Application of Formal Methods to the Analysis of Web Services Security*

Llanos Tobarra, Diego Cazorla, Fernando Cuartero, and Gregorio Díaz

Escuela Politécnica Superior de Albacete,
Universidad de Castilla-La Mancha. 02071 Albacete, Spain
{mtobarra, dcazorla, fernando, gregorio}@info-ab.uclm.es

Abstract. Web Services technologies have introduced a new challenge for security protocols. Traditional security protocols cannot handle intermediaries and the flexibility of Web Services bindings. Thus, several proposals for introducing security in Web Services have been presented. One of these is *Web Services Security*. In this paper we illustrate how this protocol works, with an example, and analyse whether it is a good option guaranteeing the security of Web Services.

Keywords: Protocols and standards for WS, Security of WS, Secure Electronic Commerce.

1 Introduction

The rapid development of the World Wide Web in recent years has dramatically increased the exchange of information between clients and companies, and has also boosted electronic commerce transactions. Traditionally, the environment where electronic transactions occur consists of a web server that offers 'services' to human clients who use a web browser to select the information or products they wish to obtain. Nowadays this view is changing; companies wish to offer and use services automatically, i.e., they want to 'live' in a world where interoperability between various software applications running on separate platforms is possible, and, for example, Java can talk with Perl, and Windows applications can talk to UNIX applications.

A technology that has emerged recently and offers these kinds of transactions is Web Services [27]. Web Services implements a new Service Oriented Architecture (SOA), which is based on loosely coupled services. In a Web Services environment we find the following components:

- *Service providers*: they implement the web service and, in most cases, publish the service interface and the service registry information.

* This work has been partially supported by the MCyT project "Description and Performance of Distributed Systems and Application to Multimedia Systems" (Ref. TIC2003-07848-c02-02) and the JCCM project "Design and Implementation of Efficient Multimedia Systems by using Formal Techniques" (Ref. PAC-03001)

M. Bravetti et al. (Eds.): EPEW 2005 and WS-FM 2005, LNCS 3670, pp. 215–229, 2005.

- *Service brokers*: they allow clients to access the service interface and the implementation information.
- *Service clients*: they look for a service in a broker registry and then connect to the service provider in order to use it.

One of the most important issues in Web Services development is that each functional block should be platform or programming language-independent, and accessible for everybody. Thus, each block has to be described using an internet standard. The most important internet standards related to web services are the following:

- *XML* (eXtensible Markup Language)[26] is a markup language which underlies most of the specifications related to Web Services. XML is actually a metalanguage (a language for describing other languages) which lets you design your own customised markup languages for unlimited different types of documents
- *SOAP* (Simple Object Access Protocol)[22] is an XML-based messaging protocol used to encode the information in Web Services request and response messages before sending them over a network. SOAP messages are independent of any operating system or protocol and may be transported using a variety of Internet protocols, including SMTP, MIME and HTTP.
- *WSDL* (Web Services Description Language)[23] is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. Service providers use a WSDL document to specify available services. It also contains service access information.
- *UDDI* (Universal Description, Discovery and Integration)[19] is the client-side API and a server implementation based on SOAP. It stores and retrieves information about service providers and web services.

In order to use a web service, a client obtains a WSDL file in which a particular web service is described. If the client knows where the web service is located, they can retrieve it directly. Otherwise they can search for it using the UDDI protocol. Then, they prepare a SOAP request, which is an XML document that follows an XML schema. Each SOAP message is composed of a main element called an *envelope*. Each envelope has two main parts: payload data, included into the body, and one or more headers that contain control data such as addressing data, security items or quality options. The client sends the request through a transport protocol, usually HTTP. When a web service receives a request, it executes the requested actions and responds to the client with a SOAP response message. This message includes the result of the actions.

One of the main problems in using web services is that they are exposed to security attacks. Traditional security protocols, such as SSL [7], TLS [1] and IPsec[13], are used to protect communication between two agents in a network. Nevertheless, these are point-to-point technologies, whereas web services need end-to-end level security because the information does not travel straight to the endpoint; usually information needs to pass through several intermediate agents