Quality of Security Service for Web Services within SOA

Hany F. EL Yamany, Miriam A. M. Capretz and David S. Allison
Department of Electrical and Computer Engineering
Faculty of Engineering
University of Western Ontario
London, Ontario, N6A 5B9, Canada
{helyaman, mcapretz, dallison }@uwo.ca

Abstract

Service-Oriented Architecture (SOA) is a paradigm for creating and encapsulating business processes in the form of loose-coupling, autonomous and abstracted services. Managing the non-functional requirements of SOA such as security, is an overarching problem due to the wide variety of ways the service consumer can access the services offered by the service provider and the equally varied restrictions the service provider can set for gaining access by the service consumer. In this work, we propose a metadata for quality of security service for SOA. The proposed metadata provides different levels to describe the available variations of the Authentication, Authorization and Privacy features that are related to SOA security. A Web Service for Quality of Security Service (QoSS) is then constructed to encapsulate the suggested metadata in order to assist the service consumer and provider to achieve a QoSS agreement meeting both of their requirements. The QoSS agreement will perform as an enforced policy for managing the interactions between the service provider and consumer. The service of QoSS is located inside a complete framework for securing SOA.

1. Introduction

Service-Oriented Architecture (SOA) is a paradigm for creating and using business processes. The business processes are encapsulated inside loose-coupling, autonomous and abstracted components, called services [1]. These services are offered and published by a service provider who acquires the business processes. The services are designed to be easily published and discovered through a service registry such as the Universal Description, Discovery and Integration (UDDI) platform by registering its abstracted Web Service Description Language (WSDL) file within that platform. The service consumer can search the UDDI for the services he/she needs. Once the service consumer finds the required services, he/she sends a request to the service provider asking for access to those services. The service consumer and provider exchange request/response messages through the use of protocols such as Simple Object Access Protocol (SOAP). Figure 1 depicts the basic structure of SOA [2].

Figure 1. The main structure of SOA

One of the greatest challenges in designing SOA is organizing and managing its security requirements. This challenge stems from the large number of available security policies that Web services, and typically SOA, can use for protection. The variety of security policies available may confuse the service provider and consumer as to which security policy they should adopt in order to achieve both a straightforward and secure intercommunication. There is also always a conflict between the loose services access that the service consumer may ask for and the security policies that the service provider insists upon. The service provider needs these security policies in order to prevent any breaches or vulnerabilities that may result from the unconditional and unsafe access requested by the service consumer. This conflict requires a highly managed technique to be deployed as an intermediary between the service provider and consumer to organize policy concerns. Consequently the Quality of Security Service (QoSS) becomes an important requirement for SOA security.

Some of the known works has addressed the QoSS as just an authentication lemma such as the presented works by Irvine and Levin[3], and Papazoglou et al. [4], classifying it in several categories based on some defined metrics such as the strength of the authentication tokens and the encryption methods. This is a poor definition of QoSS for SOA security as it does not properly reflect all...
the main elements that constitute the SOA security structure, including authorization and privacy [7]. In fact, Ran [5] has suggested expanding the definition of QoSS to include the other terms of SOA security such as authorization and non-repudiation, but no specific details have been mentioned. However, one recent work presented by Mohamed et al. [6] has already added some authorization terms within the QoSS definition, demonstrating the variations of the access control techniques that might be applied inside an SOA environment. This proves the importance of expanding the QoSS to include the other SOA security aspects, such as privacy [7], in order to achieve a fine-grained definition of QoSS.

Encapsulating the QoSS definition inside a Web Service running for SOA is another essential requirement for managing SOA security. It is advantageous to the service provider to publish an abstract version of the QoS terms for SOA that he/she offers to the consumers. Announcing the QoS terms in this way provides two basic benefits. First, the consumer knows generally what main aspects of the SOA security requirements are needed to let him/her access a service located at the provider side. For example, he/she would know if the service provider is interested with the privacy terms, including collecting the consumer’s attributes and how long those attributes can be stored, in order to understand how his/her attributes will be protected. A second benefit is that since the consumer already knows the general SOA security requirements, he/she does not have to send enquiry messages, reducing the demand on the service provider. Some works, such as by Ran [5], have offered a solution to the QoS publishing problem by extending the UDDI to have a further special register for the QoS terms. This proposed solution makes the UDDI more complex and it is neither logical nor efficient to modify the UDDI for every special publication problem.

