Improving Security Assurance of Services through Certificate Profiles

Marioli Montenegro, Antonio Maña, Hristo Koshutanski

Escuela Técnica Superior de Ingeniería Informtica
Universidad de Málaga (Spain)
{marioli,amg,hristo}@lcc.uma.es

Abstract. Cloud and Web Services technologies offer a powerful cost-effective and fast growing approach to the provision of infrastructure, platform and software as services. However, these technologies still raise significant concerns regarding security assurance and compliance of data and software services offered. A new trend of a service security certification has been recently proposed to overcome the limitations of existing security certificates by representing security certification in a structured, machine-processable manner that will enable automated reasoning for certified security features in security-critical domains. However, the richness and flexibility of the underlying certificate models and languages comes with the price of increased *complexity* in processing and comparing those certificates and related security claims in practice. In this paper, we propose the concept of *certificate profile* to provide a mechanism to address processability and interoperability of service security certificates. We present a conceptual model and a concrete realization of the model within the context of the European project ASSERT4SOA.

1 Introduction

Service Oriented Computing (SOC) has facilitated a paradigm shift in software provisioning models, such as Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), providing enormous benefits [1]. However, lack of security assurance of third-party services is hampering their wider adoption in business- and security-critical domains. In traditional software provisioning models, security certification of software by trusted third party entities is used to provide *security assurance* to consumers. Certification schemes such as Common Criteria [2] are well established and quite successful in providing the required security assurance to consumers. Thus, software *compliance* to established security certification criteria will provide certain guarantees on security assurance of that software.

However, applying security certification as is to SOC is infeasible. A key obstacle being the natural language representation of the certificates, that requires manual inspection, preventing their usage in typical SOC scenarios like service discovery, selection, and composition. To overcome the limitations of existing

security certificates, and facilitate adoption of security certification in security-critical domains, the concept of *service security certification* has been proposed [3–5]. Consequently, an outcome of a service security certification is a security certificate of a service. A security certificate is realized by a language that enables the representation of a certificate in a structured, machine processable manner that would enable automated reasoning to be performed on them and thus make it feasible for certified security features to be part of any SOC scenario [6].

Given the complexity of the service provisioning models, the languages describing security certificates are expected to cover a rich set of fields and structures that enables the representation of processes and results of different services security certification activities. For example, representing certification artefact for cloud-based services would require complex and rich representation of underline security properties and evidences supporting those properties. Therefore, languages provide users with different representation alternatives and structural choices that are necessary to accommodate the heterogeneity of the processes and results of certification.

However, this flexibility and expressiveness comes with the price of an increased difficulty in determining the semantic soundness of a certificate with respect to the certification that is the origin of such certificate, and places higher complexity on the process of comparing certificates. As a side effect, security assurance of services provided by certification activities may not face expected adoption and success given the complexity in processing and comparing security certificates, thus making impractical any sort of automated reasoning to be performed on them, and consequently neglect an adequate scalability of any service selection based on certified security features.

For service consumers, the possibility to compare the certified security features of a service with their security requirements is an important aspect during service discovering, selection and composition process. The integration of non-functional security aspects of services with other non-functional properties (such as performance and scalability) can be well handled on the level of service query language and the corresponding service selection logic [7].

We propose the use of a concept of *certificate profile* to provide a mechanism to address processability and interoperability of service security certificates. There are three main use cases where the certificate profile plays a key role:

(i) *Facilitate comparison among security certificates.* Given the flexibility and richness of certificate languages and ability to express similar security assertions in a different way, a certification authority may wish to define a certificate profile (e.g., by defining various certificate structure and content mandatory) to enforce uniformity of content of certificates when issued by accredited entities.

(ii) *Facilitate production of security certificates compliant to specific certification criteria.* Given that a certificate language can support various certification schemes, a certification authority has to define its certification criteria in a certificate profile, so that all issued security certificates will conform to the criteria defined by the certificate profile.

(iii) *Enable consumers to specify their security requirements for the services.* Similarly to CC-PP [8], the consumers or consumer groups may wish to define a certificate profile with domain-specific security requirements (criteria). When services conform to such certificate profiles, it eases the decision making process for the consumers as the conformance to a profile implies that their requirements are met by the service.

The rest of the paper is organized as following. Section 2 presents related work on security certification of services. Section 3 introduces the concept of certificate profile and its structure. Section 4 presents the core of profile-based management of security certificates. Section 5 describes a proof-of-concept realization of certificate profile within a European project ASSERT4SOA. Section 6 concludes the paper and outlines future work.

