IJWA

An Approach to Discover Web Service Providers Based on Security Support

Urjita Thakar[1], Nirmal Dagdee[2]
[1]Department of Computer Engineering
Shri G.S. Institute of Technology & Science
23, Park Road
Indore 452003 (MP) India
urjita@rediffmail.com

[2]Department of CS & IT
S.D. Bansal College of Technology
A.B. Road, Umaria
Indore (MP) India
nirmal_dagdee@rediffmail.com

***ABSTRACT*:** *Web service invocation occurs over the Internet, which is a public network. This increases concern of the requester for adequate security of his private data. While using the web services, the security requirements of the user usually pertain to confidentiality, integrity, authentication and non-repudiation for the input information that the user provides and the information received as output from different services. For message level security, specifications such as WS-Security are available that use existing algorithms. Different service providers may use different algorithms for security; however, there is presently no mechanism available to the user to discover the service providers supporting the algorithms of their choice. In this paper this important issue has been addressed and an approach to enable a service provider to publish the information related to the security support offered by it has been presented. The presented approach facilitates the service requester to discover the providers as per his security related needs. The presented method is general and easy to use as existing Application Programming Interfaces (APIs) for publishing and discovery of services in the registry are used.*

Keywords: Web service, User preference, Security, Service Registry, Service Discovery

Received: 11 December 2008; Revised 29 January 2009. Accepted 12 February 2009.

1. Introduction

Popularity of Internet has given rise to its increased usage as a means of on-line businesses. Due to the inherent advantage of loose coupling, web services have become the technology of choice for such businesses. The basic web services architecture allows a service requester to discover the available services from service registry where the service provider publishes the services and the requesters may discover them from these registries.

Online businesses can be offered with some quality of service (QoS) related issues such as availability, accessibility, performance, reliability, security etc. The businesses usually include the cost of Quality of Service to the service charges while charging to the customer. The web service invocation occurs over the Internet, which is a public network. Security thus becomes highly important. While using the web services, the security requirements of the user usually pertain to confidentiality, integrity, authentication and non-repudiation for the input information that the user provides and the information received as output from different services. Also, the user (customer) is generally concerned not only about the security of its private information but also the cost required to pay for it.

Web services use Simple Object Access Protocol (SOAP) which is an XML based messaging protocol for communication. Some specifications such as WS-Security [1], XMLENC [2] exist that allow usage of some existing security algorithms for message level security. Generally, a service provider uses default algorithms to provide various security functions. Different service providers may offer different algorithms for various security functions. A requester may want to choose a provider based on the desired algorithms for the security function and the cost that suits his budget. For this, the requester should be able to select a web service not only based on its functionality, but also the available security support. Thus, it will be highly useful if the services are published in the registry by the service providers along with details about the security algorithms and their costs, and the requesters could find the services based on them.

In [3] the authors have proposed a method to facilitate composition of services based on the user's security preferences. However, no mechanism for publishing of security related information and discovery based on it has been discussed in this work. Due to non-availability of a mechanism to allow a user to discover the service suiting his security requirements, the existing specifications seem to be provider centric rather than user centric. Thus it is important to enhance the registries to allow businesses to publish the information about the security support available and the users to search the services that satisfy their security requirements.

Universal Description and Discovery Interface (UDDI) [4] is a popular specification for service registry. Current UDDI implementations allow some specific information to be published and search to be carried out on these limited attributes. These include attributes such as service name, business key, service key, location etc. Some methods proposed earlier [5] [6] [7] either need some extra resources or modifications need to be made in publish and search APIs. Thus these methods cannot be readily and easily be adopted by the service requester and providers as some changes are necessarily to be made at the registry as well as the requester.

In this paper we propose an approach to use the existing UDDI registry to allow the service provider to publish the information related to the security support. This registry can also be queried for discovery and precise selection of the service satisfying security constraints of the requester. Rest of the paper is organized as follows: Related work is discussed in section 2. In section 3 the requisite background is discussed. The proposed approach is discussed in section 4. In section 5, testing and results of the presented approach have been discussed. The features of the proposed approach are presented in section 6. The paper is concluded in section 7.

2. Related Work

Some work related to discovery of services based on QoS parameters has been done in the past. Liu et al. [5] have proposed a method to allow service providers/requesters to publish/search services based on service properties and constraints in a specific domain. In this work, an extra database has been used to store the non-functional information rather than using the existing facilities in the UDDI specification. Thus the existing 'publish' and 'find' APIs need to be modified. Guo et al. [6] have proposed a service repository which extends UDDI registry with capability description of Web Service which is defined in OWL-S profile. It uses the united data structure to express semantic descriptions related to Web Services and ontology data.

Soydan et al. [7] developed a service repository that extends UDDI registries. Their approach combines ontology of attributes with evaluation data. ShuPing et al. [8] have proposed a QoS driven service discovery method based on extended UDDI with the help of Semantic Web technology. It supports Web service discovery guaranteeing QoS. A matching algorithm based on fuzzy correlation calculation has been proposed to improve the discovery accuracy.

