Security Aspects in Semantic Web Services Filtering

Witold Abramowicz [1]), Andreas Ekelhart [2]), Stefan Fenz [2]),
Monika Kaczmarek [1]), A Min Tjoa [3]),
Edgar R. Weippl [3]), Dominik Zyskowski [1])

Abstract
Security and trust aspects, perceived as difficult to quantify, have been neglected in various service interactions. However, factors related to security and trust are in fact crucial in the overall value of service quality. A security ontology that enables a quantification of risks related to the usage of Semantic Web services in enterprise information systems was created to meet users' requirements and enhance Semantic Web services with machine processable security information. This article presents how this security ontology can be integrated into the Web service description and how it enhances the process of Web services filtering.

1. Introduction

In conjunction with the increased usage of Web services (WS) [5], the demand for an appropriate description of services gains in importance. The syntactic description of Web services (provided by WSDL [2] and UDDIs [3]) has reached a limit that can be expanded only by usage of semantics. This applies to the level of automation of interactions between services that cannot be increased without changing the way that services are described. That is why the utilization of Web services together with Semantic Web technology, so in consequence Semantic Web Services (SWS), has been proposed. A few languages have been developed to semantically represent Web services, e.g. OWL-S [6], WSMO [20], SAWSDL [7]. The common agreement is that a Web service should be represented by its surrogate describing semantically its functional, non-functional, and behavioral characteristics. The functional features focus on what a WS does, the non-functional ones on how it does it, and behavioral ones inform which parties are involved in the process of providing services.

There exists a common agreement on how the functional properties of a Web service should look like (usually in the form of IOPE i.e. inputs, outputs, preconditions and effects), whereas there is still an on-going discussion on the scope and the formalism that should be utilized to express the non-functional side of a service that may be further on used within various SWS-based interactions. It is beyond any doubt that businesses need a rich description of non-functional properties (NFP) of services that they are going to use. It would not be wise or acceptable to include as a part of a

[1] Department of Information Systems, Poznań University of Economics, Al.Niepodległości 10, 60-967 Poznań, Poland {w.abramowicz, m.kaczmarek, d.zyskowski}@kie.ae.poznan.pl
[2] Secure Business Austria, A-1040 Vienna, Austria {aekelhart, sfenz}@securityresearch.at
[3] Institute for Software Technology and Interactive Systems, Vienna University of Technology, A-1040 Vienna, Austria {tjoa, weippl}@ifs.tuwien.ac.at

business process, a service that is underspecified in terms of its non-functional properties. If a company does not know the values of the execution duration of a service, for example, how can be guaranteed that a customer will get the required service within five hours? The performance of a WS is not the only relevant non-functional attribute whose value should be known in advance. Security related issues are also included here since nobody would use insecure services in mission-critical applications. Moreover, NFP may be treated as distinctive criteria for the success of businesses offering their services using Web services technology. They make it possible to differentiate between Web services offering the same (or quite similar) functions since in most cases service substitutes differ in terms of the value of specific non-functional properties.

In general, non-functional parameters can be represented as qualitative and quantitative measures of a Web service. Qualitative measures include security and transactions, whereas quantitative measures include such attributes as cost and time [8]. The complex interactions between SWS require rich and informative descriptions of NFP [1][10]. In order to allow reasoning on values of NFP, they need also to be presented in the ontological form. Although some attempts have been undertaken to define NFP and/or QoS for WS and SWS (e.g. [11][12][14][15]) in fact, not enough work has been done up till now to represent the security related aspects. These approaches fail to consider the security of WS as an aspect that should be taken into account during discovery, filtering and selection [16][18]. Thus, it is not possible to simulate how the security affects an overall quality of services. Since NPF are not restricted solely to type-value pairs, the task remains of developing a descriptive and widely applicable solution for the representation of heterogeneous parameters in a machine understandable manner, while supporting reasoning and prediction.

