abstraction the description of the capabilities of a computing device secured with TPM 2.0. This covers trust primitives offered by TPM 2.0, capabilities and security objectives and their semantic relationships. A key activity in this step is the crafting of a set of competency questions. Competency questions refer to those questions the ontology should be able to answer when an entity wish to understand the capabilities of the trustworthy computing device. These questions will later serve to verify if the developed ontology suits the intended application.

There are generally two types of competency questions. The first type refers to exploratory questions which the developer can ask to obtain a broad understanding of the computing device. The second type refers to specific questions on technical details. The following are examples of some of the possible competency questions:

**Exploratory Questions**

- What are the properties of the notion of confidentiality?
- Which TPM 2.0 capability enables the device capability of access control?
- Which TPM 2.0 subsystem does the TPM 2.0 capability of protected location uses?

**Specific Questions**

- Is TPM 2.0 capability of attestation using the asymmetric engine?
- Is AES one of the symmetric key algorithms used by this TPM 2.0?

The next step is to consider if an existing ontology can be reused. We studied the ontologies reviewed by Singh et al. [14] but we could not identify any ontology that focused on either trusted computing or TPM. We also undertook online searches and we could not find such ontology. Thus, we believe that our endeavor is the first of its kind.

In step 3, we enumerated all the terms of TPM 2.0 that we considered to be relevant to the purpose of this ontology. This step was non-trivial as it took considerable time to acquire knowledge by understanding TPM 2.0 specifications [5], the book A Practical Guide to TPM 2.0 [15] and Trusted Computing Platform, TPM 2.0 in context [16]. This knowledge is supplemented by our experience from working on TPM 2.0 research projects and from our interaction with key personnel who have been involved in the development of TPM 2.0 specifications. Meanwhile, we investigated about trust notions from our previous socio-technical study on user centered trust notions in the domain of Information Technology [17]. We also studied the security objectives listed in SP800-33. We believe that security and trust are not orthogonal. Trusted components are used to build secure systems. On the other hand, security properties are often evaluated as part of the process of establishing trust in a computing system. This association of trust and security is articulated in the Trusted Computer System Evaluation Criteria (Orange Book) of the United States of America Department of Defense in 1985 [18]. Therefore, our ontology will include both security and trust notions.
However, this step generates a long list of terms and we decided to put aside those that are not relevant to the purpose of this ontology. For example, there are terms such as power detection, execution engine, startup, shutdown and self-test which relate to the internal operations of TPM. These terms are not included in the ontology because they do not enable any capabilities of the computing device.

The remaining steps created the TPM 2.0 ontology. The steps are - defining classes and class hierarchy, define class properties and their values and creating instances. These steps are best illustrated together with the ontology and they are described and explained in the next section.

3.2 An Ontology of a Computing Device Secured with TPM 2.0

The development of classes, their hierarchy and properties are closely intertwined. We adopted a top down approach to develop the class hierarchy. The list of terms we obtained from step 3 were scrutinized and we defined the most general concepts. These general concepts are considered as classes. Then we zoomed in and identified those terms that belong to these classes. These terms were considered as subclasses. We reviewed the subclasses to check that they do not refer to the same concept. This avoided repetitions and ensure that there were not further groupings. For example, TPM 2.0 sub system is a class and it has asymmetric engine as a subclass. Asymmetric engine further contains the algorithms RSA, ECC and SM2 as subclasses. Next, we leveraged upon our knowledge of TPM 2.0 and created properties that describe the semantic relationships between the classes. These steps were repeated to refine the ontology. After several exercises of sorting the terms into meaningful classes and ensuring that the ontology based description could meet our requirements, we derived 4 different classes.

The 4 classes are:

- TPM 2.0 Capability: this describes the capabilities a TPM 2.0 can offer to the host computing device. The capabilities are defined as subclasses of this class.
- TPM 2.0 Subsystem: this describes the subsystems of the TPM 2.0. Each of the subsystems is defined as a subclass.
- Security and Trust Notion: this is an amalgamation of user level trust notions and security objectives. Each of the notions is defined as a subclass of this class.
- Device Capability: this describes at the abstract level what trusted functions are offered in a particular computing device. Some examples of trusted functions are defined as subclasses in this ontology.

The ontology presented in this paper only consists of classes, subclasses and their relationships. Specific configuration of different computing devices and their TPM 2.0 will be defined as instances of these subclasses. They can be populated later to give a more detailed description to a particular computing device. We
Fig. 1. Graphical representation of the ontology based description.
use OWL as the language for our ontology and this ontology is created using Protege [19]. Protege is an open source platform that provides a suite of tools to construct ontologies. We chose Protege because it supports RDF and XML format and the ontology can be queried using SPARQL, an RDF data access language [20]. Figure 1 shows the graphical representation of these classes.