3. Background

3.1 The UDDI Specification for Service Registry

Universal Description Discovery and Integration (UDDI) is a popular service registry specification given by OASIS [4]. Service provider publishes the services to the registry with some service related information such as business name, business description, business related service details such as service name, service description, publisher id, some details for accessing the services such as end point reference (target end point), protocol, etc. Service provider categorizes the services as per UDDI supported classification scheme such as NAICS, UNSPSC, ISO 3166, and provides some classification details of services in the registry. Service requester discovers the services from registry service and extracts the URL of Web Service Description Language (WSDL) document that contains more details of the service pertaining to service parameters and invocation.

3.2 Security in Web Services

With the increase in the use of web services which are delivered over the public Internet, there is a growing concern about security. Different web service providers may apply different approaches and levels of security to the service while communicating with the requester.

Security for web services pertains to providing authentication, authorization, confidentiality, traceability and auditability, data encryption, and non-repudiation. Each of these aspects is described below [9][10][11].

- Authentication: Users (or other services) who can access service and data should be authenticated.
- Authorization: Users (or other services) should have rights to access the protected services.
- Confidentiality: Data should be treated properly so that only authorized users (or other services) can access or modify the data.
- Traceability and Auditability: It should be possible to trace the history of a service when a request was serviced.
- Data encryption: Data should be enciphered.
- Non-Repudiation: A user cannot deny requesting a service or data after the fact.

Security needs to be ensured when the services are consumed by the service requester. Some specifications have been defined to ensure security in web services. The WS-Security [1] specification defines security at the message level. It uses other specifications such as XMLENC [2] for this. A specification such as WS-Securitypolicy [12] is used to define security assertions for web service use. The specification WSSecureconversation [13] defines extensions that build on WS-Security to provide a framework for requesting and issuing security tokens, and to broker trust relationships. The specification WS-Trust, [14] enables applications to construct trusted SOAP message exchanges.

4. Proposed approach

In the proposed approach, a web application is envisaged that acts as an agent. This agent obtains user's security requirement and finds the services from the UDDI registry matching them. The agent obtains user's requirements for service functionality as well as security requirements through a user interface as shown in figure 1. Discovery module of the agent then uses those requirements to discover services from the UDDI service registry. It displays the list of available matching set of services to the user as per the indicated security and cost related requirements. The requester may choose any one from this list.

4.1 Addition to the UDDI registry

The existing classification schemes of UDDI specification use tModels for the purpose of classification of services. In the proposed approach the existing classification scheme of UDDI registry has been extended to store information related to

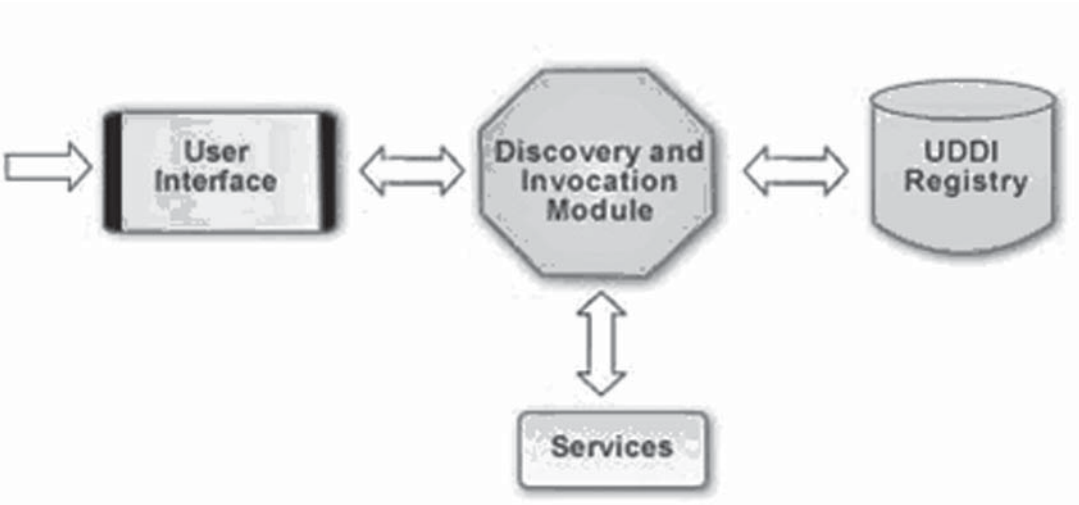


Figure 1. Agent Architecture

algorithms for various security functions and their corresponding costs. Some new tModels that have been added to the conventional UDDI registry are referred by service providers to put additional information pertaining to the available security support.

This information may then be used to search the service from registry. Existing APIs are used for publishing and discovery of services.