The appropriate way to present security issues necessary for SWS and their usage arises as a new challenge. How should information on security-related aspects be presented and further on processed in order to be utilized within the SWS-based interactions? In this article we argue that the creation of a security ontology will allow for quantification of risks related to the usage of SWS in enterprise information systems. We present ways of exploiting a security ontology in the SWS filtering process in order to provide results that are tailored to users' needs. The main aim is to identify from the stream of incoming WS only those services that are relevant to a user profile, which describes bot the functional properties that interest the user and the NFP that should be assured. Security parameters crucial for all IT systems should also be taken into account during the filtering process, however, in order to accomplish that, they need to be properly represented.

The article is structured as follows. First, we introduce the security ontology and its possible application for the needs of enterprises using SWS. We then describe a SWS filtering algorithm which supports security related properties. Finally, we give conclusions and details of future work.

2. Integrating security aspects in Semantic Web services descriptions

The Web service community focuses on functional and behavioral aspects of WS. However, non-functional aspects are also being given more attention due to their importance in service discovery,

selection, and substitution. The semantic description of Web services' security capabilities, in particular, is necessary for meeting the defined security requirements of critical applications, such as services for financial transactions. OWL-S [6] and WSMO ([4], [20], [21]) are initiatives for describing Semantic Web services that enable automated discovery, composition, and invocation. This section examines how Semantic Web service initiatives (OWL-S and WSMO) integrate security parameters in their Web service descriptions.

2.1. Existing approaches

The following subsections examine the current coverage of security aspects in OWL-S and WSMO.

2.1.1. OWL-S

OWL-S considers the following NFP: service name, text description, and quality rating. Parameters from the service profile can be used to describe further non-functional properties. In 2003, Denker et al. [22] proposed security annotations for DAML-S [23] (predecessor of OWL-S) descriptions. Security-related ontologies were introduced to describe the security requirements and capabilities of Web service providers and agents. Reasoning engines can then decide if the Web service and the agent have comparable security parameters. In 2004, Kagal et al. [24] incorporated privacy and authentication policies into OWL-S descriptions and requester profiles and implemented algorithms for checking policy compliance. Kim et al. [25] refine the aforementioned approaches by proposing concrete security ontologies for security algorithms, credentials, etc.

2.1.2. WSMO

To model non-functional aspects in WSMO, a working draft has been proposed [21] that introduced 17 ontologies describing non-functional domains, such as network-related QoS, performance, reliability, scalability, and security. These ontologies can be imported and concepts can be instantiated and used in the service descriptions. The proposed models begin with an initial set of the most widely used non-functional properties, which can be adopted and extended for various needs. To describe security as a NFP, WSMO community provides a security ontology [26] to define the identification requirements and confidentiality mechanisms. A shortcoming of this security ontology is the general and non-restricted description of its elements. More, it lacks descriptions of authentication and other security requirements, such as integrity and non-repudiation. The missing specifications make it difficult to use the ontology.. The following clipping (see Listing 1) shows the *EncryptionTechnique* concept of the WSMO security ontology:

Each instance of the *EncryptionTechnique* concept is described by the attributes *hasName*, *hasKeyLength*, and *hasDetails*. Especially the name and the key length of an encryption technique are defined by general data types, such as string or non-negative integers. These types do not restrict the possible values (within the data type) and make it nearly impossible to compare different instances. For example, although the instances *AES*, *AdvancedEncryptionStandard*, and *AES256* describe the same encryption algorithm they can't be matched due to their different identifiers.

```
concept EncryptionTechnique
        nonFunctionalProperties
         dc#description hasValue ”ConfidentialityAgreement concept definition”
        endNonFunctionalProperties
         hasName ofType (1 1) string
         hasKeyLength ofType (1 1) loc#nonNegativeInteger
         hasDetails ofType (1 1) qua#Standard
```

Listing 1: WSMO security ontology - EncryptionTechnique concept [21]

2.2. WSMO security ontology extensions

As we have seen in 2.1, non-functional security properties are already supported by WSMO, offering a generic security ontology for identification and confidentiality. Since this is a valuable approach, we extend this ontology to facilitate automatic evaluation of SWS. To include security requirements in the discovery, filtering, and selection process of Semantic Web services there must be a shared terminology that provides a consistent definition of concepts used. Therefore, we propose the extension of security ontologies by [22] and [25] to integrate them into the WSMO security ontology. This enables Web services to describe their security capabilities and requesters to define minimum security requirements for their specific needs.

