Secure Business Transaction Models for Trading Game Contents Based on ebXML Applying Web Service Security Standards

Dongkyoo Shin and Dongil Shin

Department of Computer Engineering, Sejong University
98 Gunja-Dong, Gwangjin-Gu, Seoul 143-747, Korea
{shindk, dshin}@sejong.ac.kr

Abstract

ebXML (Electronic Business using eXtensible Markup Language) is an e-business standard developed by UN/CEFACT and OASIS, which enables enterprises to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes using Web services. Web service security technologies emerging recently have extensibility and flexibility suitable for security implementation such as encryption, digital signature, access control and authentication. In this paper, we propose ebXML business transaction models for trading game contents that allow trading partners to securely exchange business transactions by employing Web service security standard technologies. We show how each Web service security technology meets the ebXML standard by constructing the experimentation software and validating messages between the trading partners.

Keywords: electronic commerce, ebXML, Web service security, secure business transaction, game contents

1. Introduction

ebXML (Electronic Business using eXtensible Markup Language) is a suite of standards providing tools for negotiating business processes and contracts using Web services, developed by OASIS (Organization for the Advancement of Structured Information Standards) and UN/CEFACT (United Nations' Center for Trade Facilitation and E-business) [1, 2, 3]. However, since it was developed as a replacement for EDI (Electronic Data Interchange) it is often considered too complex to use for regular Web services [4]. A "Web service" is defined by W3C(World Wide Web Consortium) as a software system designed to support interoperable machine-to-machine interaction over a network with an interface described in a machine-processable format called WSDL(Web Services Description Language), and other systems interact with the Web service in a manner prescribed by its description using SOAP(Simple Object Access Protocol) messages [5].

The ebXML security challenge is to understand and assess the risk involved in securing this Web service framework based on existing security technology, and at the same time track emerging standards and understand how they will be used to resolve the risks that must be mitigated or reduced to an acceptable level to perform business functions. A list of key risks for ebXML is identified as unauthorized transactions and fraud, loss of confidentiality, error detection (application, network/transport, platform), potential loss of management and audit, and potential legal liability [2]. Because ebXML relies on Web service and some of the same underlying HTTP and Web-based
architecture as common Web applications, it is susceptible to similar threats and vulnerabilities. Web Service security is based on several important concepts including identification, authentication, authorization, integrity, non-repudiation, confidentiality and privacy [4]. The ebXML suite of standards provides support for security properties in contracts, but it does not fully support automatic security properties negotiation [4, 5].

There are well-known conventional security technologies that can be used by ebXML implementers to resolve the risks. Existing technologies such as user-id and password, PKI (Public Key Infrastructure) [6] and token can provide user identification and authentication to solve the unauthorized transactions and fraud problems in electronic business systems. For the loss of confidentiality problem, SSL (Secure Socket Layer) [7] and S/MIME (Secure Multi-Purpose Internet Mail Extensions) [8] are used to provide confidentiality and authentication of endpoints.

Web Service security technologies emerging recently have extensibility and flexibility suitable for ebXML security implementation such as encryption, digital signature, access control and authentication. XML digital signatures [9] and SAML (Security Assertion Markup Language) [10] can be exploited to solve the unauthorized transactions and fraud problems in electronic business systems. XML digital signatures are used in ebXML to provide data integrity on messages, existing authentication and authorization schemes as well as non-repudiation between entities. SAML is recommended to provide identification, authentication and authorization and often used with XACML (eXtensible Access Control Markup Language) [11] to allow or deny access to an XML resource. XML Encryption [12] is recommended to solve the loss of confidentiality problem. Also XKMS (XML Key Management Specification) [13] is recommended for key management as a substitute for PKI.

In this paper, we propose secure business Web Service models for trading game contents based on ebXML that allow trading partners to securely exchange XML based business transactions solving key risks by employing Web Service security technologies. We have also developed the experimentation software, which shows how each Web Service security technology meets the ebXML standard by checking messages between the modules.

2. Background

ebXML is a modular suite of specifications for the XML-based global infrastructure for e-business transactions, and provides a standard method to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes [1, 2, 3]. The technical infrastructure of ebXML is composed of the following major elements:

- **Messaging Service**: The actual information communicated as part of a business transaction. A message will contain multiple layers. On the outside layer, an actual communication protocol such as HTTP or SMTP must be used. SOAP is an ebXML recommendation as an envelope for a message "payload". Other layers may deal with encryption or authentication.