In this work, we suggest a metadata for QoSS that includes an expressive definition for three of the important SOA security aspects: Authentication, Authorization and Privacy [7]. The QoS metadata will be encapsulated inside a Web Service that could be easily published and discovered by the service consumer through accessing the traditional UDDI. The QoS service offers several hybrid levels of the QoS metadata to cope with the many security requirement combinations possible between the service provider and consumer. When the two parties agree on the proper requirements, a QoS agreement will be produced to save and guarantee the rights of both sides. Consequently, the QoS agreement will be considered as an enforced QoS model in order to control and secure the transactions between the service provider and consumer.

The remainder of this paper provides specific details about our proposed QoS Service and its encapsulated QoS model. Section 2 demonstrates a brief summary of the SOA security framework that embeds the service of QoS. Section 3 provides a full description of the proposed QoS metadata, and Section 4 depicts the suggested Services of QoS (SQoS). Section 5 illustrates the implementation of SQoS and Section 6 describes the current research on SOA QoS and QoS specifically. Finally, Section 7 contains a summary and presents future work.

2. SOA Security Framework

Designing a security framework that supports and maintains the QoS of SOA is a large challenge for the SOA QoS management. This is because we cannot describe a fine-grained QoS for SOA unless we have a good description and configuration for the main aspects of SOA security, including the Authentication, Authorization and Privacy. Thus, we have proposed in prior works [8,9] two complete Authentication and Authorization services for SOA in addition to a promising description for a new privacy metadata for SOA [10]. The suggested services constitute the backbone of the proposed QoS service as they provide the suitable values for the suggested elements inside the encapsulated QoS model. Figure 2 shows a portion of the proposed SOA security framework. A brief description of each service is introduced as follows:

The Authentication and Security Service (NSS) [8]: The NSS is divided into two basic parts: authentication and intelligent security. The authentication part is responsible for authenticating the service consumers that may use diverse types of authentication tokens, such as a username token, to identify themselves to the service provider. The intelligent security section’s primary function is to drop or accept the incoming request SOAP message through the use of a data mining core. The intelligent core produces an association rules model by connecting selected metrics such as the SOAP message parsing time and the request SOAP messages, with the possible web attacks. The suggested intelligent core uses the produced rules for predicting web attack that may occur.

The Authorization Service (AS) [9]: The AS is the second intelligent service on the proposed SOA security framework. Embedded within the AS is a 4-attribute vector consisting of the attributes User, Object, Environment and Condition. The interaction between the authorization roles and this vector framework manages the access control requirements for Web Services in an SOA environment.
The proposed authorization structure is mainly constructed based on the definition of Attribute Role-Based Access Control (ARBAC) which has been suggested to work mostly with Web Services [11]. This ARBAC approach is a hybrid of the Role-Based Access Control (RBAC) [12] and Attribute-Based Access Control (ABAC) [13] techniques. The AS also encapsulates an intelligent mining core. This intelligent core uses the centralized attributes-roles registry within the AS to semi-dynamically assign the access control roles to a new user or object when added to an SOA environment.

The Privacy Service (PS) [10]: A group of rules and principles such as the Collection Limitation Principle and the Purpose Specification Principle have been introduced towards building a fine-grained privacy metadata. The main principles are based on the fair information practices developed by the Organization for Economic Co-operation and Development (OECD) [14]. The suggested rules are working with the principles in order to reach a satisfied privacy plan that keeps the service provider and consumer confident about their data safety.

