A Flexible and Secure Web Service Architectural Model Based on PKI and Agent Technology

Maher Khemakhem
MIRACL, Lab, ISG,
University of Sousse,Tunisia

Wiem Rekik
HANA Group, ENSI,
University of Manouba, Tunisia
DIOM, Lab,
Universityof Jean Monnet
Saint Etienne,
France

Jacques Fayolle
DIOM, Lab, University
of Jean Monnet,
Saint Etienne, France

Abstract

Web services (WS) are considered amongst the most successful and convenient way which can take the advantages of the Internet in a flexible and reusable manner. WS constitute now the corner stone of any exchanged transaction over the Internet between any given organizations or more specifically between a client and a provider. Unfortunately, despite the considerable efforts made by researchers of this field in the recent years, security and flexibility of WS still constitute a big challenge. So far WS and the corresponding providers' URLs are, indeed, advertised on specific UDDIs (Universal Description, Discovery and Integration) where the only role to be achieved is limited to the advertisement. As such, after finding the requested service any given client contacts the right provider to negotiate the service access procedure. These first contacts between clients and providers are usually unprotected (not Encrypted) yielding enough room for Hackers to intrude into these unprotected messages. To solve this problem, we proposed in our previous study [12] a new idea based on the utilization of PKI and the improvement of the UDDI which must play in addition to its initial missions the role of a trust centre lading, thus, to secure WS.

In this paper, we pursue our study by attempting to solve also the flexibility problem of WS by using agent technology. More specifically, we are concerned by the flexibility that allows to any given provided WS to be used in different ways (several possibilities of composition with other WS to achieve several missions) by a given client within a fixed delay with a single contract.

1. Introduction

The World Wide Web is evolving into a medium for providing a wide array of e-commerce, business-to-business, business-to-consumer and other information based services. Web Services is emerging as the enabling technology that bridges decoupled systems across various platforms, programming languages and applications.

A web service is more than a program that can communicate and exchange data between applications and heterogeneous systems in a distributed environment [1] [2]. It can also offer a particular functionality relative to an application and/or a set of applications. As the authors in [3] and [4] define it, a WS is an application (software) that can be accessed through the Web by using both Internet protocols (e.g., Single Object Access Protocol, SOAP) to communicate with any environment, and a standard language (Web Service Description Language, WSDL) to describe its own interface.

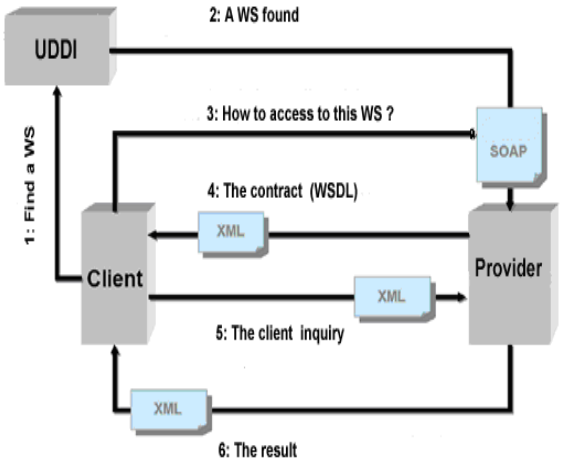

Figure 1.Web service advertising mechanism

WS are advertised over specific Directories named UDDI [5] where any client can first looking for the appropriate WS and then get the corresponding provider URL. After that, the client has to contact the adequate provider to get the access grant (which is a kind of certificate) to the requested WS. With this access grant, the client becomes able to access to this WS. Figure1. illustrates the advertising mechanism of WS.

Unfortunately, WS security still constitutes the big challenge; in fact, despite the multitude of security proposals done mainly by specialized consortium, organizations and researchers such as

W3C, OASIS [6], [7], [8], [9], [10], [11], [12], this problem seems to be not yet well solved. In our previous study [12], we have proposed a new model based on PKI and the improvement of the UDDI which must work rather as a trust centre to improve substantially the security of WS. In this paper, we are concerned by another WS challenge which concerns their flexibility in terms of possibility to be used in several ways by their clients within a fixed delay and with a single contract with the corresponding provider. In fact, a deep observation of the existing WS, the way they are implemented then provided and the user's needs leads to the conclusion that the WS flexibility problem is, indeed, unsolved and needs urgent solutions. Agent technology which provides, in particular, autonomy, intelligence and flexibility seems to be the adequate tool to reach our expectation.