2 Related Work

Security Certification Schemes: There are quite a few established and successful schemes such as Common Criteria for Information Security (CC), Commercial Product Assurance (CPA) and so on. Security certification schemes can be broadly classified based on the domains that they are applicable in, the recognition of the certification schemes, the descriptive or normative character of the issued certificates and so on. Among the existing schemes, CC is a widely recognized, used descriptive certification scheme. The CC scheme avoids an all or nothing benchmark, by providing security assurance at varying levels, called Evaluation Assurance Levels (EAL). This provides flexibility for product vendors to get their product certified at lower assurance levels and improve the EAL over time. The CC scheme is primarily "claims" based, where the vendor makes claims about the security functionalities in the product in a document called "Security Target" (CC-ST) [2]. However, consumers can specify their requirements in a document called "Protection Profile" (CC-PP), and vendors can build products that conform to a CC-PP (and claim conformance in the CC-ST).

However, in practice, the comparison of products having different "claims" can be very hard. This is due to the representation of the CC-related documents (CC-PP, CC-ST) in natural language, which is often filled with legalese and heavy security jargon making it rather complex to understand for non-security experts. Hence, it becomes quite difficult to determine if a particular product satisfies a consumer's security requirements and to compare different products against their requirements.

Digital Security Certificates: The resulting security certificates from current security certification schemes are not represented in a digital format. Though there are a few "digital security seals" such as the TRUSTe privacy seal [9], McAfee SECURE seal [10] and so on. These seals are normative statements regarding the security feature of an entity, which can be seen as a step towards digital security certificates, but cannot provide any meaningful assurance to consumers as they do not contain any information regarding the certified entity.

There are several digital certificate standards for identity and authorization management used in SOC, such as X.509 [11] and SAML [12]. Both standards support public-key (identity) certificates and attribute certificates for purposes of user authentication and authorization. These certificates are used as a means to gain a *security functionality* (such as authentication and authorization) and are quite different from the notion of digital security certificates used to provide *security assurance*.

Security Certification of Web Services: The wide spread adoption of Service-Oriented Architectures (SOAs) and Software-as-a-Service (SaaS) provisioning model enables large-scale heterogeneous ICT infrastructures be dynamically built from loosely coupled, well-separated services, where key non-functional properties like security, privacy, and reliability are of increased and critical importance. In such scenarios, certifying service's security properties will be crucial. Today's certification schemes do not provide, from an end-user perspective, a reliable way to assess the trustworthiness of composite services in the context where (and at the time when) these will be actually consumed.

ASSERT4SOA project [5] is filling this gap by producing novel techniques and tools fully integrated within the SOA lifecycle for expressing, assessing and certifying security properties for complex service-oriented applications. The purpose of ASSERT4SOA is to provide a framework for handling Advanced Security Service Certificates, called ASSERTs. The originality of these new ASSERT certificates resides in the embedded abstractions security properties, targets of certification, evaluation-specific results (such as formal model-based, or test-based), validation algorithms, and service binding mechanisms.Therefore, when an ASSERT certificate is bound to a service, the service consumer will benefit from an insight on the security capabilities of the service, going well beyond the information conveyed by existing digital certificates (refer also to Section 5).

Security Certification of Cloud Services: Cloud technology offers a powerful and fast growing approach to the provision of infrastructure (IaaS), platform (PaaS) and software (SaaS) as services. However, despite its appeal, cloud technology still raises significant concerns regarding the security, privacy, governance and compliance of data and software services offered through it. Such concerns arise from the difficulty to verify security properties of the different types of services available through clouds and the uncertainty of the owners and users of such services about the security of their services once the services are uploaded and offered through a cloud. This difficulty stems from the fact that the provision and security of a cloud service is sensitive to potential interference between the features and behaviors of all the inter-dependent services in all layers of the cloud stack, as well as dynamic changes in them.

CUMULUS project [4] proposes a research program whose aim is to address these limitations by developing an integrated framework of models, processes and tools supporting the certification of security properties of infrastructure (IaaS), platform (PaaS) and software application layer (SaaS) services in cloud using multiple types of evidences regarding security, such as service testing, monitoring

and trusted computing proofs, and based on models for hybrid, incremental and multi-layer security certification.