The Service of Quality of Security Service (SQoSS): The SQoSS is generally responsible for creating a QoSS agreement that will be established as a policy to organize the interaction between the service provider and consumer in terms of the features exposed in the authentication, authorization and privacy services. The SQoSS offers several levels of QoSS to grant the security requirements for the service provider and consumer together. The following sections describe in detail the SQoSS structure as well as the embedded operations.

Designing and encapsulating the logic of SOA security aspects in services have several benefits, which are:

1. The suggested SOA security services maintain the same characteristics of SOA. For example, each service is autonomous, controlling its logic and behavior so it is easily maintained and modified without impacting other services. All the services are loosely-coupled, where each service executes its functionalities independently, but maintain relations and exchanges messages with other security services thus guaranteeing a much more secure environment for the service provider. Finally, the security services are designed to be abstracted services by hiding their logic and structure from the outside world.

2. The suggested SOA security services are easily discoverable and reusable. This means that those services can be navigated among the trusted SOA enterprises individually or collectively in order to perform the task that is required of them.

3. The security services are capable of protecting and securing a huge number of services offered by the service provider [7].

3. The QoSS Metadata

The following subsections demonstrate the full structure of the QoSS metadata and the suggested levels of QoSS that bridge the gap between the security requirements of the service provider and consumer.

3.1 The QoSS Metadata Structure

In this research, we propose a novel QoSS metadata for SOA as we have added to the traditional authentication specifications, created more attributes describing the authorization and introduced a complete description for the privacy principles. The QoSS metadata will be encapsulated inside an agreement running as an enforced policy between the service provider and consumer. To the best of our knowledge, the privacy principles have never been discussed in terms of QoSS. Also, some attributes of the QoSS metadata have been
constructed based on the suggested hypotheses in our proposed SOA security framework. The metadata is fully described in Extensible Markup Language (XML) format which makes it easy to encapsulate in a service for secure SOA. Figure 3 depicts the high level structure of the proposed QoSS metadata for SOA. Table 1 provides a full description for each element involved in the suggested QoSS metadata for SOA.

<table>
<thead>
<tr>
<th>Table 1. QoSS Metadata Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Element Name</strong></td>
</tr>
<tr>
<td><strong>Authentication</strong></td>
</tr>
<tr>
<td>Certificate</td>
</tr>
<tr>
<td>Digital Signature Algorithm</td>
</tr>
<tr>
<td>Encryption Method</td>
</tr>
<tr>
<td>Web Attack Prediction</td>
</tr>
<tr>
<td><strong>Authorization</strong></td>
</tr>
<tr>
<td>Technique</td>
</tr>
<tr>
<td>Roles Types</td>
</tr>
<tr>
<td>Role Class</td>
</tr>
<tr>
<td>Location</td>
</tr>
<tr>
<td>Access Time</td>
</tr>
<tr>
<td>Condition</td>
</tr>
<tr>
<td>Roles Assign Process Type</td>
</tr>
<tr>
<td><strong>Privacy</strong></td>
</tr>
<tr>
<td>Collector</td>
</tr>
<tr>
<td>Data Category (What)</td>
</tr>
<tr>
<td>Purpose</td>
</tr>
<tr>
<td>Retention Time</td>
</tr>
</tbody>
</table>

```xml
<QoS>
  <Authentication>
    <Certificate/>
    <DigitalSignature/>
    <EncryptionMethod/>
    <WebAttackPrediction/>
  </Authentication>
  <Authorization>
    <Technique/>
    <RolesTypes/>
    <Object>
      <Class/>
      <Location/>
      <AccessTime/>
    </Object>
    <Condition/>
    <RolesAssignProcess/>
  </Authorization>
  <Privacy>
    <Collector/>
    <DataCategory/>
    <Purpose/>
    <RetentionTime/>
  </Privacy>
</QoS>
```