The core class in this ontology is the \textit{TPM2.0Capability} class. This class serves as the linkage between low level trust primitives described in \textit{TPM2.0SubSystem} class and high level device capability description and user notions. It mirrors the support services described in SP800-33 and characterizes the capabilities of TPM 2.0 which enable many trusted functions of the computing device. It consists of the subclasses \textit{Certification}, \textit{Attestation}, \textit{ProtectedLocation} and \textit{IntegrityMeasurement}. These subclasses were mainly referenced from TPM 2.0 specifications while additional information was gathered from [15] and [16]. We consider them as subclasses because they are general concepts of the capabilities offered by TPM 2.0.

The \textit{TPM2.0Capability} class is supported by primitives that are described in the \textit{TPM2.0SubSystem} class. There are 11 subclasses in \textit{TPM2.0SubSystem} class and they are \textit{AsymmetricEngine}, \textit{SymmetricEngine}, \textit{HashEngine}, \textit{KeyGeneration}, \textit{RNG}, \textit{Authorization}, \textit{NVMemory}, \textit{VolatileMemory}, \textit{ExecutionEngine}, \textit{PowerDetection} and \textit{Management}. We have also defined another level of subclasses to describe the algorithms used in the subclass of \textit{AsymmetricEngine}, \textit{SymmetricEngine} and \textit{HashEngine}. These subclasses were referenced from TPM 2.0 specifications and supplementary information was gathered from [15] and [16]. The motive for \textit{TPM2.0SubSystem} class is to depict the primitives that underlie TPM 2.0 capabilities. Thus, the \textit{TPM2.0Capability} class is mapped to \textit{TPM2.0SubSystem} class by the property \textit{usesTPM2.0subsystem}. We can make use of the property value restriction feature of OWL to indicate that a particular subclass of \textit{TPM2.0Capability} class is only mapped to some subclass of \textit{TPM2.0SubSystem}. We also define \textit{supportsTPM2.0Capability} which is an inverse to the property \textit{usesTPM2.0subsystem}. We have a two way mapping because it will be useful when we query the ontology and wish to transverse through the classes to uncover more information. An example of how \textit{TPM2.0Capability} class is mapped to \textit{TPM2.0SubSystem} class is as follows: The encryption of a protected location uses a symmetric key and this symmetric key is generated from a seed. Meanwhile, the seed is stored in the non-volatile memory of TPM 2.0. Thus, \textit{ProtectedLocation} is defined to have \textit{SymmetricEngine}, \textit{KeyGeneration} and \textit{NVMemory} as its \textit{useTPM2.0subsystem} property values.

\textit{TPM2.0Capability} class is mapped to the \textit{SecurityAndTrustNotion} class to illustrate how TPM 2.0 capability supports the high level security and trust notions. This mapping also serves as the bridge to low level trust primitives. The \textit{SecurityAndTrustNotion} class contains 6 subclasses and they are \textit{Confidentiality}, \textit{Integrity}, \textit{Availability}, \textit{Accountability}, \textit{ExpectedBehaviour}, \textit{Identity} and \textit{Authenticity}. These subclasses are referenced from the security objectives of SP800-33 and the trust notions mentioned in [17]. We consider these notions as subclasses and they can be populated later with instances that identify
specific references. For example, secrecy of user password can be an instance of the subclass Confidentiality. The TPM2.0Capability class is mapped to SecurityAndTrustNotion class via the property supportsSecurityAndTrustNotion. Conversely, we also define isSupportedByTPM2.0Capability which is an inverse to the property supportsSecurityAndTrustNotion. As an example, assume that the high level user trust notion of ExpectedBehaviour is desired and from the ontology based description, we know that the Attestation and IntegrityMeasurement subclass of TPM2.0Capability supports this notion via the property isSupportedByTPM2.0Capability. As we trace the ontology based description further, we can see that Attestation is linked via the property of useTPM2.0subsystem to AsymmetricEngine of TPM2.0SubSystem class while IntegrityMeasurement is linked via the same property to HashEngine and VolatileMemory.

Lastly, we define the DeviceCapability class to demonstrate how low level trust primitives can be mapped to high level device functions. The DeviceCapability class is mapped to TPM2.0Capability class by the property isEnabledByTPM2.0Capability. We also define supportsTrustNotion which is an inverse to the property isEnabledByTPM2.0Capability. In this ontology based description, we defined 3 examples of device functions that are described in [15]. They are SecureDataBackup, AccessControl and IdentityManagement. These subclasses are used as examples here and hence are not fixed. The user of this ontology based description can define more subclasses to describe other functions. As an example, in access control, we want different users to have separate levels of access to a database. This function can be enabled by leveraging on the protected location function of TPM 2.0 which in turn makes use of the authorization subsystem and symmetric key cryptographic engine. Hence, from the ontology based description, AccessControl is mapped to ProtectedLocation of TPM2.0Capability class via the isEnabledByTPM2.0Capability property. In turn, ProtectedLocation is mapped to Authorization and SymmetricEngine of TPM2.0SubSystem class via the usesTPM2.0SubSystem property.