Service Security Certification and SLAs: The concept of Service Level Agreement (SLA) was introduced with an objective similar to the one of our proposal. SLAs provide means for service providers to declare explicitly claims about "quality" aspects of their services. SLAs can be used to inform users about different aspects of a service such as performance, limitations of use, security, etc. There are many scenario in which these provider-backed claims are enough for clients. However, there are also other scenarios in which clients need additional assurance provide by trusted external entities. In these cases, SLAs do not suffice and other mechanism are required in order to establish the necessary trust between elements and services. It is important to note that our proposal is not an alternative to SLAs, but much more a complement. In fact, an important application of security service certificate is their use in conjunction with SLAs. For example, by using WS-Agreement [13], a widely used SLA standard, a service provider can provide (claims) non-functional security properties to potential consumers described via an *agreement template* specifying the service and its guarantees including the security properties provider's services are certified for. Thus, service consumers will gain additional level of security assurance provided by the service security certificates to the trust in the claims stated by the service provider on the security aspects of his services. Other approaches define SLAs to enable specification of trust relationships used to derive service interactions enriched with security functionality such as authentication and non-repudiation [14]. These approaches focus on specifying security functionality of services but not on specifying security assurance of services.

3 Certificate Profile

The main goal of a certificate profile is to provide suitable means for creation of certificates by ensuring semantic uniformity of certificates for a specific (domain of) certification capturing any certification scheme of expertise, evaluation specific expertise, products certified, specific vocabulary of use for expressing security aspects of certified products, and other certification artefacts relevant to defining the semantics of certificates.

3.1 Profile Structure

A certificate profile is a mechanism to specify the contents and semantics of a class of security certificates. A certificate profile is composed of three parts: (i) *Certificate Template*: specification of the common structure and the values of specific fields mandatory for a given certificate class, (ii) *Semantic Rules*: specification of the semantics of the certificate class in the form of semantic rules, and (iii) *Vocabulary*: specification of vocabulary terms (ideally ontology-referenced terms) providing restrictions on use of vocabulary for language artefacts of security certificates of the given certificate class.

Fig. 1. Certificate Profile Structure

Figure 1 shows the abstract structure of the certificate profile. The three profile components provide certificates content uniformity in three different dimensions: certificate template ensures structural uniformity; semantic rules ensure integrity of intended semantics of certification; while certificate vocabulary ensures common ontology-based ground of terms and ranges of possible values of certification (in a given domain).

Certificate Template The certificate template is a partially filled certificate that establishes the common structure and content of all certificates created based on a certificate profile. Therefore, any certificate conformant to a profile must include the fields, structure and values defined in the template of the profile. A certificate template specifies an incomplete certificate structure with respect to a given certificate syntax (e.g., XML schema). It is used as baseline for creating new certificates.

Alternatively, a certificate template can be considered as a set of implicit (semantic/integrity) rules. These rules are simple and easy to understand. For this reason, it is not required to represent a template as a set of rules, but used as a certificate template - a more intuitive notion for expressing predefined structure and values of profile elements. We have defined some high-level interpretation rules for any certificate template structure:

(i) If a template defines a certificate artefact instance but with an empty content (value), the resulting certificate must have the identified artefact as part if its structure with possibly any (syntactically valid) structure or content inside.
(ii) If a template defines a certificate artefact instance but with certain content (value), the resulting certificate must have the identified artefact instance as part if its structure and the same value determined by the template.
(iii) If a template defines N number instances of a specific certificate artefact (if certificate syntax allows) where each instance with specific structure and content, the resulting certificate must have at least the same number of the certificate artefact instances each one with the same structure and content as defined in the template.

If we want to enforce the existence of a certificate artefact but with an empty structure one can achieve that by using rule (i) defining the artefact with empty content in a template, and by using a semantic rule that enforces, restricts or checks whether the given artefact has an empty content (value) in a resulting certificate structure.

The goal of rule (iii) is to allow a template to predefine multiple instances of a certificate artefact each one with specific structure and content. For example,

a template may define two instances of a certificate artefact TypeSpecificEvaluation, the first one defining some specific structure and content of test-based service evaluation with mandatory test cases, while the second instance defining formal model based service evaluation under a specific formal model language.

Semantic Rules The Semantic Rules define semantic constraints and dependencies between content of certificate artefacts within a given class of certificates. While the implicit rules defined by the certificate template are enough for structure-wise restrictions (requiring an optional element be mandatory, constraining specific structure or content of certificate artefacts, etc.), there are cases where more complex restrictions are needed. Some examples of more complex rules can be (but not limited to):

(i) Artefact dependencies: define presence or content of an artefact depending on the presence or content of another artefact.
(ii) Artefact content constraints: restrict an artefact content within a range of acceptable values, or restrict artefact content as a function of the content of other artefacts.

Semantic rules represent a solution, allowing to formulate rules to ensure integrity of an intended semantics of a given certificate class, i.e., preserving specific semantics of certification artefacts. Semantic rules can be formulated in rule based languages (such as Schematron [15] or variants of OCL [16]) or imperative languages (such as Java or Javascript) in function of the underlying certificate language and supported implementation. The choice of a language for expressing semantic rules has an important implication to achieve machine processability and reasoning of the rules. The language should allow rich fine-grained expression of *patterns* over certificates content and structure.