Figure 3. A High-level Structure of QoSS Metadata for SOA

While Table 1 clarifies most of the involved elements in the QoSS metadata, we believe that there are some expressions that need to be explained in greater detail. For example, in the authentication section, we add a new element to check the ability of an SOA security framework to predict and prevent possible web attacks. It is an important feature to be planned to ensure the consumer that the services he/she would like to access are available and his/her data is in a secure server. Many new elements have also been configured in the authorization section. Since our proposed AS is dealing with a 4-attribute vector structure to handle the access control for SOA, it would be better to have many available categories for both the service provider and consumer. This increases the flexibility of the SOA security options and manages the dynamic nature it possesses. The most important element in the QoSS in terms of authorization is the categorization of the object class or grade into a particular level according to its importance to the service provider. This classification will determine the degree of the privileges that the consumer will be granted. Finally, the privacy section is built on the suggested principles in [14]. The proposed QoSS privacy is a very crucial section since the service provider and consumer often disagree
over the amount of access to each other’s data. This problem can be solved by the inclusion of privacy in the QoSS and the creation of an agreement of privacy rights.

It is of importance to note that the suggested QoSS metadata is flexible. The elements are editable and their values can be changed. As an example, the service provider might not like to show the structure of the security services he/she runs, such as predicting web attacks or the management of authorization roles. In this case, he/she can eliminate the elements related to that information. In this paper we also classify the technique element in the authorization section, as shown in Table 1. This classification is based on who can define the required authorization techniques that the provider should consider to manage the access rights. The service provider can change the value of the technique element and define another authorization technique. For example, RBAC [12] which is introduced as a value of the technique element in Table 1 may be replaced by the proposed access control technique in the work suggested by Damiani et al. [15].

3.2 The QoSS Metadata Levels

We have defined 4 different levels for each section of the QoSS metadata: High, Moderate, Low and Guest.

3.2.1 The QoSS Authentication Levels

Figure 4a. The QoSS Authentication Levels

3.2.2 The QoSS Authorization Levels

Figure 4b. The QoSS Authorization Levels

3.2.3 The QoSS Privacy Levels

Figure 4c. The QoSS Privacy Levels

Figure 4. The QoSS Metadata Levels

Figure 4 shows that each element in the QoSS metadata has a different value according to the suggested four levels. Also, as previously explained, the suggested QoSS metadata is much more flexible and editable. As an example to demonstrate this flexibility, the service provider and consumer may prefer another definition for the Data Category element inside the privacy section. It may be classified into several categories such as personal, private and public with the regard to the service consumer attributes. The personal category can include the attributes belonging to him/her as a person such as email, address and date of birth. Meanwhile, the private category contains information related to his/her financial data such as social insurance number (SIN), credit card number and bank account number. Finally, the public data contains simple data that can be revealed publically with no reservations, such as first name, last name and the consumer city. Determining which category is more critical than the others depends on the perspective of the service provider or consumer.

The service consumer is free to select any level from the available four levels that are related to the three SOA security aspects: Authentication, Authorization and Privacy. However, the consumer selection should not conflict with the service provider security policy. This may require that both sides go through a long and complicated negotiation process to establish an agreement satisfying the security requirements of both sides. The agreement will be structured in XML format and saved at the location of the service provider. The consumer can also review this agreement and update it where possible as needed.

An example representing the conflict that may occur between the service provider and consumer is demonstrated when the service consumer selects a low authentication level and a high authorization level in order to be able to access the objects with a Class A degree. The security policy created by the service provider may refuse the consumer policy because the objects with Class A require the consumer to be authenticated with the available certificates in the high or moderate authentication levels. In another example, the consumer may choose a high or moderate authorization level while at the same time selecting a high or moderate privacy level. The contradiction is that the requested authorization levels cannot work with the requested privacy levels. The high or moderate authorization levels mean that the consumer authorized the service provider to collect the attributes he/she needs as the defined access control technique in those levels are Attribute-Role Based Access Control (ARBA) and Attribute Based Access Control (ABAC). On the other hand, the high and moderate privacy levels mean that the consumer is the side who determines the attributes that the service provider should collect which creates a contradiction to what was previously stated by the selected authorization levels.