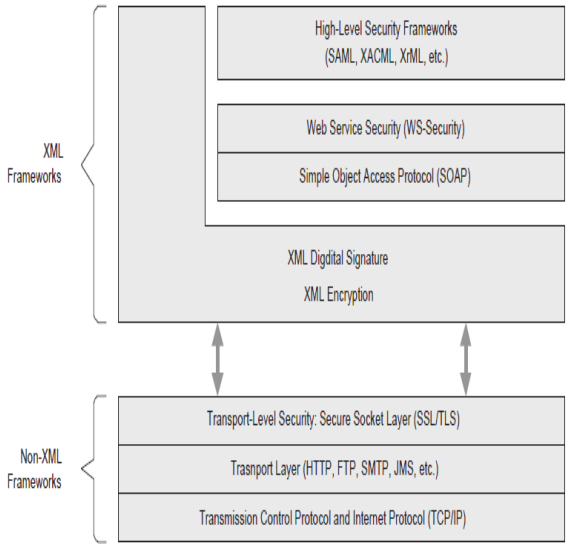

Figure 2. WS Security framework

The remainder of this paper is organized as follow: In section 2, we give an overview on the proposed solutions for WS security and we show, also, how the existing architectural model of WS is unsecure. In section 3, we present agent technology. In section 4, we describe the proposed model and show how agent technology can improve its own functioning by providing flexible and secure WS. Finally in section 5, we summarize the proposal and outline its perspective.

2. Background: Security of Web Services

Security of web services (WS) can be viewed from different sides because of the multitude of corresponding utilization. If we consider the case of utilization of WS for sharing information and services across organizations, then we can say that the corresponding security proposals and solutions are not bad but still require more improvements. But if we consider the case of public WS which can be provided to any given client by specialized providers then, we can say that the existing proposals and solutions lack a lot of security. This is mainly due to the non possession of the adequate security tools by most of the clients as we will explain next. Thus, our main concern in this paper focuses on this last case.
Specialized consortium and organization such as W3C, OASIS, ... have proposed several standards and solutions which attempt to provide security for WS. The framework illustrated by Figure 2. is the most commonly used one to provide security for WS. This framework shows that the security problem is divided into two levels; the transport level
and the application level.

2.1. Transport Layer Security (SSL/TLS):

Commonly, SSL/TLS [13] offers a secure channel on the transport channel. The server authentication is based on specific certificates and the client is authenticated via password or certificates.

2.2. Web Services Security (WS Security):

If we assume malicious hosts then the transport level security is possibly not enough. The transport level is controlled by the host, so the web service itself cannot verify the user credentials. The industry standard WS-Security [14] offers application level security as an extension to SOAP [15]. It defines how to integrate various XML Security concepts as XML Signature [17], XML Encryption [8] or the Security Assertion Meta Language (SAML) [18] into SOAP.

2.3. Security Assertion Meta Language (SAML):

SAML defines a XML-based framework for creating and exchanging authentication and authorization information.
The standard purpose of using SAML is to realize Web Single-Sign-On. The user authenticates at the first site, retrieves an authentication and authorization token and subsequently uses this token to access further services without
the need of re-authentication.

2.4. XML Access Control Specifications XACML:

XACML [19] is an extension to SAML that focuses on access control rights. XACML defines how to express access policies. Furthermore it specifies a request/response protocol between a policy decision and a policy enforcement point. XACML is considered the better way to implement role based access control (RBAC) [6] which restricts the WS accessibility according to predefined security policies and rules.

Unfortunately, despite the consistency and the robustness of these security tools, WS still require more protection especially if they are intended for the public as such we will explain in the next subsection.

2.5. Limit of the existing architectural model

So far, public WS and the corresponding providers URLs are advertised over specific UDDI where the only role to be achieved is limited to the advertisement. Consequently, users (clients) can just look for then find the required services for their different needs as illustrated in Figure 1. The first contact between any given client and the selected provider can't be protected (encrypted) especially if this client don't know the PKI (public key) of the provider and in addition he don't own a personal PKI. Consequently, any hacker can easily interfere in exchanged messages between any given client and provider to do what he wants. To deal with this problem, we proposed in our previous study [12] a new architectural model which can provide safe communications between involved parts and consequently secure WS as such we explain next. In this paper, we try to improve the overall functioning of our previous proposal by using agent technology which provides, particularly, autonomy, mobility, intelligence and flexibility.