Following is the set of insert queries used to add information related to algorithms for confidentiality-

INSERT INTO TMODEL (TMODEL_KEY, AUTHORIZED_NAME, OPERATOR, NAME, OVERVIEW_URL, LAST_UPDATE) VALUES ('uuid: 48eb2518-c1bd-354f-92c9-21a53b0ff2u1','thakar','jUDDI.org','ubr-uddiorg: security: confidentiality', 'http://uddi.org/taxonomies/UDDI_Taxonomy_tModels.htm#confi', NULL);

INSERT INTO TMODEL_DESCR (TMODEL_KEY, TMODEL_DESCR_ID, LANG_CODE, DESCR) VALUES ('uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2u1', 0, 'en', 'UDDI Type Taxonomy');

INSERT INTO TMODEL_DOC_DESCR (TMODEL_KEY,TMODEL_DOC_DESCR_ID,LANG_CODE,DESCR) VALUES ('uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2u1',0,'en','tmodel for confidentiality algorithm');

INSERT INTO TMODEL_CATEGORY (TMODEL_KEY, CATEGORY_ID, TMODEL_KEY_REF, KEY_NAME, KEY_VALUE) VALUES ('uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2u1',0,'uuid:C1ACF26D-9672-4404-9D70-39B756E62AB4', 'types', 'categorization');

INSERT INTO TMODEL_CATEGORY (TMODEL_KEY, CATEGORY_ID, TMODEL_KEY_REF, KEY_NAME, KEY_VALUE) VALUES ('uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2u1',1,'uuid:C1ACF26D-9672-4404-9D70-39B756E62AB4','types','checked');

Similar set of queries are used to add information related to integrity and authentication.

In table 1 some details of the new tModels created corresponding to confidentiality, integrity and authentication are given. In table 2, the details of algorithms supported by various service providers are given. The services can be identified by the service key. The 'keyname' column shows name of the algorithm. This table also shows the cost charged by the service providers corresponding to the algorithms. The tModel key is used as categorization information for these security functions.

5. Testing, Results and Discussions

The jUDDI version 0.9 implementation of UDDI version 2.0 specifications has been used in the presented work. The proposed method doesn't require any modification in the 'publish' and 'search' APIs. For testing purpose some test services along with some security algorithms are published and various security and cost related requirements of the user are obtained. The services are then searched based on these. The interface to collect user's requirements is as shown in figure 2. The user inputs the desired service functionality along with the security and cost related requirements through this interface.

tModel Key	Operator	Name	Overview_URL	Last_Update
uuid:48eb2518-c1bd-354f-92c 9-21a53b0ff2u1	jUDDI.org	ubr-uddi-org:security:confidentiality	http://uddi.org/taxonomies/UDDI_Taxonomy_tModels.htm#confi	2008-08-19 12:46:03
uuid:48eb2518c1bd-354f-92c9-21a53b0ff2v1	jUDDI.org	ubr-uddi-org:security:integrity	http://uddi.org/taxonomies/UDDI_Taxonomy_tModels.htm#integrity	2008-08-19 12:56:35
uuid:48eb2518-c1bd-354f-92c9 -21a53b0ff2w1	jUDDI.org	ubr-uddi-org:security:Authentiction	http://uddi.org/taxonomies/UDDI_Taxonomy_tModels.htm#auth	2008-08-19 12:56:34

Table 1. Details of the tModels created for confidentiality, integrity and authentication

SERVICE_KEY	TMODEL_KEY_REF	KEY_NAME	KEY_VALUE
79B28C00-6DDC-11DD-8C00-FF31A9FFCFB6	Uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2w1	Digital_Signature	12
79B28C00-6DDC-11DD-8C00-FF31A9FFCFB6	Uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2w1	Digital_Signature_ with time stamp	15
79B28C00-6DDC-11DD-8C00-FF31A9FFCFB6	Uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2v1	SHA384	7
79B28C00-6DDC-11DD-8C00-FF31A9FFCFB6	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2v1	SHA512	9
79B28C00-6DDC-11DD-8C00-	uuid:48eb2518-c1bd-354f-92c9-	MARS	6
FF31A9FFCFB6	21a53b0ff2u1		
79B28C00-6DDC-11DD-8C00-FF31A9FFCFB6	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2u1	3DES	8
79B28C00-6DDC-11DD-8C00-FF31A9FFCFB6	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2u1	AES	10
A67AD780-6DCB-11DD-9780-95D7965677F7	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2w1	Digital_Signature	7
A67AD780-6DCB-11DD-9780-95D7965677F7	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2w1	Digital_Signature_with_tim e_stamp	12
A67AD780-6DCB-11DD-9780-95D7965677F7	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2v1	SHA1	3
A67AD780-6DCB-11DD-9780-95D7965677F7	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2v1	SHA256	5
A67AD780-6DCB-11DD-9780-95D7965677F7	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2v1	SHA384	7
A67AD780-6DCB-11DD-9780-95D7965677F7	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2v1	SHA512	9
A67AD780-6DCB-11DD-9780-95D7965677F7	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2u1	RC6	2
A67AD780-6DCB-11DD-9780-95D7965677F7	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2u1	MARS	4
A67AD780-6DCB-11DD-9780-95D7965677F7	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2u1	3DES	6
A67AD780-6DCB-11DD-9780-95D7965677F7	uuid:48eb2518-c1bd-354f-92c9-21a53b0ff2u1	AES	8

Table 2. Algorithms for security functions supported by various service providers

For the input given by the user to obtain information about all the security related algorithms available with various service providers and giving no constraint on the cost, following screen shots show the list of all the service providers with the security algorithms and their costs.