Some examples of rules are the following:

(i) The content of an artefact TargetOfCertification must be of type one of "Software-as-a-service" or "Platform-as-a-service";
(ii) A security property definition artefact and the property formal model definition artefact of model-based evaluation must use the same abstract security property category (e.g., "Confidentiality");
(iii) Restrict the certification of security mechanisms to a pre-defined set of mechanisms for a given application domain. For example, in the domain of eHealth a profile can define by semantic rules that all confidentiality properties on storage services must be certified based on the evaluation of the use of AES block cipher [17] with the approved modes of operation [18].

Certificate Vocabulary The certificate vocabulary part of the profile provides a means to define and restrict use of vocabularies on different certificate artefacts. One of the goals of the vocabulary part is to enable specific per profile (i.e., per a class of certificates) integration of the underlying certificate language with different ontology terms coming from different domains of knowledge. In

that way, ontology integration will enhance the semantic robustness among all certificates conformant to a given profile and even among certificates conformant to different profiles, which have been diminished by flexibility and openness of security certificate languages (models). Ontologies provide not only a source of semantically defined terms but also provide means to define relations between terms, and equivalences between different terms. That gives us a powerful way to query ontologies for different aspects of certification and related semantics.

Restricting the range of values of certificate artefacts to terms defined in ontology will make all certificates conforming to the given profile processable and comparable on those artefacts, as their values are ontology terms with defined semantics and relations among them. For example, a vocabulary used for a certificate artefact named AbstractSecurityProperty can be restricted to one of "Confidentiality", "Integrity" or "Availability" (also known as CIA triad of core attributes of information security), and other properties could be ontology-modeled using meta-data or relationships between information. For example, non-repudiation can be viewed as a property related to integrity of relationship between information and information issuer.

Similarly to the certificate template and semantic rules, one can see the certificate vocabulary section of the profile as a set of implicit rules each one restricting use of vocabulary for certificate artefacts. However, by defining explicit vocabulary section we have, first a more intuitive notion for expressing vocabulary restrictions and, second enable the use of dynamic values based on queries over ontologies, which otherwise would be difficult to achieve as semantic rules.

The certificate vocabulary section enables the use of *static* or *dynamic* vocabularies. A static vocabulary defines actual terms inside a profile. It is suitable for offline processing, but could be out-dated by an ontology evolution/update. In contrast, a dynamic vocabulary defines actual terms by means of a query over ontology, which requires Internet connection for online processing. Ontology queries will be executed at the time of use of a given profile, i.e., the actual terms (values) will be dynamically retrieved from ontology when the profile is used. Static vocabulary provides a means to define ontology terms or just terms without any ontology context to be used statically without subject to further refinements/changes.

By the time being, we limit the use of either static or dynamic type vocabularies per certificate artefact, but not both types. Our main motivation is to provide a consistent vocabulary solution across all certificates during the lifetime of a certificate profile. If one specifies both types vocabulary per artefact, assuming dynamic vocabulary takes precedence over static vocabulary, there could be a case where ontology evolves (e.g., removing some terms or redefining those) in a way that makes the static part of the vocabulary inconsistent with respect to the actual values in ontology. Then, in the case of offline use of the profile, certificates will be created considering the static vocabulary, which will be inconsistent with those certificates created based on the dynamic vocabulary. This aspect may significantly decrease processability of certificates conformant to the profile given that the ontology of the dynamic vocabulary gives the seman-

tics (interpretation/reasoning) of the vocabulary terms when used to process or compare the corresponding certificate artefacts.

It is the responsibility of the issuer of a certificate profile to ensure that any domain ontology used as part of the certificate profile is consistent with the overall vocabulary of the profile. We assume that certification authorities produce *well-formed* certificate profiles with consistent vocabulary definitions.

An issuer of a profile may decide to enforce or not the use of vocabularies. When a vocabulary specification is defined mandatory the referenced language artefact must have a value from the vocabulary. If a vocabulary is optional the referenced language artefact should have a value from the vocabulary.