3. The agent Paradigm

The Agent paradigm is very attractive because it mimics human or animal societies or communities in terms of interactions and coordination to achieve together common goals or to solve complex problems [20], [21], [22], [30].

An agent is a physical or logical entity that owns certain characteristics, sometimes looking like human ones. An agent is a semiautonomous entity like a human or an animal. It can conduct some tasks to achieve a goal, possibly without a central control and sometimes without a prior planning. In addition, an agent can interact and cooperate with other agents to achieve complex tasks. It is able sometimes to adapt its process face to some situations, e.g., environment changes. Overall, an agent can be mobile or stationary [23], [24], [25], [26].

3.1 Stationary agent vs mobile agent

Stationary agent is usually executed at the same location (for example on the same PC on a network), or mobile may be executed at different locations. An agent can be transferred to different locations for several reasons: special resource availability, execution load balancing, communication reduction, better QoS, etc. In fact, the use of mobile agents in some distributed applications is considered as a fundamental technology in next generation computing [27].

3.2 Multi agent system

Agents are often grouped to solve complex, often distributed tasks. The grouped agents form a multi-agent system where they communicate to achieve a common goal. A multi-agent system mimics humans or animals societies in several aspects especially in terms of interaction, cooperation and sometimes negotiation to solve complex problems. A multi-agent system can have the following advantages [25]: speed-up due to concurrent processing, less communication bandwidth requirements because processing is located nearer the source of information, more reliability because of the lack of a single point of failure, improved responsiveness due to processing, sensing and effecting being co-located and finally an easier system development due to the modularity inherent to the decomposition into semiautonomous agents. In addition, several researchers believe that the agent paradigm is the best way to solve efficiently distributed problems. Within any multi-agent system, communication is vital because agents solve collectively a given complex task by allocating one or more sub-problem per agent. Thus, agent interaction and cooperation are considered among the major issues in multi-agent system design and development. The difficulty resides in how to get agents to cooperate effectively. Interaction is further complicated whenever more than one agent can solve the same sub-problems or overlapped sub-problems but with different algorithms or data. Many works have been proposed to solve interaction and cooperation problems [20], [24].

3.3 Agent security

Secure communication between agents is considered as a challenge because of the inherent complexity. In fact, security of agents can be viewed at different levels such as agent authentication, message authentication several works have been proposed to secure agent communication [28], [29].

4. The proposed model and its improvement

Recall that our proposal aims at improving the security and flexibility of public WS which use the model illustrated by Figure 1.

This proposal was based, at the beginning [12], on two essential components; the first one is the necessity of utilization of the PKI by each involved party (clients and providers). Which means that any given client or provider must has its own certificate (private and public keys). The second component concerns the UDDI which must play also the role of a trust centre between clients and providers. This means that any given client or provider has to be subscribed first at the UDDI [12] (trust centre) in order to ease the authentication of each other's and leads, consequently, to secure WS. Our concern here

is, however, to improve the overall functioning of our model such that any provided WS will be, at the same time, well secure and more flexible. In fact, flexibility can be viewed from different sides; the one which we consider here is the possibility to provide WS that can be used by any given client for multiple purposes (can be composed with several other WS to achieve several tasks) within a fixed delay and with a single contract with the corresponding provider. So, we are faced to a complicated problem which requires using somehow intelligent WS.

Agent technology seems to be a convenient tool to reach our expectation. The idea consists to implement such WS in the form of intelligent and volatile agents which are able to reach our expectation. Of course and as we will explain next, the security and flexibility of provided WS will be amongst our main concerns.

4.1 The proposal outlines

Our previous proposal consists to use the PKI by each involved party and to extend the role of the UDDI in such a way that it also plays the role of a trust centre. Of course, to play the role of a trust centre, a prior subscription and registration of every involved party (a client or a provider) over the UDDI is mandatory. Every involved party must publish its own public key over the UDDI, during the subscription process, to allow the other party to communicate together in a safe manner. Thus, once subscribed, any given party can, easily, authenticate then access and use the public key of the other party to communicate together in a secure manner. As such we have suggested in [12], to reduce the communications of the trust centre, every provider shall record all useful data of his authenticated clients on a special list so that if one of them requests a WS another time, he don't need to turn back to the trust centre. In the same manner, we suggest that every client shall do the same things with his providers' data.