The result shows that the service providers named TravelA has published its service with four algorithms for confidentiality, four algorithms for integrity and two algorithms for authentication and non-repudiation. The service provider named TravelB has published three algorithms for confidentiality, three algorithms for integrity and two algorithms for authentication respectively. The service provider TravelC has published two algorithms for confidentiality and integrity each and one for authentication respectively. TravelD provider has published the service with three algorithms for confidentiality, integrity and two for authentication respectively as shown in the figure 3.

Test case 1: The user requires travel service with security requirements as Confidentiality and integrity and cost requirement as maximum 8.

Following figure 4 shows the user's inputs for this requirement.

Figure 5 shows the screen shot displying the results given by the discovery module. Three service providers are shown along with names of algorithms available for confidentiality and authentication along with the cost that will be charged by them.

Test case 2: The user requires travel service with security requirements as integrity and authentication. The cost constraint is given as maximum 13.

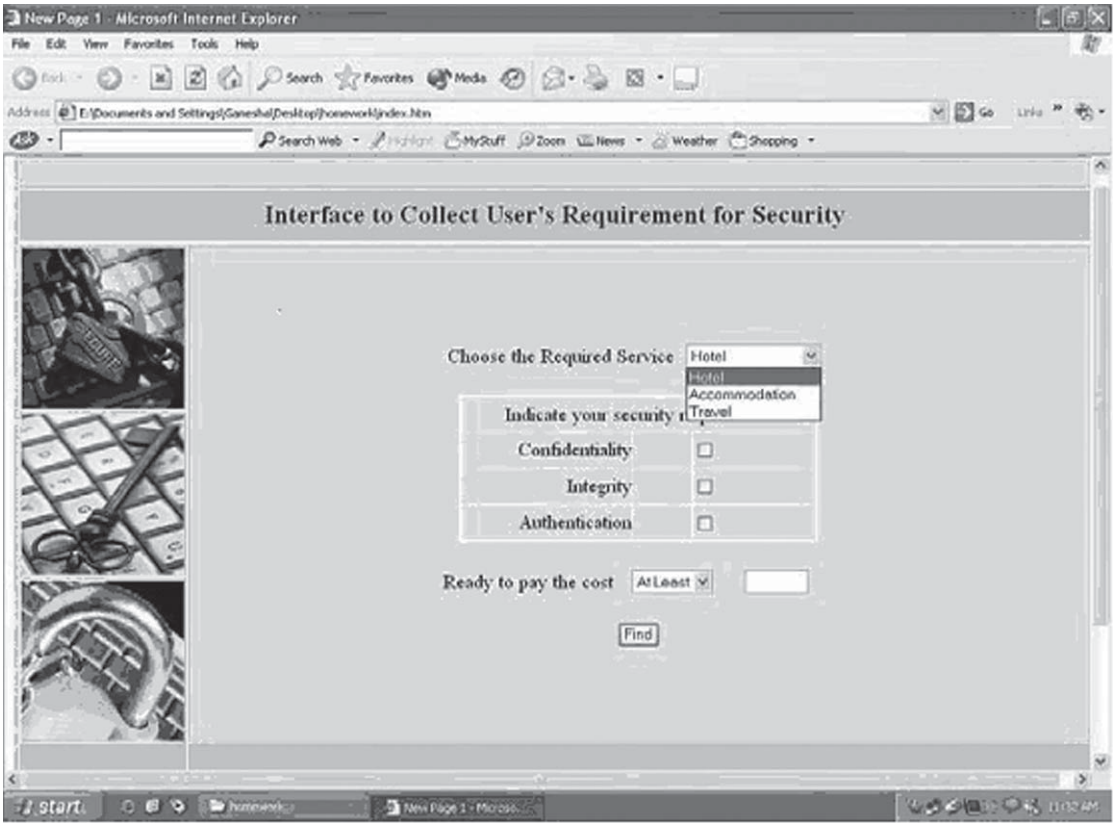
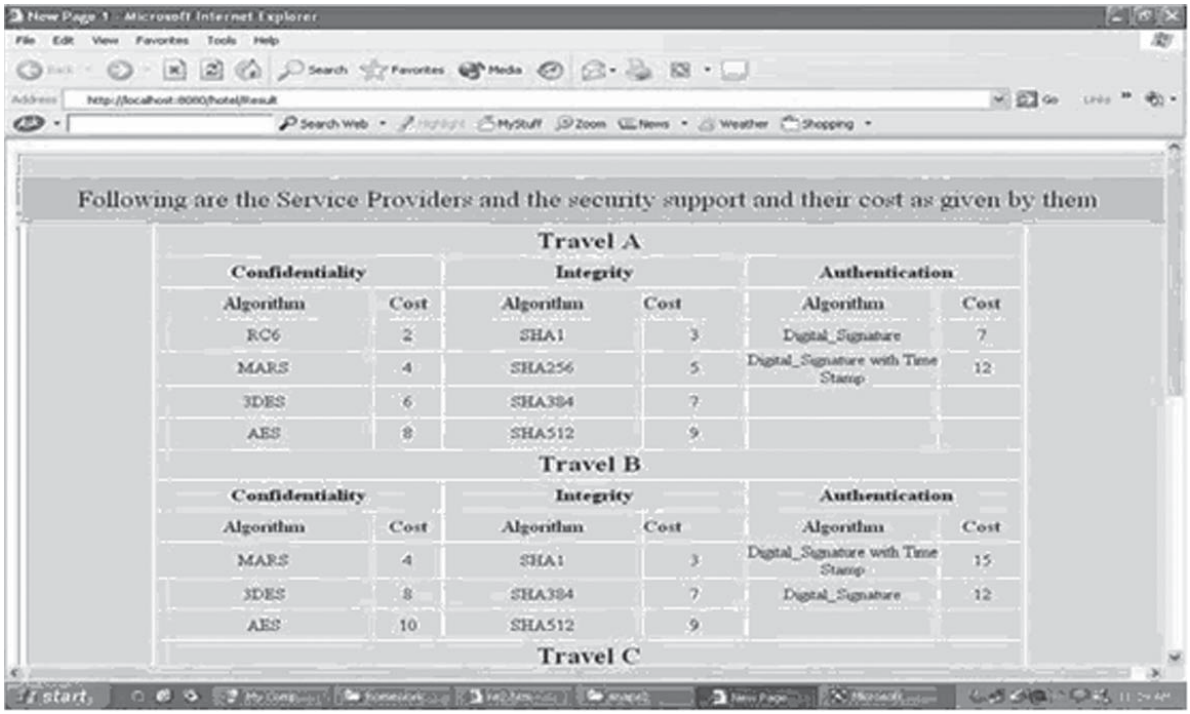