In this section, we introduced the ontology and characterize the constituent classes. We also gave examples of how the ontology can be used to describe the trustworthiness of a computing device by providing a map of low level trust primitives to high level notions and capabilities of computing device. This ontology will serve as an useful foundation whereby experts can further populate the ontology with instances that describe more specific implementation and configuration information.

4 Querying

This section will address how a developer can query the ontology based description to uncover its content. We studied the ontology and its OWL/RDF format and propose to translate the competency questions in Section 3.1 into SPARQL queries. There are two types of questions. The intent of exploratory questions is to uncover information by checking for the content of the components in a RDF triple. The content can be used to infer certain information about the computing
device. For example, we refer to the competency question on the properties of confidentiality. Figure 5 of Appendix A shows the description of Confidentiality class in RDF/XML format. If we wish to know all about its properties and values, a query using the SPARQL function of SELECT and WHERE can be issued. Figure 2 shows the query and table 1 shows the result.

```
SELECT ?property ?value
WHERE
{tpm2.0:Confidentiality rdfs:subClassOf ?object .
?object owl:someValuesFrom ?value }
```

**Fig. 2.** SPARQL query for Confidentiality class.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>isSupportedByTPM2.0Capability</td>
<td>ProtectedLocation</td>
</tr>
<tr>
<td>isBackedByDeviceCapability</td>
<td>SecureDataBackup</td>
</tr>
<tr>
<td>isBackedByDeviceCapability</td>
<td>AccessControl</td>
</tr>
</tbody>
</table>

**Table 1.** Result for SPARQL query on Confidentiality class.

In another example, we refer to the competency question on which TPM 2.0 capability enables the device capability of access control. Figure 6 of Appendix A shows the OWL/RDF description of AccessControl class. For this query, we can make use of the FILTER function of SPARQL. Figure 3 shows the query and table 2 shows the result. Meanwhile, the same query can be altered to obtain answer to the competency question of which TPM 2.0 subsystem does the TPM 2.0 capability of protected location uses.

```
SELECT ?property ?value
WHERE
{tpm2.0:AccessControl rdfs:subClassOf ?object .
?object owl:someValuesFrom ?value
FILTER regex (str(?property), "isEnabledByTPM2.0Capability")}
```

**Fig. 3.** SPARQL query for AccessControl class.

A specific competency question is concerned with the presence of RDF triples. For example, one can check the ontology to determine if the RDF triple (Attestation, usesTPM2.0SubSystem, AsymmetricEngine) exists. This example refers
Table 2. Result for SPARQL query on AccessControl class.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>isEnabledByTPM2.0Capability</td>
<td>ProtectedLocation</td>
</tr>
</tbody>
</table>

to the competency question of asking if TPM 2.0 capability of attestation is using the asymmetric engine. The existence of this triple indicates that the TPM 2.0 capability of attestation uses the asymmetric engine of TPM 2.0 subsystem. Figure 7 of Appendix A shows the OWL/RDF format of the description for Attestation class. We can use the ASK command to query the ontology for specific triple and the command will return true or false depending on if the specific triple can be found. Figure 4 shows the SPARQL query. The result for this query is "True". Meanwhile, the same query can be modified for the competency question "Is AES one of the symmetric key algorithms used by this TPM 2.0?"

\[
\text{ASK} \{ \text{tpm2.0:Attestation rdfs:subClassOf } ?\text{object} . \\
?\text{object owl:onProperty tpm2.0:usesTPM2.0SubSystem} . \\
?\text{object owl:someValuesFrom tpm2.0:AsymmetricEngine} \}
\]

Fig. 4. SPARQL query for Attestation class.

In this section, we only discuss the queries that can be used to help to answer the example competency questions posted in section 3.1. This discussion will serve as an introduction and there can be many other questions that the ontology can answer using the powerful SPARQL tool. In the mean time, the ability of the ontology described in figure 1 to answer these questions validate that its structure and content are suitable for use by a developer to uncover information about the design of a computing device secured by TPM 2.0.