Recall that, our expectation here is to provide both secure and flexible WS in such a way that any one of them can be used several times for multiple purposes for a fixed duration and with a single contract with the provider.

In fact, if we can build any given WS in the form of an intelligent and flexible agent which is able:

- to be used in a several ways (can be composed with several other WS to achieve complex tasks and missions) by its client;
- to be self destroyed once the contract is expired or someone (client side) try to intrude it or violate it ;
- to send a notification message to its provider just before its self destroying;

then, the goal will be reached. The improvement of the existing agent development tools can lead to the implementation of such agents. We suppose next, that every WS is implemented as an intelligent and volatile agent which includes all the required methods interfaces and knowledge that allow it to achieve properly the mentioned missions.

4.2 The mechanism of the proposed model

As mentioned earlier, every involved party (client or provider) has to own its personal public and private keys to guarantee safe communications between them. A prior subscription and the publication of the public key of every involved party over the trust centre (UDDI) are also mandatory. Consequently, we can distinguish the following cases or scenarios:

- case where the provider is not yet subscribed over the trust centre. In such a case, he must be subscribed and registered at first to take an identity which allow him to be authenticated by the future clients ;
- case where the client is not yet subscribed over the trust centre In such a case, he must be subscribed and registered at first to take an identity which allow him to be authenticated by the future providers;
- case where the client is subscribed over the trust centre but he never made a contract with a given subscribed provider. In such a case, he must only take the public key of the target provider through the trust centre and then contacts directly this provider with an encrypted message by using this acquired public key. After receiving this message, the provider checks and authenticates the client from within the trust centre. If the authentication process succeeds, then they negotiate together the contract through encrypted messages till the agreement. In case of agreement (contract), the provider proceeds to the record of the useful data of his new client in his clients list to avoid the involvement of the trust centre next time and sends the requested WS which is in the form of an intelligent and volatile agent such as agreed and mentioned in the contract. In the same manner, the client records also, the useful data of his new provider in his providers list;
- case where the two parties knew already each others. In such a case, the trust centre will not be involved. These two parties will communicate in encrypted manner, authenticate each other and negotiate together the contract till the agreement.

Of course, provided WS which are in the form of intelligent and volatile agents will be sent from providers to clients in encrypted messages. This guarantee substantially the security of the exchanged messages and consequently the security of provided WS.

Two factors makes, indeed, very attractive our model; the first one concerns the substantially improvement of the security of the provided WS.

The second one concerns the flexibility of these WS which can be used in several ways by any given client during a fixed delay negotiated in the contract with the corresponding provider.

5. Conclusion

We proposed in this paper a new architectural model based on the utilization of PKI and agent technology in order to provide secure and flexible WS. The proposed model is, in fact, the result of the improvement of the existing one owing the fact that it conserves all features of its predecessor except that the role of the UDDI has been extended to become, in addition, a trust centre and the way to implement WS must be changed to be in the form of intelligent and volatile agents. We think that the improvement of the existing agent development tools can lead to the implementation of the proposed model. We are investigating in some of these problems in order to reach our expectation.

6. References

[1] Available at: "*htt://fr.wikepedia.org/w3c*," 2007.

[2] H. Kadima, V. Monfort, "*Les Web Services : techniques, démarches et outils*,", France, 2003.

[3] T. Melliti, "*Interopérabilité des services Web complexes. Application aux systèmes multi-agents'* Doctorate dissertation, University of Paris IX Dauphine, December, 2004, Paris, France.

[4] S. Rampacek, "*Sémantique, interactions et langages de description des services web complexes*' Doctorate dissertation, University of Reims Champagne-Ardenne, November,2006, Reims, France.

[5] P.Fremantle, D.Koeing, and C.Zenter. "*Building Web Services with java: making Sense of XML, SOAP, WSDL and UDDI*', 2nd Edition. (Developer's Library), Sams,, 2004.

[6] D. A. Haidar, F. C. N. Cuppens-Boulahia, and H. Debar. "An extended RBAC profile of XACMl'. *In Proceedings of ACM SWS'06*, Alexandria, Virginia, USA, November 3 2006.

[7] D. Booth, H. Haas, F. McCabe, and E. Newcome. Web services architecture. In Available at, *http://www.w3.org/TR/2003/WD-ws-arch-20030808/*, August 2003.