Figure 2. User Interface to collect security and cost related requirements of the user.



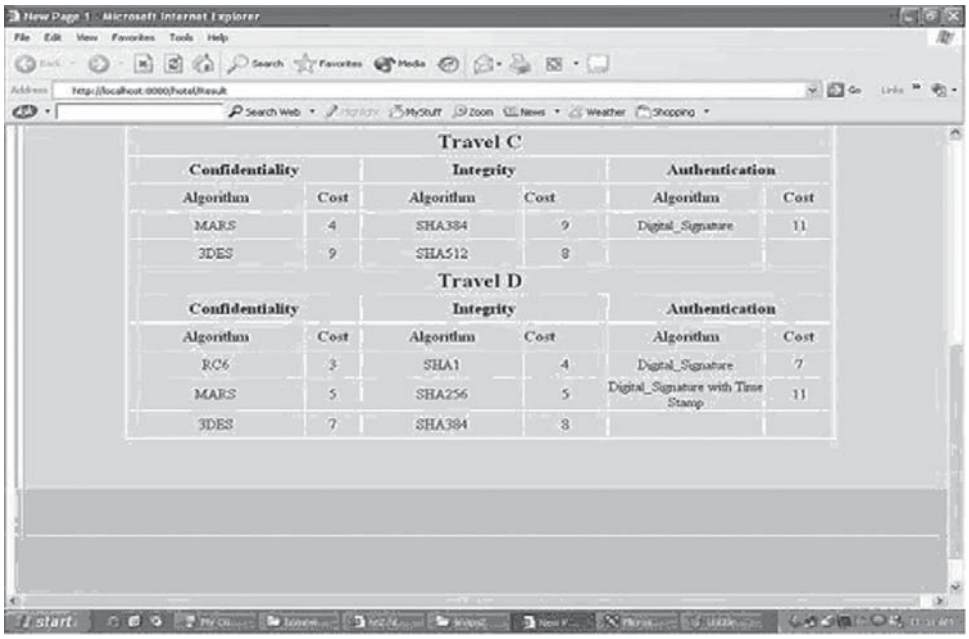


Figure 3. List of the service providers offering the desired functionality along with the available

The following screen shot 6 shows user's requirement.
There are two service providers offering two options for each can be discovered from the UDDI registry meeting user's requirements of functionality of 'Travel' and security requirement of integrity and authentication. When the number of available services in the registry is very large, results show that the list displayed to the user consists of services which are quite small in number. The list consists of only those services that match user's security and cost requirements. Thus it becomes very easy for the user to choose one from them.