The Security Algorithms Ontology defined in [25] provides a solid structure for modeling security algorithms and classifies instances. *Algorithm* is the root concept followed by four subconcepts for different algorithm types (*KeyExchangeAlgorithm*, *EncryptionAlgorithm*, *ChecksumAlgorithm*, and *SignatureAlgorithm*). The *Algorithm* concept has one boolean property, namely *IsNISTStandard*. For *SymmetricAlgorithm*, a subconcept of *EncryptionAlgorithm*, three properties exist: *hasNSALevel*, *modeofOperation*, and *keyLength*. Various algorithm instances have been defined for the existing concepts, including Diffie Hellman, KEA, DES, AES, and SHA-1. As an example of encryption algorithms, we show which points have to be changed to meet our requirements for a semantic security encryption ontology:

1. More semantic information about the algorithms must be included for reasoning with the data and for identifying encryption standards unambiguously. The *SymmetricAlgorithm* concept is the only one that includes properties for key length, mode of operation, and the NSA security level. Still missing, however, are properties for performance description, block size, number of rounds, and information on the developer to describe each algorithm in the most granular form.
2. In [25] algorithms are modeled at a different level of granularity: Although a key length of 64 has been assigned to the DES instance, there is no key information given for AES. We require that such information be specified in the mandatory properties.
3. [25] does not define semantics for all properties (e.g., the unit of the key length is not defined).
4. The paper was published in 2005, which means that the list of algorithms requires updating.

```
ontology _"http://www.securityresearch.at/ontologies/nfp/securityNFPOntology"
  …
  importsOntology
  { _"http://www.wsmo.org/ontologies/nfp/qualityNFPOntology",
   _"http://www.wsmo.org/ontologies/nfp/measuresNFPOntology",
   _"http://www.wsmo.org/ontologies/computerEquipmentOntology"}

concept EncryptionAlgorithm subConceptOf Algorithm
  keyLength ofType (1 1) _"http://www.wsmo.org/ontologies/nfp/measuresNFPOntology#Information"
  encryptionPerformance ofType Performance
  keySetupPerformance ofType Performance

concept Algorithm
  hasDetails ofType (1 1) _"http://www.wsmo.org/ontologies/nfp/qualityNFPOntology#Standard"

concept SymmetricEncryptionAlgorithm subConceptOf EncryptionAlgorithm
  rounds ofType (1 1) _integer
  blockLength ofType (1 1) _"http://www.wsmo.org/ontologies/nfp/measuresNFPOntology#Information"

concept AES subConceptOf SymmetricEncryptionAlgorithm

concept Performance
  cycles ofType (1 1) _"http://www.wsmo.org/ontologies/nfp/measuresNFPOntology#Cycle"
  onSystem ofType (1 1) _"http://www.wsmo.org/ontologies/computerEquipmentOntology#System"
          …
```

Listing 2: WSMO security ontology extension (excerpt)

Listing 2 shows our WSMO security ontology extension. The instance *AES128* is modeled to show how concrete instances are integrated into this framework. For the unambiguous identification of encryption standards we extend the existing WSMO framework by the properties: key length, encryption performance, key setup performance, rounds, and block length; and for detailed information about the algorithm publisher we adopted the *qualityNFPOntology#Standard* WSMO concept. For defining the units of the newly introduced properties, we include in the WSMO measures ontology, the concepts *ClockCycle* and *Bit* (see Listing 3). Each *Performance* instance connects a specific system with its encryption performance, which is given in clock cycles. There is no constraint on the cardinalities for *encryptionPerformance* and *keySetupPerformance*, indicating that it is possible to omit performance information or assign multiple performance instances due to benchmarks on different systems. The properties of key length and block length refer to concrete WSMO measure instances defining the quantity and the unit of the values. Along with encryption algorithms we also modeled hashing and checksum algorithms (Figure 1 depicts the modeled encryption algorithms, concrete hashing and checksum algorithms are omitted to enhance readability) taken from the Security Ontology [27] [28].

