Component-based Security Model for XML Messaging Services

Komathy K.
Center for Professional Development Education, Anna University, Chennai, India 600025.

Ramachandran V.
School of Computer Science and Engineering, Anna University, Chennai, India 600025.

Vivekanandan P.
School of Mathematics, Anna University, Chennai, India 600025.

Abstract

The objective of this paper is to enlighten the applicability of component technology in deploying security models to protect web transactions. The proposed component-based security model is a substitutive methodology for the conventional methods that are vendor dependent. This model prefers Extensible Markup Language (XML) format to represent web transactions, because of the multi-platform nature of the modern web applications. Granular approach is adopted in signing and encrypting the XML document. This paper also describes a distinct method used for key distribution. In addition, access control measures are also recommended in the model to protect data resources at the server against unauthorized access.


1. Introduction

A component-based model for the deployment of security components is based on a multi-tier, distributed object architecture. Security components can be replicated and distributed across any number of servers. Further, these server components can be deployed in any vendor's implementation. The protection methodologies proposed for XML messages are encryption and digital signature. The other design patterns proposed for this security model are façade beans and Model-View-Controller (MVC) pattern, which make the model to be scalable and reliable for distributed applications.

2. Design of the security model

The encryption component is responsible for protection of data in transit and as well as in storage. This component includes processes such as data-to-XML conversion, digital signature and encryption. An Advanced Encryption Standard (AES) [1] based on Rijndael algorithm is utilized to protect XML documents against brute-force attack [4]. The encryption component uses XML DOM [2] that provides an access to a number of node-types including elements, attributes, processing instructions, comments, and text nodes. The encryption component attaches the digital signature first and then encrypts the data subsequently. The security is provided to resources at any level in the hierarchical structure of the document.

Basic Authentication takes place through the submission of a user ID and a password. Authorization is also necessary because client can attempt to masquerade an identity to access the resource. When a legitimate client requests a resource, the authorization component first determines if the user has been granted permission to use that resource. The server creates a token and a session key for that client. The session key is generated based on the current time and the user-id. The token specifies the user’s access rights to access the data source such as retrieve, modify or delete data from the application database. Every new session is authenticated by a new token. This security token is highly useful in a network of active servers where the identity of the servers is known to each other. This component however, allocates a ‘guest’ profile to an unknown web client that permits only retrieval data from the database.

Figure 1. Secret key distribution through XML

Key distribution becomes one of the major problems in web-based communication. XML document could be used to carry the secret key.
Hiding the secret key in XML can be done using the attributes of the XML. Figure 1 portrays the process of hiding and extracting the secret key using XML.

3. A component-based approach

The component-based security model is implemented using BEA Weblogic server 6.1 that includes EJB 2.0 specification [3] for credit card application. This application comprises a network of active servers where the secure communication happens in XML. The layout of the security components in a client-server architecture using XML is illustrated in Figure 2.

The credit card details, submitted by the customer are converted into XML and forwarded in an encrypted format to the merchant’s application server, AS1. The controller component in AS1 receives the request and allocates the tasks to the EJB components. The server AS1 now acts as a client and initiates the session with the corresponding banker’s application server, AS2 for validating the credit card. The request before and after encryption from AS1 to AS2 is shown respectively in Figure 3 and Figure 4. The response from AS2 confirms the status of the credit card.

4. Conclusion

This comprehensive security model supports, integrates and unifies several popular security mechanisms and technologies including both symmetric and public key technologies in a way that enables a variety of systems to securely interoperate in a platform-neutral and language-neutral manner. Server can dynamically update users and groups by leveraging an external centralized security database or service such as LDAP or Kerberos. This proposed model might be enhanced in future to provide security for dynamic web services.

5. References