4. The Service of Quality of Security Service (SQoSS)

The suggested QoSS metadata needs to be placed inside an SOA application or component in order to gain
the benefits described in Section 2. Since our proposed SOA security framework works at the service layer, the QoS metadata will be encapsulated inside an autonomous service to manage the security requirements between the service provider and consumer. The main functionality beyond building the Service of Quality Security Service (SQoSS) is producing a QoS agreement establishing the security requirements that the service provider and consumer should follow. The security services working inside the proposed SOA security framework such as the Authentication and Security, and Authorization Services, will organize their activities in authenticating and authorizing the consumers based on that QoS agreement. Figure 5 demonstrates a portion of the WSDL file of the SQoSS.

![Figure 5. A portion of the SQoSS WSDL file.](image)

As shown in Figure 5, the SQoSS has four main operations to handle the QoS metadata. The following is a description of each of those operations:

**ShowQoSDetails**: The SQoSS is being accessed as an object. Therefore the service consumer should be authenticated and authorized first to call this operation in order to be able to load the details of the SQoSS.

**ShowQoSConfiguration**: It is responsible for illustrating the general configuration of the abstract QoS metadata that the service provider can offer for its consumer.

**ReviewAgreement**: The negotiation process between the service provider and consumer about the QoS metadata is done through this operation. Once the service provider and consumer establish an agreement, a QoS agreement is produced and saved in the database belonging to the SQoSS.

**RetrieveAgreement**: The service consumer can review his/her QoS agreement through this operation. The operation works automatically as the service consumer is permitted the access to the SQoSS.

The service consumer accesses the SQoSS by sending SOAP messages over Hypertext Transfer Protocol (HTTP). An example of a request SOAP message is shown in Figure 6.

```xml
<soap:Header>
  <GetUserCredentials
      xmlns="http://www.SOASecurity.com/">
    <Username>string</Username>
    <City>string</City>
  </GetUserCredentials>
</soap:Header>

<soap:Envelope xmlns="http://www.SOASecurity.com/">
  <soap:Body>
    <ShowQoSDetails
        xmlns="http://www.SOASecurity.com/">
      <tns:ShowQoSConfigurationSoapIn/>
      <tns:ShowQoSConfigurationSoapOut/>
    </tns:ShowQoSConfiguration>
  </soap:Body>
</soap:Envelope>
```

**Figure 6. A request SOAP message.**

The shown request SOAP message is the initial message that the service consumer sends to access the SQoSS. The service consumer would be authenticated first by a Username/Password token and authorized second by his/her city, for example. After the QoS agreement is established, the service consumer must be authenticated and authorized according to the listed QoS agreement items.

5. The QoS Metadata and Service Implementation

All the proposed security services, including the SQoSS and the SOA security framework shown in Figure 2, have been implemented using .NET technologies. All the security databases have been constructed using SQL Server with the possibility of transforming the stored data in those relational stores into XML and vice versa according to the requirements of the situation. All the security services are suggested to be located inside the Enterprise Service Bus (ESB) which manages the routing process between the service providers and consumers.

The process of establishing a QoS agreement is described as follows:

1. The service consumer first searches the UDDI until he/she finds the QoS configuration that may comply with the security requirements he/she needs.
2. The consumer registers him/her self at the service provider by following the QoS configuration that he/she found by searching the UDDI. The registration process is completed by the consumer obtaining a Username/Password token in addition to supplying the provider with additional attributes, such as his/her location. These additional attributes help the provider authorize the consumer when he/she requests access to the SQoSS in the future.
3. The service provider automatically adds any authenticated consumer to the authorization role that gives the privilege to access the SQoSS.

4. To obtain access to the SQoSS, the consumer should provide the correct Username/Password token and the attributes he/she has previously provided.

5. Once the consumer has been granted access, he/she can evaluate and review the general QoSS configuration. This general configuration is derived from the QoSS metadata. The consumer can then build the QoSS levels he/she requires. Figure 7 shows the interface of the SQoSS, including the authentication, authorization and privacy levels where the left side shows the general QoSS metadata whereas the right side shows the QoSS contact with the selected levels.