4 Profile-based Certificate Management

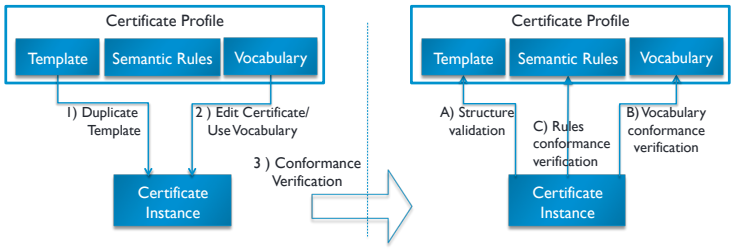
We will describe two core certificate management operations based on profiles: profile-based creation of certificates, and the opposite one, profile conformance verification of certificates. The former facilitates certification authorities, certificate issuers or even service providers/owners (in case of self-signed certificates) in creation of certificates conformant to a profile, while the latter operation will facilitate service consumers be that developers or system designers during a service-based system development lifecycle.

For example, during system design to discover relevant services a client can query a service repository for functional and non-functional security aspects for services of interests [7]. However, given the openness and flexibility of certificate language artefacts in expressing security properties and related evidences supporting those, the client would be much more interested in referring to a certificate profile along the query to the repository in order to restrict (not use the entire variety of) security assertions to a limited subset of those specified by a certificate profile. In that way, certificate profiles enable much more effective and practical comparison of security aspects of services during a service discovery phase, where matching and discovery of non-functional security aspects is reduced to matching within those services with security certificates conformant to a profile. Certificate profiles provide an important step towards a fully automated security assessment of non-functional security aspects of certified services.

There are also other relevant aspects of profile-based certificate management that can occur during a service composition phase and during runtime system adaptation, where service replacement is achieved not only based on functional service aspects but also if non-functional security properties are preserved by the new service [19]. In this case, certificate profiles can be well used to verify if the new replaced service is certified conforming to a given certificate profile specifying the required security assertions.

4.1 Profile-based Creation of Certificates

Given that a certificate language can support various certification schemes, profile-based certificate creation process will facilitate production of security



(a) Profile-based Certificate Creation (b) Profile Conformance Verification

Fig. 2. Profile-based Certificate Management

certificates compliant to specific certification criteria. Figure 2(a) shows the profile-based creation process. Prerequisite to the creation process is the discovery or selection of a certificate profile specifying domain specific security aspects relevant to the certification process a service has to undergone. Once the profile is selected and loaded, all dynamic vocabulary specifications (e.g., ontology queries) are processed. If some dynamic vocabulary specifications depend on other artefacts and values in order to be processed, these vocabularies should be processed at the time when the issuer creates the corresponding artefacts.

Once the profile is processed, first a duplicate of certificate template is done, and a certificate instance is created with an initial structure and content of the duplicated template data. Next step is the actual process of editing the certificate artefacts and creation of new artefacts as needed by the issuer. This step heavily relies on the use of certificate vocabulary defined in the profile. When an artefact's vocabulary is specified as mandatory, the process should enforce the choice of the vocabulary terms. Otherwise, if optional, the process should recommend, suggest a choice of terms but leaving the issuer to specify own terms when he finds necessary. Third step of certificate creation process, the final certificate instance is verified for conformance to the profile (presented in the next subsection). All non-properly used artefacts and corresponding vocabularies will be reported. Step 3 will give a feedback to redo step 2 of the creation process by repeating it until the certificate instance conforms to the profile.

4.2 Profile Conformance Verification of Certificates

The conformance verification process described can be generally used to verify a certificate for profile conformance, and not only as part of the certificate creation process. Figure 2(b) shows the three main steps of conformance verification process. There is always a validation step taking place before the conformance verification process, validating if the certificate instance conforms to the syntax of a given certificate model, that is, if the certificate instance is a syntactically valid certificate. Otherwise, the verifier should not proceed with the verification process. If the certificate instance is a valid certificate, the first step of conformance verification is a certificate structure validation against the template part

of the given profile. The certificate structure is validated if it contains all the required artefacts and artefacts' content as defined in the template.

If structure validation succeeds, the second step is vocabulary conformance verification. Prerequisite to this step is to first process all dynamic vocabularies. That is, retrieving all certificate artefacts' vocabulary terms from the corresponding ontologies by executing the queries. Once dynamic vocabularies are instantiated, all certificate artefacts' vocabulary terms within the vocabulary part are checked against the corresponding artefacts' content in the certificate instance. All certificate artefacts defined to have an optional (non-mandatory) vocabulary will not be verified for conformance.

If vocabulary conformance succeeds, the third step is the semantic rules conformance verification. All semantic rules are processed, checked if satisfied by the certificate structure and content. Since the semantic rules of the profile may depend on the actual content (vocabulary) of a certificate artefacts in order to determine the semantic integrity of the certificate content, it is important to verify vocabulary conformance first, and then the semantic rules conformance.

We note that the vocabulary section of the profile does not enforce mandatory use of certificate artefacts. An optional certificate artefact can be forced to be mandatory either by the template part of the profile or by the semantic rules.