- **Registry**: The registry is a database of items that support doing business electronically. It specifies how applications interact with the registry service interfaces and the minimum information model which states the types of information that are stored about registry items.
Trading Partner Information: It consists of two specifications: CPP (Collaboration Protocol Profile) and CPA (Collaboration Protocol Agreement) [14]. The CPP provides the definition, in the form of a DTD or W3C XML schema, of an XML document that specifies the details of how an organization is able to conduct business electronically. It specifies items such as how to locate, contact, and other various information about the organization, including but not limited to the types of networks and file transport protocols it uses, network addresses, security implementations, and how it does business. The CPA specifies the details of how two organizations have agreed to conduct business electronically through combining the CPPs of the two organizations. A CPA can be used by a software application to configure the technical details of conducting business electronically with another organization. The CPA/CPP specification discusses the general tasks and issues in creating a CPA from two CPPs. However, it doesn't specify an actual algorithm for doing it.

Business Process Specification Schema: The Specification Schema provides the definition of an XML document that describes how an organization conducts its business. While the CPA/CPP deals with the technical aspects of how to conduct business electronically, it deals with the actual business process. It identifies the overall business process, the roles, transactions, identification of the business documents used, document flow, legal aspects, security aspects, business level acknowledgments, and status.

Core Components: A set of standard "parts" that may be used in larger ebXML elements. For example, core processes may be referenced by business processes. The core components are contributed by the ebXML initiative itself, while larger elements may be contributed by specific industries or businesses.

Some researches on ebXML application have been carried out by many companies, agencies and universities [15]. For example, a research group named CECID (Center for E-Commerce Infrastructure Development) at the University of Hong Kong developed a platform “Hermes” based on ebXML message specification [16]. The L2L (Library to Library) was proposed and set up as a service integration program based on SOA (Service Oriented Architecture) standards and ebXML, proposed by Sebastian in Ireland University [17]. Recently certified e-Document authority systems are built based on SOA and ebXML standard by some companies and agencies in Korea [18]. The research on the ebXML still goes on all over the world.

When a standard is deployed as openly using Web service, businesses are bound to have security concerns. Several organizations, including OASIS, W3C, the Liberty Alliance, and various members of industry have put together numerous security standards and techniques for securing Web services. These standards and techniques all complement or extend one another, but there are some conflicting or competing standards [4]. There are Web service security standards that deal with security issues related to electronic business as follows [19].

- XML signature [9, 20]: XML signatures are used to ensure that the content within an XML document hasn’t changed. When a document is received, the client system performs an XML signature decryption transformation, which distinguishes between content that was encrypted prior to signing and content encrypted after signing.

- XML Encryption [12]: Besides being able to use standard methods of encryption when transmitting XML documents, the W3C and IETF delivered a standard for encrypting the XML data and tags within a document. This would encrypt portions of a document, with the idea that only sensitive information needs to be protected.
● XKMS (XML Key Management Specification) [13]: The XKMS protocol is a standard maintained by the W3C. It defines a way to distribute and register the public keys used by the XML signature specification. XKMS is made up of two parts: the XML Key Registration Service Specification (X-KRSS) and the XML Key Information Service Specification (X-KISS). X-KRSS is used to register public keys, and X-KISS is used to resolve the keys provided in an XML signature.

● SAML (Security Assertion Markup Language) [10, 21]: SAML, managed by OASIS, handles the actual exchange of authentication and authorization requests and responses. An SAML request contains information such as authentication username and password, or other details about the individual making the request. This information is then delivered to an application designed to process it to allow or deny access to an XML resource. Three general kinds of assertion statements can be used: authentication, authorization decision, and attribute. These three statements are used at various times in an application to determine who the requestor is, what they are requesting, and whether or not their request has been granted.

● XACML (eXtensible Access Control Markup Language) [11, 22]: XACML is a specification from OASIS. It is used in conjunction with SAML and it provides a means for standardizing access control decisions for XML documents. XACML is used to define whether to permit requested access to a resource, whether it’s an entire document, multiple documents, or a partial document. XACML receives a SAML request to determine if access should be granted to a resource based on rule sets, or policies, that are defined by the provider. Once the policy is evaluated and returns a value to indicate whether or not access is granted, an SAML authorization decision assertion is returned, which is then processed accordingly.