By describing the security attributes of a SWS service in the most granular form, requesters are able to adjust their parameters for the discovery phase to a very detailed level. But not every requester is familiar with terms such as AES, Blowfish, and other security technologies and the average user

does not usually know the significance of the terms key length and block size. The strength of an encryption algorithm depends on the algorithm and the key length. Elliptic Curve encryption, for instance, provides more security than RSA with the same key length. Therefore, we have to introduce profiles for different security requirements.

```
instance KeyLength_AES128 memberOf Information
ofUnits hasValue Bit
numUnits hasValue 128
instance EncryptionCycle_AES128_PentiumPro memberOf Cycle
numUnits hasValue 440
ofUnits hasValue ClockCycle
```

Listing 3: WSMO measures ontology extension (excerpt)

2.3. Security profiles

It is possible to model security aspects in SWS with the aforementioned concepts. Prior to service discovery or filtering, the client must decide on its security requirements. To take the burden away from the client and facilitate automatic or semi-automatic matchmaking we suggest profiles based on NIST's recommendations [29]. When a Web service invokes another service it usually has specific security requirements that must be fulfilled. However, in most cases it does not matter which mechanism is used to guarantee the requirement. For instance, if encryption is to be used, a typical requirement would be "symmetric algorithm to guarantee confidentiality against attacks by powerful adversaries until at least 2015." According to [29], the "smallest" possible algorithm is AES-128. While this example only requires the encryption algorithm name and the corresponding key length, a more complex profile could also take into consideration performance metrics of an algorithm. Section 3 describes the usage of these parameters to enhance Web service filtering in terms of security aspects.

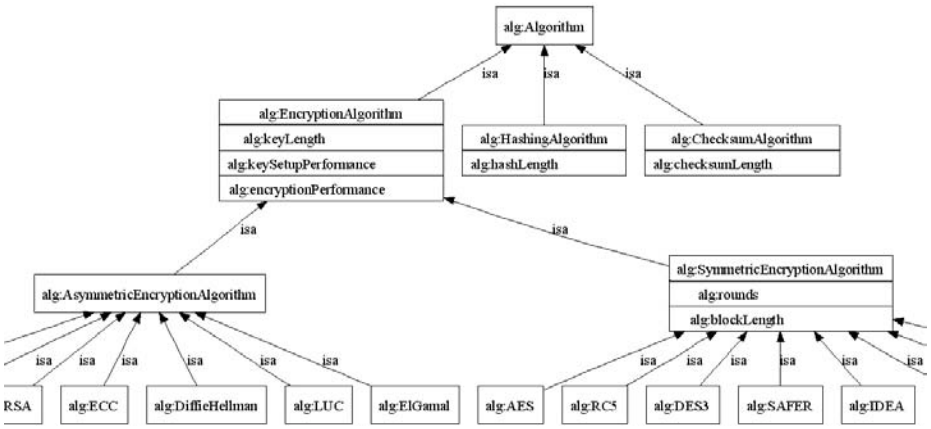

Figure 1: Security algorithms ontology (clipping)

Our non-functional property descriptions separate static knowledge on security and rankings allowing clients to use best practice profiles (as previously described) for their needs or to specify their own profiles if more granular control is required. For example, a company could demand or prohibit certain algorithms or key lengths to implement their own security requirements.

3. Enhancement of SWS filtering with security aspects

Information filtering can be defined as the specification of objects from a given stream in terms of relevance to a profile. In the IF field, a profile represents regular information interests (e.g., remaining current on a topic) that may change over time. This representation is used in filtering systems in order to provide users with the most relevant results. Therefore, the process of SWS filtering should be perceived as applying a set of semantic queries to a stream of single, semantic representations of services. The aim is to provide users with information about Web services by means of processing the content of service descriptions. Representation of SWS is possible using the enhanced OWL-S profile or the extended WSMO ontology described in the previous chapter. Users' profiles include, in addition to a security profile, a description of the requested service functionality and a description of preferences regarding the values of non-functional properties.