5 Proof-of-concept Realization

We will present a realization of the concept of security profile within the European project ASSERT4SOA. The project has developed a concept of a digital security certificate for services, called ASSERT. An ASSERT certificate is realised by an XML-based language which enables representation of a service security features in a structured, machine processable manner [6].

5.1 ASSERT Certificate

An ASSERT security certificate consists of the following main parts: *ASSERTCore* and *ASSERTTypeSpecific*. An ASSERTCore artefact defines the common aspects of a certificate, which are evaluation independent, such as certification process-specific information, target of certification, security property, security problem definition, service binding information, ASSERT issuer, etc. A *TargetOfCertification* artefact, part of the ASSERTCore, provides details about the service and its underlying architecture. Services can be of different types, such as SaaS, PaaS, or IaaS. It is important to define the TOC type in order to analyse if the certified properties are sufficient for a particular service type. A *SecurityProperty* artefact, part of the ASSERTCore, provides consumers with information on what property is certified and how the security property is realized by the service. Defines varying levels of abstraction such as an abstract security property, property context, assets being protected, etc.

An ASSERTTypeSpecific artefact defines the representation of details and results of a service evaluation process supporting the certified security property. Three evaluation categories are defined: Evaluation through testing, called

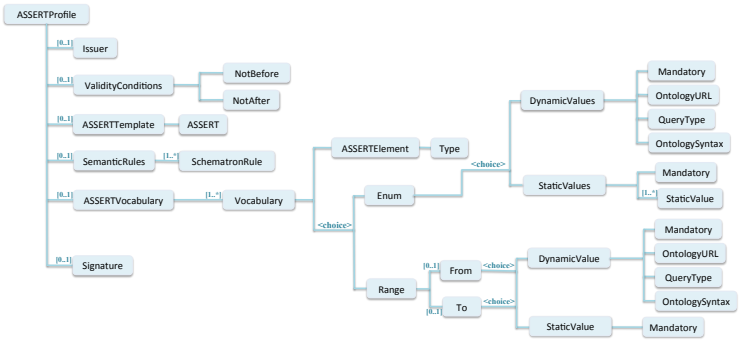


Fig. 3. ASSERT Profile Structure

ASSERT-E [20], Evaluation through formal analysis, called ASSERT-M [21], and Evaluation through ontology-based analysis, called ASSERT-O [22]. A *Property* artefact, part of the ASSERTTypeSpecific, defines type-specific property specification facilitating advanced reasoning such as comparison/ordering of security properties among services of same type evaluation.

5.2 ASSERT Profile

We have defined the structure of a certificate profile as an XML scheme, shown in Figure 3, and called the new structure an ASSERT Profile. For the sake of presentation, we show the profile structure in a rather informal way abstracting away some irrelevant XML schema details to better focus on the actual structure.

We will go through the main elements. The certificate template is called *ASSERTTemplate*. An ASSERTTemplate contains one element of type ASSERT certificate. Thus, an ASSERTTemplate contains an incomplete XML instance of an ASSERT certificate (according to the ASSERT XML schema). The semantic rules are implemented in Schematron [15]. Thus, semantic rules contain a set of *SchematronRule* elements. Schematron is an ISO standard rule-based validation language expressed in XML. Using Schematron, it is possible to make assertions about the presence or absence of patterns in XML trees.