5 Related Works

Kim et al. noted that annotation with security related metadata enables discovery of resources that meet security requirements [6]. They presented the Naval Research Laboratory (NRL) Security Ontology which focuses on the annotation of security mechanisms, protocols, algorithms, credentials and objectives. The NRL Security Ontology consists of the core class of Security Concept. It has subclasses of Security Protocol, Security Mechanism and Security Policy. The Security Concept class is defined to support the Security Objective class. The other classes of the NRL Security Ontology include Security Algorithms, Security Assurance, Credentials, Service Security, Agent Security and Information Object. The authors explained that the classes of Service Security, Agent Security and Information Object classes are extensions of the DAML Security
Ontology [7]. On the other hand, the Credentials, Security Algorithm and Security Assurance classes provide values for properties defined for concepts in the Security Concept class. As the authors applied this ontology to a Web Service Oriented Architecture, the Information Object class was added to allow for the annotation of web service inputs and outputs. In the paper, the authors also described an algorithm to perform matching of security capabilities to security requirements. Although our ontology has the same intent and approach as the NRL Security Ontology, we differ in the domain of the ontology. Our domain is to describe the capabilities of a computing device secured by TPM 2.0. We also explicitly explain how we can use SPARQL to query the ontology for answers to the competency questions. On the other hand, the NRL Security Ontology described how the matching algorithm works but was not clear on how the algorithm can be implemented.

6 Conclusion

We have produced details on how an ontology of a computing device secured with TPM 2.0 is developed and explained the contents and arrangements of the classes and their properties. We have also given sample competency questions that should be answered by this ontology and use SPARQL to demonstrate how to obtain answers to the competency questions. Thus, from this work, we have gained insights into how an ontology can be used as a standard vocabulary for experts to share information on TPM 2.0 with developers of computing devices.

References

10. OWL Web Ontology Language Overview, http://www.w3.org/TR/owl-features/

A Appendix
<owl:Class rdf:about="&tpm2.0;Confidentiality">
  <rdfs:subClassOf rdf:resource="&tpm2.0;SecurityAndTrustNotion"/>
  <rdfs:subClassOf><owl:Restriction>
    <owl:onProperty rdf:resource="&tpm2.0;isSupportedByTPM2.0Capability"/>
    <owl:someValuesFrom rdf:resource="&tpm2.0;ProtectedLocation"/>
  </owl:Restriction></rdfs:subClassOf>
  <rdfs:subClassOf><owl:Restriction>
    <owl:onProperty rdf:resource="&tpm2.0;isBackedByDeviceCapability"/>
    <owl:someValuesFrom rdf:resource="&tpm2.0;Confidentiality"/>
  </owl:Restriction></rdfs:subClassOf>
  <rdfs:subClassOf><owl:Restriction>
    <owl:onProperty rdf:resource="&tpm2.0;isBackedByDeviceCapability"/>
    <owl:someValuesFrom rdf:resource="&tpm2.0;AccessControl"/>
  </owl:Restriction></rdfs:subClassOf>
  <rdfs:subClassOf><owl:Restriction>
    <owl:onProperty rdf:resource="&tpm2.0;isEnabledByTPM2.0Capability"/>
    <owl:someValuesFrom rdf:resource="&tpm2.0;ProtectedLocation"/>
  </owl:Restriction></rdfs:subClassOf>
</owl:Class>

Fig. 5. Description of Confidentiality class in RDF/XML format.

<owl:Class rdf:about="&tpm2.0;AccessControl">
  <rdfs:subClassOf rdf:resource="&tpm2.0;DeviceCapability"/>
  <rdfs:subClassOf><owl:Restriction>
    <owl:onProperty rdf:resource="&tpm2.0;backsTrustNotion"/>
    <owl:someValuesFrom rdf:resource="&tpm2.0;Accountability"/>
  </owl:Restriction></rdfs:subClassOf>
  <rdfs:subClassOf><owl:Restriction>
    <owl:onProperty rdf:resource="&tpm2.0;backsTrustNotion"/>
    <owl:someValuesFrom rdf:resource="&tpm2.0;Integrity"/>
  </owl:Restriction></rdfs:subClassOf>
  <rdfs:subClassOf><owl:Restriction>
    <owl:onProperty rdf:resource="&tpm2.0;backsTrustNotion"/>
    <owl:someValuesFrom rdf:resource="&tpm2.0;Confidentiality"/>
  </owl:Restriction></rdfs:subClassOf>
  <rdfs:subClassOf><owl:Restriction>
    <owl:onProperty rdf:resource="&tpm2.0;isEnabledByTPM2.0Capability"/>
    <owl:someValuesFrom rdf:resource="&tpm2.0;ProtectedLocation"/>
  </owl:Restriction></rdfs:subClassOf>
</owl:Class>

Fig. 6. Description of AccessControl class in RDF/XML format.
Fig. 7. Description of Attestation class in RDF/XML format.