The algorithm for the filtering of Web services, as described in [19], consists of two stages: ontology based filtering directed at the detection of service substitutes (encompassing the matchmaking of Semantic Web services profiles) and constraint-based filtering, directed at the comparison of substitutes based on user preferences.

Since security related issues are among the NFP, in order to include the security ontology and a security profile as described in the previous section, the constraint-based filtering stage must be enhanced to also perform the analysis of security parameters and take into account a user's security profile. At this stage, the ontology describing relations between different parameters must also be taken into account.

No modifications to the first stage are necessary, thus we refer the reader to [19] for details and now focus on the second stage. Its objective is to identify the best service (according to user preferences) from the set of relevant services. Here, Multiple Criteria Analysis (MCA) is employed, allowing the comparison of a set of services based on their characteristics, such as price, response time, accessibility, and also the required level of security. It is necessary to assign each characteristic with a weight reflecting the user's preferences.

In order to employ MCA, the non-functional properties (along with the security related ones) must be divided into two groups, namely qualitative variables (e.g., payment method) and quantitative variables (e.g., response time). Security is, in fact, a qualitative parameter, but since it also needs to be compared, it must be quantified. This is necessary, for example, when we have a set of three service substitutes differing only in the values of non-functional properties, namely the encryption algorithms that are applied. The question arises which service can be established as the best and meeting the needs of a user who has defined security and high performance in the profile. At this stage the encryption algorithms must be compared taking into account their quantitative metrics as included in the security ontology.

Deciding on the applicable encryption algorithm is dependent on two main factors: security and performance. Within the classes of encryption algorithms (such as Asymmetric, Symmetric, Discrete Logarithm, or Elliptic Curve), key length is a characteristic number for security. It is possible to measure performance in the clock cycles for key setup and encryption. If security is the main concern, key length will dominate the decision; on the other hand, if performance is more important, e.g. restrictions on mobile devices, the clock cycles take top importance. As the performance can vary depending on the system, benchmarks of different system classes are required. For example, [30] provides detailed information on symmetric encryption algorithm performance. An issuer of the algorithm can present a further filtering criterion. All these properties can be found in the security ontology. A more general selection criterion could be (depending on the user profile) to search for an algorithm that is secure for a given period of time (compare Section 2.3). The Cryptographic Key Length Recommendation [9], for example, is a valuable source for such information, listing recommendations by ECRYPT, NIST, DCSSI, and NSA.

So, let us assume that each of the abovementioned three services employs a different encryption algorithm, namely: RSA, ECC, and XTR. To decide on a service, it is necessary to compare the encryption algorithms. As stated above, there are possible differences related to two factors: security level and performance. Therefore, we have listed competing encryption algorithms in Table 2, including their key length and performance characteristics on a Pentium Celeron 450 MHz [13]. Based on these tests we converted the values to the units used in our security ontology as shown in the same table. The conversion was done for a given processor. This results in the possibility of automatically evaluating the algorithms to determine which one is most appropriate for a specific purpose stated in the user preferences. Please note: our choice of parameters is arbitrary as we are concentrating on issues related to performance and key generation.

Table 2: Crypto algorithms comparison

Parameter/algorithm	RSA	ECC	XTR
Key length (bits)	1024	161	Comparable with ECC
Key generation time (processor clocks)	1 261 261 261	40 540 540,5	Less than ECC
Encryption time (processor clocks)	11 261 261,3	3 243 243 243	Comparable with ECC

Of course, there are other algorithms that can be used. For example, Rijndael (standard for AES encryption), which with a key length of 128 bits provides a level of security comparable to RSA with a key in the range of 2790–3390 bits [17]). To provide trustworthy estimations of the parameters considered in Table 2 all tests must be run on the same hardware configuration.