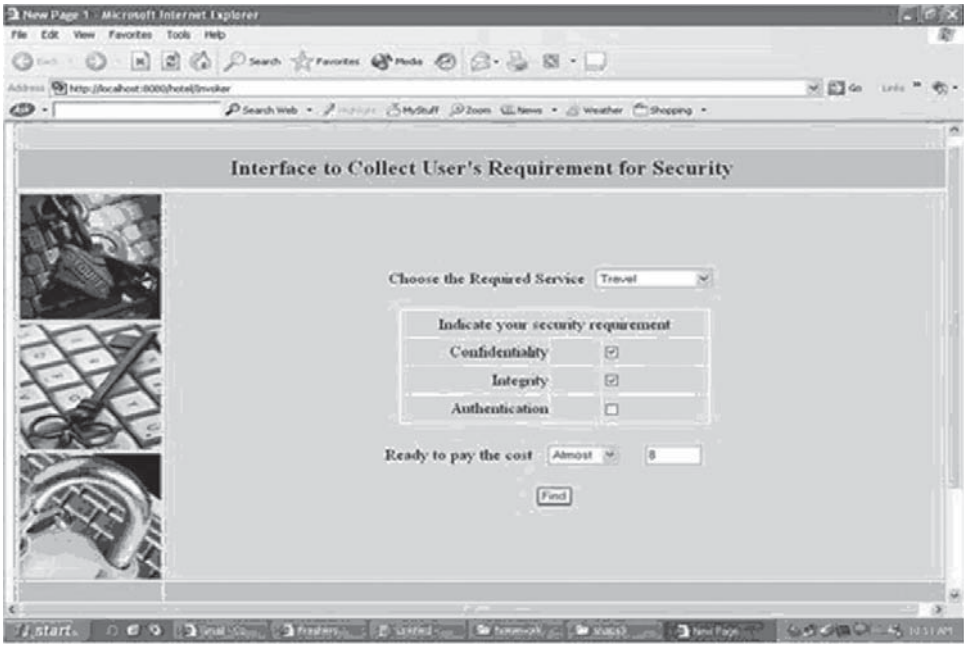


Figure 4. User's requirement is to have travel functionality with security requirements of confidentiality and integrity. The user is ready to pay maximum 8.

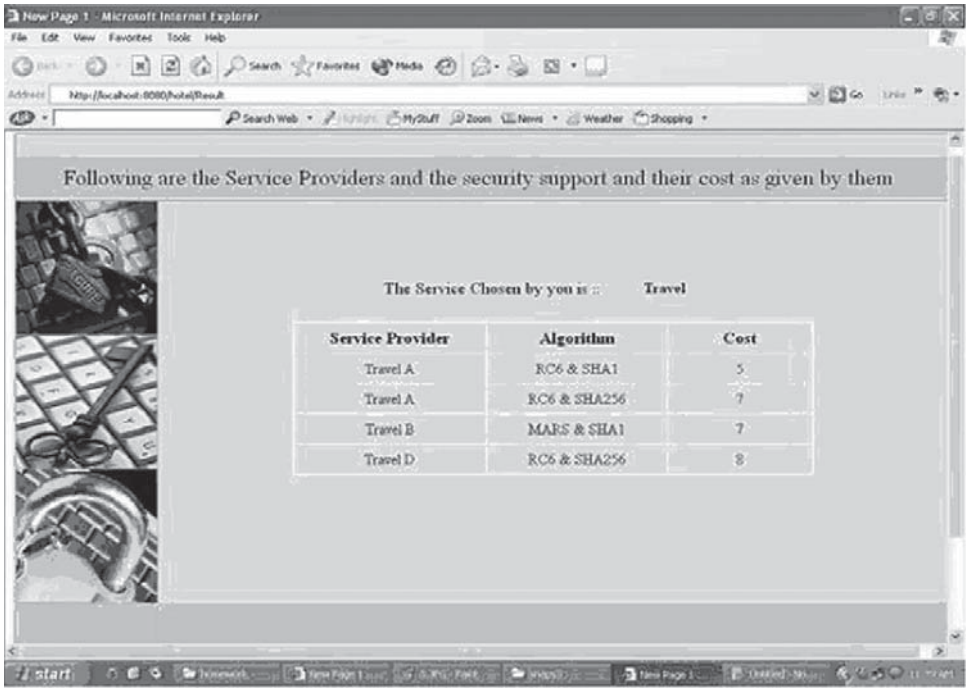


Figure 5. List of discovered service providers along with names of algorithms and their cost.