The certificate vocabulary is called *ASSERTVocabulary*, which contains a set of *Vocabulary* elements each defining a specific vocabulary per an artefact (or set of artefacts) of ASSERT certificates. An *ASSERTElement*, part of the Vocabulary, identifies the ASSERT field(s) where specific vocabulary will be applied. Currently, we support the use of XPath [23] as a query language to identify nodes of ASSERT certificates where the vocabulary is to be applied. There is a choice of *Enumeration* or *Range* type of a Vocabulary. The former defines an explicit set of values, while the latter instead defines a range of values as *From* and *To* boundaries, such as integer range, double range (e.g., percentage), date range, etc. Each of the Enumeration and Range types are further defined as a choice of *DynamicValues* or *StaticValues* with an attribute field Mandatory indicating mandatory or optional use of the vocabulary data.

```
<ASSERTProfile>
 <ASSERTTemplate>
   <ASSERT>
    <ASSERTCore>
     <ASSERTIssuer>O=University of Malaga,OU=Computer Science Department,C=ES</ASSERTIssuer>
     <TargetOfCertification Type="http://assert4soa.eu/ontology/a4s-language#Platform-as-a-service"/>
    </ASSERTCore>
    <ASSERTTypeSpecific>
      <ASSERT-E/>
    </ASSERTTypeSpecific>
   </ASSERT>
 </ASSERTTemplate>
 <SemanticRules>
    <sch:schema queryBinding="xslt" xmlns:sch="http://purl.oclc.org/dsdl/schematron">
        <sch:pattern>
           <sch:rule context="ASSERT/ASSERTTypeSpecific/ASSERT-E/Property/PropertyName">
               <sch:assert test="//ASSERT/ASSERTCore/SecurityProperty[@PropertyAbstractCategory=current()]">
                  [Property E and property Core integrity check] SecurityProperty.PropertyAbstractCategory
                  has to match the same value of ASSERT.ASSERTTypeSpecific.ASSERT-E.Property.PropertyName
               </sch:assert>
           </sch:rule>
        </sch:pattern>
    </sch:schema>
 </SemanticRules>
 <ASSERTVocabulary>
  <Vocabulary>
   <ASSERTElement Type="XPATH">//ASSERT/ASSERTCore/SecurityProperty/@PropertyAbstractCategory</ASSERTElement>
   <Enum  Mandatory="true">
    <DynamicValues OntologyURI="http://assert4soa.eu/ontology/security.owl" OntologySyntax="RDF/XML" QueryType="SPARQL">
      PREFIX rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt;
      SELECT ?subClass WHERE { ?subClass  rdfs:subClassOf
                    &lt;http://assert4soa.eu/ontology/security#AbstractSecurityProperty&gt;. }
    </DynamicValues>
   </Enum>
  </Vocabulary>
  <Vocabulary>
   <ASSERTElement Type="XPATH">//ASSERT/ASSERTCore/SecurityProperty/@PropertyContext</ASSERTElement>
   <Enum Mandatory="false">
    <StaticValues>
     <StaticValue> http://assert4soa.eu/ontology/a4s-language#PersistentStorage</StaticValue>
     <StaticValue> http://assert4soa.eu/ontology/a4s-language#TemporalStorage</StaticValue>
     <StaticValue> http://assert4soa.eu/ontology/a4s-language#Transit</StaticValue>
     <StaticValue> http://assert4soa.eu/ontology/a4s-language#Usage</StaticValue>
    </StaticValues>
   </Enum>
  </Vocabulary>
 </ASSERTVocabulary>
</ASSERTProfile>
```

Fig. 4. ASSERT Profile Example

The *Dynamic Values* artefact defines an OntologyURI of how to retrieve the ontology; OntologySyntax specifies the ontology syntax; QueryType identifies the query language used to encode the query; and the actual query value. We currently support the use of SPARQL [24] as an RDF query language to retrieve information and manipulate data store in RDF format. The *Static Values* artefact defines a set of vocabulary terms as a simple list of values, or in case of a Range type a single vocabulary term.

5.3 ASSERT Profile Example

An example of an ASSERT profile structure shown in Figure 4 defines the following class of ASSERT certificates. The ASSERTTemplate defines all ASSERTs conformant to this profile must: *(i)* Be for software-as-a-service (SaaS) model services, i.e., all ASSERTs must have TargetOfCertification element with an attribute Type qualified as "http://assert4soa.eu/ontology/a4s-language#Platform-as-a-service"; *(ii)* Be issued by the University of Malaga as authority, i.e., all ASSERTs must have an ASSERTIssuer element with the defined value structure; *(iii)* Be produced by a test-based certification process, i.e. must contain

ASSERT-E type-specific structure, but without defining any particular content for ASSERT-E. This means that ASSERTs conformant to the profile can contain any specific ASSERT-E content.

The SemanticRules define one Schematron rule which forces the security property abstract category value as defined in the SecurityProperty element in the ASSERTCore of the ASSERT be the same value with that of the PropertyName of Property definition of ASSERT-E. The ASSERTVocabulary defines two vocabularies one for the PropertyAbstractCategory attribute of the SecurityProperty element and another for the PropertyContext attribute again of the SecurityProperty element. The first vocabulary defines dynamic values encoded as a SPARQL query marking those as mandatory. These terms are defined as subClassOf of the ontology class "http://assert4soa.eu/ontology/security#AbstractSecurityProperty". The second vocabulary defines static values for the artefact PropertyContext as terms within an ontology-specific definition.

6 Conclusions and Future Work

We have presented the concept of certificate profile to provide a mechanism to address processability and interoperability of service security certificates. We have presented the conceptual model and a concrete realization of the model within the context of the European project ASSERT4SOA. Validation of the use of security certificates and certificate profiles under specific criteria have been conducted and results reported in [25].