When the crypto algorithms considered in the filtering process are represented in a semantic way by using our ontology, the reasoning presented above can be automated. The ontology allows the user to specify relations between different algorithms. It also enables the user to define parameters of these algorithms that will later be used in the constraint-based part of the filtering process. The

clipping shown in Figure 1 is also a taxonomy of algorithms that distinguishes between asymmetric and symmetric encryption algorithms. Assuming that a user puts a hard constraint on the type of algorithm desired, it is possible to infer easily and automatically whether the algorithm that is used in a certain Web service truly meets the criterion of type (symmetric or asymmetric).

As mentioned above, security is generally a qualitative parameter. However, because we need to compare different levels of security (e.g., provided by different algorithms, such as in the use case presented above), we consider its quantitative characteristic (e.g., key length and performance) expressed in the security ontology rather than an encryption algorithm itself (also a qualitative parameter). It is necessary to remember that for some of the quantitative characteristics, the value desired is as low as possible (e.g., latency time), for others, on the contrary, as high as possible (e.g., reliability). Keeping this in mind, three kinds of quantitative variables can be distinguished: the-larger-the-better (LTB), the-smaller-the-better (STB) and nominal-the-best (NTB). The table below shows the character of all the NFP considered in the exemplary filtering process.

Table 3: Characters of non-functional properties in the filtering process

Character	Properties
LTB	Key length, Reliability, Availability
STB	Key generation time, Encryption time, Price, Execution time, Maximal response time, Average response time

The method used to compare phenomena is the computation of the synthetic indicator. For more information regarding the computation of synthetic indicator please refer to [19]. To compute the value of the synthetic indicator, the user preferences expressed in terms of weighted vectors are required. Each user can have different preferences described in his security profile.

For the purpose of our experiment, we have taken into account the results of a survey and have defined the default vector of such weights. The survey, which was conducted in the framework of research on NFP among Polish and German SMEs revealed that security of Web services is of utmost importance. The participants came from a variety of industry domains including: IT, trade, public administration, healthcare, and services. Over 78 percent of the companies polled had their own IT department. More than 42 percent of those surveyed felt that security is of very high importance and another 26 percent felt that it is of high importance. The detailed results of the survey are beyond the scope of this article. Nevertheless, these results allow us to define the default vectors for a comparison of Web services. In our use case, where security is important, the default vector might look as follows: security – 0.3; price – 0.1; execution time – 0.15; maximum response time – 0.05; average response time – 0.05; reliability - 0.2; availability – 0.15. Because security is, in fact, characterized by three parameters in our example, the weight vector to be applied in our case study looks as follows: key length – 0.15; key generation time – 0.075; encryption time – 0.075; price – 0.1; execution time – 0.15; maximum response time – 0.05; average response time – 0.05; reliability – 0.2; availability – 0.15.

The division of security weight does not simply divide the overall weight into a number of parameters. We consider the key length more important as it is the key factor determining the security level. In addition, please note that depending on the inside logic of the Web service, the encryption time and key generation time may also have an impact on the overall value of the execution time parameter. Since the Web service is considered a black box we are currently incapable of assessing it. Once weights are defined, it is easy to compute the synthetic indicator and compare services so that results can be passed on to the user. Please note that using the security ontology enables automation of some of the steps presented above, which tailors results of the filtering process more precisely to business needs.

4. Conclusions and future work

The aim of this work is to fill a gap between research conducted in the security domain and in the Semantic Web services domain. In order to add security aspects to the SWS description, the extension to the WSMO security ontology was defined. As a proof of the concept we provided an example from the encryption domain. Moreover, we have shown how to take security parameters into account during the constraint-based stage of filtering. In fact, the creation of a security ontology made it possible to quantify risks related to the use of Semantic Web services in enterprise information systems. Additionally, the security ontology exploited in the process of Web services filtering enabled the production of results that are more precisely tailored to the user's needs. Future work includes the incorporation of security profiles into user requests and extending the ontology-based filtering stage to include information inferred from the security ontology.