[8] M. A. Rahman, A. Schaad, and M. Rits. "Towards secure soap message exchange in a SOA'. *In Proceedings of ACM SWS'06*, Alexandria, Virginia, USA, November 3 2006.

[9] M. Khemakhem, H. B. Abdallah, and A. Belghith. "Towards an agent-based framework for the design of secure web services'. *In Proceedings of ACM SWS'08*, Alexandria, Virginia, USA, October 2008.

[10] J. Pamula and al. "A framework for establishing, assessing and managing trust in inter-organizational relationships'. *In Proceedings of ACM SWS'06*, Alexandria, Virginia, USA, November 3 2006.

[11] T. Imamura, B. Dillaway, and E. Simon. Xml encryption syntax and processing-w3c recommendation. In Available at, *http://www.w3.org/TR/xmlenc-core*, December 2002.

[12] W. Rekik, M.Khemakhem, A.Belghith and J.Fayolle. "PKI and UDDI based trust centre: an attempt to improve web service security" *IEEE, ICITST 2009*. LONDON, UK. November 2009.

[13] Freier, P. Karlton, and P. Kocher. "*The SSL protocol'*, version 3.0. In Internet draft, Netscape, November 1996.

[14] Nadalin, C. Kaler, r. P. Hallam-Bake, and R. Monzillo. "Web services security: SOAP message security 1.0'. *In Proceedings of ACM SWS'04*, OASIS, Alexandria, Virginia, USA, 2004.

[15] D. Box, D. Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Nielsen, S. Thatte, and D. Winer. "Simple object access protocol (SOAP) 1.1'. In available at *http://www.w3.org/TR/2000/NOTE-SOAP-20000508*., W3C Note, 2000.

[16] D. Eastlake, J. Reagle, T. Imamura, B. Dillaway, and E. Simon. XML encryption syntax and processing. In available at *http://www.w3.org/TR/xmlenc-core/*, W3C Recommendation, 2001.

[17] D. Eastlake, J. Reagle, D. Solo, M. Bartel, J. Boyer, B. Fox, B. Lamacchia, and E. Simon. "XMl-signature syntax and processing'. In available at *http://www.w3.org/TR/xmldsigcore/*, W3C Recommendation, 2002.

[18] OASIS. "*Security assertions markup language (SAML), version 2.0. working draft*'. In Organization for the Advancement of Structured Information Standards, OASIS, 2004.

[19] OASIS. "*XML access control markup language (XACML), version 2.0'*. committee draft. In Organization for the Advancement of Structured Information Standards, OASIS, 2004.

[20] J. Ferber, "*Les systèmes multi-agents. Vers une intelligence collective*," InterEdition, 1995.

[21] S. Russel et al, "*Artificial Intelligence: A modern Approach*," Prentice-Hall, 1995.

[22] M. Wooldridge et al, "Intelligent agents: theory and parctice,", *The Knowledge Engineering Review*, 10 (2), 115-152, 1995.

[23] G. Weis, "*Multi-agent Systems, a modern approach to distributed Artificial Intelligence*," MIT press, 1999.

[24] Victor R. Lesser, "Cooperative Multiagent systems: A personal View of the state of the art," *IEEE Trans. KDE*, Vol. 11 No1, January/February 1999.

[25] Caroline C. Hayes, "Agents in a Nutshell A very Brief Introduction," IEEE Trans. KDE, Vol. 11 No1, January/February 1999.

[26] M. Wooldridge et al, "A methodology for agent oriented analysis and design," Proc. Third International Conference on Autonomous agents (Agents'99), 69-76, ACM Press, Seattle, WA, USA, 1999.

[27] [P. Novak et al, "Communication Security in Multi-agent Systems", *CEEMAS 2003*, pp. 454-463, Springer-Verlag Berlin Heidelberg.

[28] S. R. Tate et al, "Mobile Agent Security through Multi-agent Cryptographic Protocols", *in Proc. the 4th International Conference on Intenet Computing*, 2003, pp. 462-468.

[29] IBM Electronic Service Agent, "*Transmission Security Information Inventory Information Privacy*", Electronic Services April, 2004, USA.

[30] M. Khemakhem and A. Belghith, "Towards trusted volunteer grid environments", *IJCNC, AIRCC*, Vol. 2, No2, March 2010, pp. 98-106.