A direction of future work will focus on using certificate profiles to express certificate issuer competence (accreditation). This is an important aspect for end-users when they receive a security certificate of a service but wishes to know if the issuer of the security certificate does have the competence, expertise for the certified security claims. Our initial idea is to use attribute certificates (e.g., X.509) to encapsulate so-called "competence" profiles so that a certificate issuer can attach or provide his accreditation along the issued security certificates. The verification of such issuer competence will follow the same lines of profile conformance verification, i.e., if a security certificate issued by a given issuer conforms to the issuer's competence profile then the security certificate is verified to be issued by an accredited issuer.

Acknowledgments

This work was supported by the European funded projects ASSERT4SOA (grant no. 257361) and CUMULUS (grant no. 318580) of Framework Programme 7.

References

1. Gartner: Forecast overview: Public cloud services. report G00234817 (2012)
2. Common Criteria: Common criteria part 1: introduction and general model (2012) http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R4.pdf.

3. Sunyaev, A., Schneider, S.: Cloud services certification. Commun. ACM **56**(2) (February 2013) 33–36
4. Spanoudakis, G., Damiani, E., Maña, A.: Certifying services in cloud: The case for a hybrid, incremental and multi-layer approach. In: 14th IEEE International Symposium on High-Assurance Systems Engineering (HASE). (2012) 175–176
5. Anisetti, M., Ardagna, C.A., Guida, F., et al.: ASSERT4SOA: Toward security certification of service-oriented applications. In: OTM Workshops. (2010) 38–40 http://dx.doi.org/10.1007/978-3-642-16961-8_11.
6. Kaluvuri, S.P., Koshutanski, H., Cerbo, F.D., Maña, A.: Security assurance of services through digital security certificates. In: 20th IEEE International Conference on Web Services (ICWS-2013). (2013)
7. Mahbub, K., Pino, L., Foster, H., Spanoudakis, G., Maña, A., Pujol, G.: D2.1 - ASSERTs aware service query language and discovery engine. Technical report, ASSERT4SOA Project (2011) Available at http://assert4soa.eu/deliverable/D2.1.pdf.
8. Ramli, N.A.: Protection profile, a key concept in the common criteria. In: SANS Institute InfoSec Reading Room. (2003)
9. Benassi, P.: TRUSTe: an online privacy seal program. Commun. ACM **42**(2) (February 1999) 56–59
10. McAfee: Mcafee secure (2007) Online at http://www.mcafee.com/us/mcafeesecure/index.html.
11. X.509: The directory: Public-key and attribute certificate frameworks (2005) ITU-T Recommendation X.509:2005 | ISO/IEC 9594-8:2005.
12. SAML: SAML v2.0 (2005) Online at http://saml.xml.org/saml-specifications.
13. Andrieux et al.: Web services agreement specification (ws-agreement) (2011) OGF - Grid Resource Allocation Agreement Protocol WG, (v. gfd-r.192).
14. TAPAS Project: Trusted and QoS-Aware Provision of Application Services Available at http://tapas.sourceforge.net.
15. Schematron: ISO/IEC 19757-3:2006 Online at http://www.schematron.com.
16. Object Constraint Language: ISO/IEC 19507:2012 Online at http://www.omg.org/spec/OCL.
17. FIPS-197: Advanced encryption standard (2001) Online at http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
18. NIST-SP-800-38A: Recommendation for block cipher modes of operation (2001) Online at http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf.
19. Pino, L., Spanoudakis, G.: Constructing secure service compositions with patterns. In: 8th IEEE World Congress on Services (SERVICES 2012). (2012)
20. ASSERT4SOA Project Consortium: D4.1 - Design and description of evidence-based certificates artifacts for services. Technical report, ASSERT4SOA Project (2011) Available at http://www.assert4soa.eu/deliverable/D4.1.pdf.
21. Fuchs, A., Gürgens, S.: D5.1 Formal models and model composition. Technical report, ASSERT4SOA Project (2011) Available at http://www.assert4soa.eu/deliverable/D5.1.pdf.
22. D'Agostini, S., Giacomo, V.D., Pandolfo, C., Presenza, D.: An Ontology for runtime Verification of Security Certificates for SOA. In: Proc. of the 1st International Workshop on Security Ontologies and Taxonomies (SecOnt 2012). (2012)
23. XPath: XML path language W3C, Online at http://www.w3.org/TR/xpath/.
24. SPARQL: SPARQL query language for RDF (2008) W3C, Online at http://www.w3.org/TR/rdf-sparql-query/.
25. ASSERT4SOA Project Consortium: D7.3 - Validation of the ASSERT4SOA framework based on the study case. Technical report, ASSERT4SOA Project (2013)