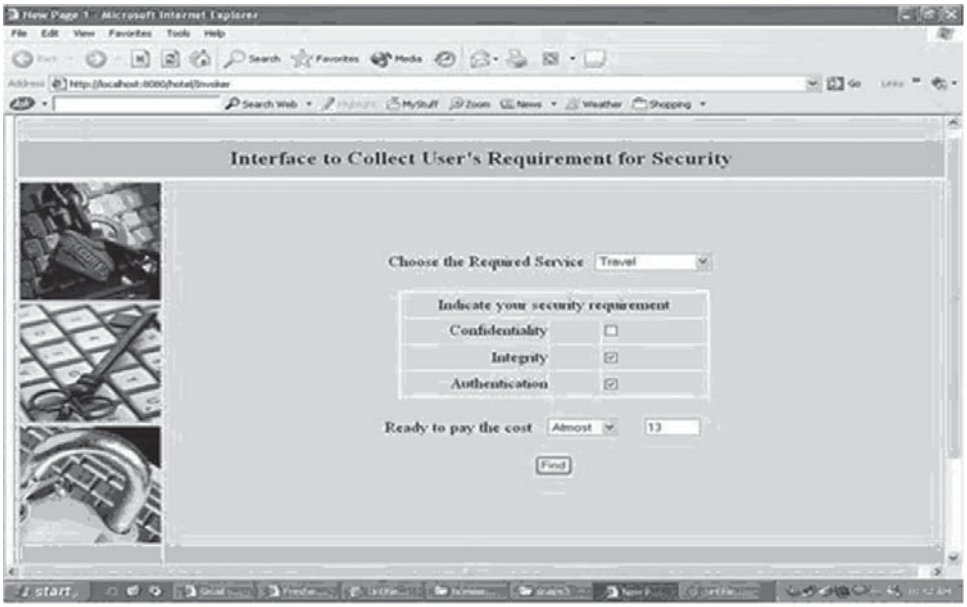


Figure 6. Screen shot showing the requirements of functionality of travel and security requirement of integrity and authentication with cost constraint of maximum 13.

6. Features of the proposed method

The results of the testing have been shown in the previous section. Here we list salient features of the system –

i. In this method, existing APIs of the UDDI implementation have been used. Thus the existing Registries can add this feature simply by creating new tModels for the security functions. Therefore no modification is needed at either provider or requester's end.

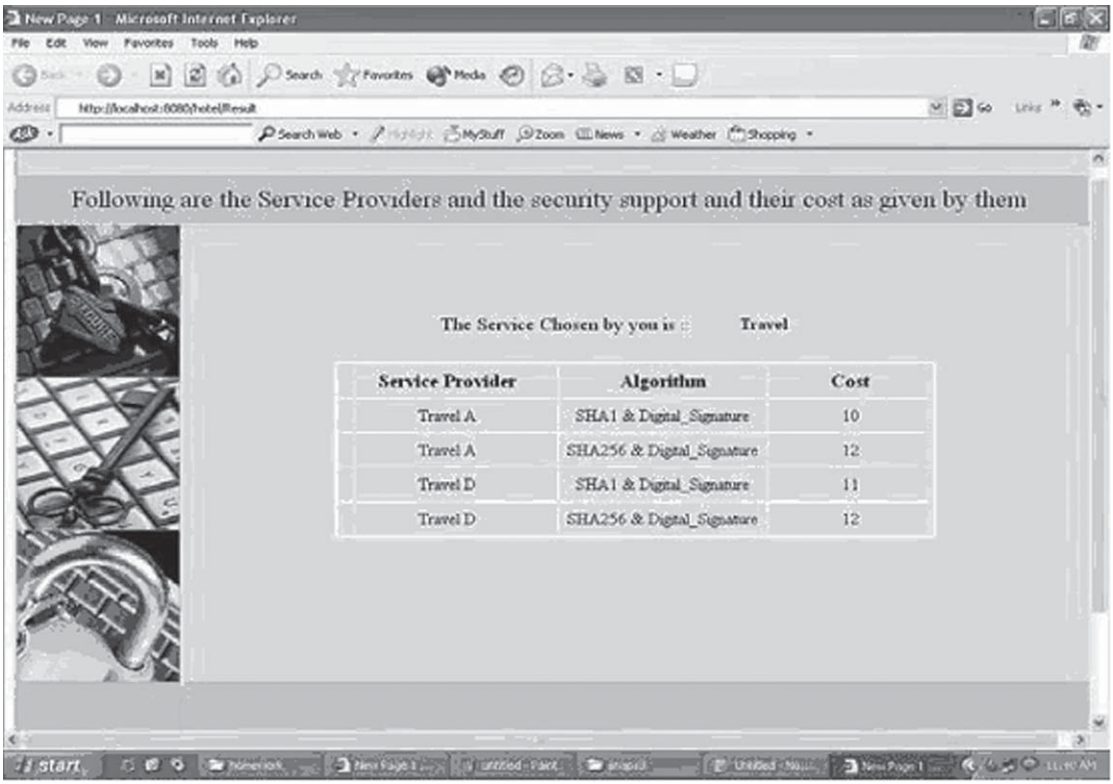


Figure 7. List of discovered service providers along with names of algorithms and their cost

ii. No extra resources are required thus the method is easy to adopt.
iii. By applying constraints on budget and security functions, the services matching more to the user's requirement are discovered. Thus search is more precise.
iv. The proposed approach is general and can be used to publish QoS properties of web services or additional features of the provided service. This approach can also be used to publish and discover services with different non-functional aspects such as availability, integrity, performance, reliability etc.
v. Though the information related to security can also be put in WSDL document, to determine this information, the requester will have to first locate the WSDL document from the registry and then open it to extract the information. Thus the process of selecting the service provider matching user's requirements becomes lengthy and inefficient. Therefore the proposed method is highly useful and increases efficiency of search for the user.
vi. The provider and requester should have the information about what and how the information can be published and the parameters based on which the services can be searched. Such details are provided to the service requester or provider at the time of registration.