6. The consumer submits his/her QoSS levels he/she has selected. The SQoSS reviews the submitted levels for conflicts and returns feedback if any are found. If no conflicts are found, a QoSS agreement is established and saved. A notification message will then be sent to the consumer. The notification is also shown in the uppermost left of the screen in Figure 7.

7. A negotiation process may be run between the service consumer and provider until both parties arrive at an agreement. The service consumer can review the QoSS agreement items anytime he/she would like. The QoSS agreement will be run as an enforced policy to manage the interactions between the two parties. Details of the agreement negotiation are beyond the scope of this paper.

Figure 7. A snapshot of the SQoSS Interface.

6. Related Work

Enormous works for QoSS for SOA have been developed by Boeing Phantom Works with their ideas and results published in several papers [6, 16, 17]. The authors Mohamed et al [6], Wang et al [16] and Wang [17] have proposed QoSS solutions starting from a policy-based approach and moving into encapsulating the QoSS policies inside a Quality Service. They have also defined the authorization aspect in their recent work [6]. However, they have only mentioned one authorization term, which was the possible types of the deployed access control technique, such as RBAC. Moreover, they did not take into their considerations privacy terms, as we have proposed in this research.

Larrucea and Alonso [18] have presented a modeling framework based on the Eclipse platform for modeling and designing security aspects in SOA. The suggested security model is based mainly on the authentication elements such as integrity and confidentiality, in addition to the availability in terms of authorization. This proposed approach is applied on the message level and as a result cannot be reused or published as we have suggested in this work. Again, neither detailed authorization nor privacy has been discussed.

Fung et al. [19] have studied the QoS management in Web Services composition. They have proposed a SOAP tracking model for supporting QoS end-to-end management in the context of Web Services Business Process Execution Language (WSBPEL) and Service-Level Agreement (SLA).

Tian et al. [20] have proposed approaches for monitoring the QoS for Web Services. It offers the WS-QoS monitor to help users check the compliance of the service offer and to identify inappropriate definitions of QoS requirements such as reliability and availability. However, the Security in their works [19, 20] was just a small part of their discussions on QoS for SOA. This limited view of security produces a lack of robust and coherent approaches for solving the real security concerns. Moreover, the privacy terms have not been discussed as well.

Finally, the QoSS term has been studied earlier in the article presented by Irvine and Levin [3]. The authors have examined the QoSS in distributed systems in general. They presented a discussion and examples of user-specified security variables such as the strength of a cryptographic algorithm and the length of a cryptographic key. Later, the QoSS definition was extended to include other items such as the service provider and consumer negotiations such as in the works proposed by Xia and Hu[21] and Chen et al [22]. However, neither of those works has discussed the QoSS terms as a policy controlling the interactions between the service provider and consumer in an SOA environment.

7. Summary and Future Work

In this work, we have proposed a Quality of Security Service (QoSS) metadata for SOA including several
elements representing the three main security SOA aspects: Authentication, Authorization and Privacy. This QoSS metadata is flexible and editable and is divided into four basic levels: High, Moderate, Low and Guest. These levels allow the varied security requirements of the service provider and consumer to be satisfied. The QoSS metadata has also been encapsulated inside an abstracted and reusable service. This helps the service provider publish its QoSS configuration as well as help the consumer find a suitable QoSS. The Service of QoSS also assists in establishing an agreed upon QoSS agreement between the service provider and consumer that will act as an enforced policy to manage the interactions between the two sides.

Our future work includes studying other aspects of SOA security such as auditing, and consequently extending the proposed QoSS metadata by adding the auditing features including the non-repudiation items such as the consumers requests proofs. We also need to examine the proposed QoSS metadata to check if it is possible to allow the consumer to select hybrid security elements from two different QoSS levels without creating conflicts with the service provider security policies.

8. References