References

[1] Abramowicz, W. et al.: A survey of QoS issues for the needs of Web Services Profiling. In the Proceedings of ISCA 18th International Conference on Computer Applications in Industry and Engineering 2005, Honolulu

[2] W3C, Web Services Description Language (WSDL) Version 2.0, 2005

[3] UDDI Spec Technical Committee, UDDI Version 3.0.2. 2004

[4] Roman D. et al.: WWW: WSMO, WSML, and WSMX in a nutshell, Proceedings of the ASWC 2006, Beijing

[5] Heffner R., et al. Planned SOA Usage Grows Faster Than Actual SOA Usage, Forester Report, February, 2007.

[6] W3C: Owl-s: Semantic markup for web services. http://www.w3.org/Submission/OWL-S/ (November 2004)

[7] WSDL-S, http://lsdis.cs.uga.edu/projects/WSDL-S/wsdl-s.pdf

[8] Abramowicz, W., et al.: Architecture for Service Profiling, Proceedings of the MDA4SOA during ISWC 2006

[9] Cryptographic Key Length Recommendation, http://www.keylength.com/, 2007.

[10] Tsesmetzis, D.T., et al.: QoS awareness support in Web-Service semantics. In the Proceedings of International Conference on Internet and Web Applications and Services (ICIW'06), Guadeloupe, French Caribbean, 2006

[11] Mani, A. , Nagarajan, A.: Understanding Quality of Service for Web Services", IBM developerWorks, 2002

[12] Lee, K., Jeon, J., Lee, W., Jeong, S., Park, S.: QoS for Web Services: Requirements and Possible Approaches. W3C Working Group Note, November 2003

[13] Endrotti C., Efficiency Analysis and Comparison of Public Key Algorithms, Conference of PhD Students in Computer Science, 2002.

[14] Zhou, C., Chia, L., Lee, B.: DAML-QoS Ontology for Web Services. Proceedings of ICWS04, San Diego

[15] Tian, M., et al.: A Concept for QoS Integration in Web Services. In WQW 2003, Rome, 2003.

[16] Bianchini, D., et al.: QoS in ontology-based service classification and discovery. In WebS 2004), Saragossa,

[17] Lenstra A., Unbelievable Security. Matching AES Security Using Public Key Systems LNCS Vol. 2248, pp. 67-86, 2001

[18] S. Ran, "A model for web services discovery with QoS", ACM SIGecom Exchanges, 4(1):1-10, 2003

[19] Abramowicz, W. et al., Application-oriented Web services Filtering, in Proceedings of the International Conference on Next Generation Web services Practices (NWeSP 2005), pp 63-68, IEEE August 2005

[20] Feier, C. et al. : Towards intelligent web services: The web service modeling ontology (wsmo). In: Proceedings of the International Conference on Intelligent Computing (ICIC) 2005, Heifei, China (August 2005), pp. 23–26

[21] Toma, I., Foxvog, D.: WSMO - non-functional properties in web services. TR D28.4 v0.1, DERI 2006

[22] Denker, G., et al.: Security for daml web services: Annotation and matchmaking. In: ISWC 2003, pp. 335–350

[23] W3C: DAML-S (and OWL-S) 0.9 draft release. http://www.daml.org/services/damls/0.9/ (May 2003)

[24] Kagal, L., Paoucci, M., Srinivasan, N., Denker, G., Finin, T., Sycara, K.: Authorization and Privacy for Semantic Web Services. IEEE Intelligent Systems (Special Issue on Semantic Web Services) 19(4) (July 2004) 50–56

[25] Kim, A., et al. : Security ontology for annotating resources. In: OTM Conferences (2). (2005) 1483–1499

[26] Bruijn, J., et al.: The web service modeling language wsml. Technical Report D16.1v0.21, DERI (October 2005)

[27] Ekelhart, A., et al.: Security ontologies: Improving quantitative risk analysis. In: HICSS'07, pp. 156–162

[28] Ekelhart, A., et al: Ontology-based business knowledge for simulating threats to corporate assets. In: PAKM'06. Volume 4333 of Lecture Notes in Artificial Intelligence., Vienna, pp. 37–48

[29] Barker, E., et al.: Recommendation for key management part 1: General. NIST Special Publication 800-57, 2007

[30] Schneier, B. et al.: Performance comparison of the AES submissions, 1999