7. Conclusion and Discussion

In this paper an important issue of facilitating a service requester to discover the services based on the security support provided by them has been addressed. At present, the service providers use some default algorithms to provide security to the requester's data and messages through existing specifications for web service security such as WS-Security. The presented method will help the overall system to be more user centric rather than server/service centric by allowing the user to choose the security providers that may use the security algorithms of his choice. This approach is highly helpful as extra resources such as additional database or use of semantic information are not required. The method also results in higher efficiency of search for the user as he has to choose from a smaller list of service providers that match his security requirements. This method is general and can be used to publish and discover services with different non-functional properties. This approach however requires that for each property a new tModel be added to the registry.

References

[1] Web Services Security: SOAP Message Security 1.1 4 (WS-Security 2004) 5 OASIS Standard Specifi cation, 1 February 2006, http://docs.oasis-open.org/wss/v1.1/.

[2] XMLENC, W3C Working Draft, XML Encryption Syntax and Processing, 04 March 2002, http://www.w3.org/TR/xmlenc-core.

[3] Nirmal Dagdee, Urjita Thakar. A Novel Scheme for Selection of Confi dentiality and Integrity Levels in Web Services Based E-Commerce Applications. *International Journal of Computer and Electronics Engineering* (Accepted for publication in Jan-Jun 2009 issue).

[4] OASIS, Universal Description, Discovery, and Integration (UDDI), UDDI Version 3.0.2, UDDI Spec Technical Committee Draft, Dated 20041019, Document identifi er: uddi_v3, Current version: http://uddi.org/pubs/uddiv3.0.2-20041019.htm, Latest version: http://uddi.org/pubs/uddi_v3.htm.

[5] Liu J, Gu N, Zong Y, Ding Z, Zhang S, Zhang Q. Service Registration and Discovery in a Domain-Oriented UDDI Registry. *In Proc. of the 2005 the 5th Int. Conf. on Computer and Information Technology, IEEE,* 2005. 276 - 283.

[6] Guo R, DongDong, Jiajin Le. Discovery for Web Services Based on Relationship Model. *In Proc of International Conference on Computer and Information Technology, IEEE,* 2006. 253-258.

[7] Soydan Bilgin A, Munindar, Singh P. A DAML-Based Repository for QoS-Aware Semantic Web Service Selection. *In Proc.of International Conference on Web Services, IEEE,* 2007. 368-375.

[8] ShuPing Ran. A Model for Web Services Discovery with QoS. *ACM SIGecom Exchanges,* spring 2003. 4;(1): 1-10.

[9] Basha SJ, Cable S, Galbraith M, Hendricks, Romin Irani, James Milbery, Modi T, Andre Tost, Alex Toussaint. Professional Java Web Services. Apress, Shroff Publishers and Distributors Pvt. Ltd., 2002.

[10] Esmiralda Moradian , Anne Hakansson**.** Possible attacks on XML Web Services. *International Journal of Computer Science and Network Security* 2006. 6;(1B): 54-170.

[11] Artem Vorobiev, Jun Han. Security Attack Ontology for Web Services. *Proceedings of the Second International Conference on Semantics, Knowledge, and Grid (SKG'06), IEEE,* 2006. 42.

[12] WS-SecurityPolicy 1.3 OASIS Standard 2 February 2009, http://docs.oasis-open.org/ws-sx/wssecuritypolicy/ v1.3/ws-securitypolicy.html.

[13] WS-Secureconversation [WS-SecureConversation 1.3, OASIS Standard, 1 March 2007, http://docs.oasisopen. org/ws-sx/ws-secureconversation/200512/ws-secureconversation-1.3-os.html].

[14] Web Services Trust Language (WS-Trust) WS-Trust 1.4 OASIS Standard 2 February 2009, http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html.

Author Biographies

Urjita Thakar: Urjita Thakar received her graduate and postgraduate degrees in Computer Engineering from Devi Ahilya University, Indore which is one of the premier universities of India. She is presently pursuing the Ph. D. degree in Computer Engineering from Rajiv Gandhi Technological University. She has 17 years of experience in teaching and research. Presently, she is working as Associate Professor in the Department of Computer Engineering at Shri G.S. Institute of Technology and Science, Indore, India. Her research interests include Web Technology, Semantic Web, Network Security etc. She has quite a few research papers to her credit.

Nirmal Dagdee: Dr. Dagdee received his graduate and postgraduate degrees in Computer Engineering from Devi Ahilya University, Indore which is one of the premier universities of India. He received his doctoral degree in 2003 from Rajiv Gandhi Technological University. He chaired the Department of Computer Engineering at Shri G.S. Institute of Technology and Science, Indore, India for over Seven years. He has about 22 years of academic experience. Presently, he leads a technical institute as Director. His research interests include Neural Networks, Genetic Algorithms, Semantic Web and Semantic Web services. He has several research publications in these areas to his credit.