A high-level use case scenario for two trading partners is explained based on the ebXML technical architecture specification [1] as follows. Company A will first review the contents of an ebXML Registry, especially the registered business processes that may be downloaded or viewed. Based on a review of the information available from an ebXML Registry, Company A can build or buy an ebXML implementation suitable for its anticipated ebXML transactions. The next step is for Company A to create and register a CPP with the registry. Company A might wish to contribute new business processes to the registry, or simply reference available ones. The CPP will contain the information necessary for a potential partner to determine the business roles in which Company A is interested, and the type of protocols it is willing to engage in for these roles. Once Company A is registered, Company B can look at Company A's CPP to determine that it is compatible with Company B’s CPP and requirements. At that point, Company B should be able to negotiate a CPA automatically with Company A, based on the conformance of the CPPs, plus agreement protocols, given as ebXML standards or recommendations. Finally, the two companies begin actual transactions.

Based on the scenario, we propose two ebXML business transaction models for trading game contents ensuring the trust relationship within the real trading partners. The first model performs a user authentication and updates the CPP in the registries. The second model performs business transactions within the trading partners. These
models will explain how each Web service security technology solves the risks for ebXML.

In the first model - Model 1, an ebXML client performs an update for its own CPP in the ebXML registries, where applying security modules to implement business processes satisfies security requirements. The premises for Model 1 are as follows:

- User registration of Company A is completed in the registries, SAML and the XKMS Web Services.
- Company A and B and the registries have trust relationships with SAML and XKMS Web Services.
- XKMS Web Service has a root role in the CA (Certificate Authority).
Each CPP of Company A and B is updated when modification is necessary.

- User based policy documents in XACML format are implemented in each registry.
- Messaging between business entities is based on HTTP-SOAP protocol and XML Signatures and XML Encryptions are applied for secure messaging.

The procedure for Model 1 is presented in the form of a sequence diagram in Figure 1, where each box in the diagram denotes a Web Service or an application program. Each step denoted by an arrow and number in the diagram is explained as follows:

1. **Generation of login information**: A Client in Company A logs into the local ebXML intranet system through authentication using user-id and password. An SAML assertion request is generated from this authentication information.

2. **Authentication request**: Generated SAML assertion is transferred to the SAML Web Service to get an access to registry.

3. **Request of key verification information for digital signature**: The SAML Web Service requests the client’s public key information to XKMS Web Service to verify the received message.

4. **Extraction of key information**: XKMS Web Service extracts public key information.

5. **Response of key verification information**: Extracted client’s public key information is transferred to the SAML Web Service using response protocol.

6. **Message authentication and generation of assertion and artifact**: Authentication on the message is performed using the public key information, and then authentication assertion, attribute assertions, and artifact are generated.

7. **Response of authentication assertion, attribute assertion and artifact**: Generated assertions and artifact are transferred to the client using response protocol.

8. **Generation of CPP update requests**: Received assertions and CPPs to be updated, and update requests are assembled in the message in the SOAP format.

9. **Access to Registry A**: An artifact generated by SAML Authority is transferred to Registry A.

10. **Req. of ebXML Client’s authentication information**: To request ebXML Client’s authentication information, ebXML Registry of Registry A sends the artifact, which is received from ebXML Client, to ebXML Client.

11. **Check the integrity of returned artifact**: ebXML Client verifies the integrity of returned artifact from ebXML Registry of Registry A.

12. **Transfer of CPP updated requests and assertions**: A generated message is transferred to the registry A.

13. **Message analysis**: The registry A analyzes the received message and perceives the requests. The update of CPP is possible when the user of the client has a role of “ContentOwner”. To check the role, the positive response from the XACML Web Service is required.

14. **Transfer of attribute assertion**: Attribute assertion of the client is transferred to the XACML Web Service.
(15) **Response of result assertion**: Authorization decision assertions are generated and transferred to the registry A, if the attribute assertion meets the XACML policy for documents.

(16) **Analysis of result assertion**: The registry analyzes the response from the XACML Web Service, and proceeds to the CPP update in case it receives authorization decision assertion. Otherwise, it cannot update CPP.

(17) **CPP update**: CPP is updated following the updated request.

(18) **Transfer of success/failure of CPP update**: Message on success/failure of CPP update is transferred to the client.

From (19) to (29) is the same to from (9) to (18).

In the second model – Model 2, two ebXML client exchange business transactions, where security requirements are satisfied by applying security modules to implement business processes. The premises for Model 2 are as follows:

- *Company A* and *B* have already exchanged CPA documents and agreed to use XML security technologies
- *Company A* and *B* have a trust relationship with XKMS Web Service.
- XKMS Web Service has a root role in the CA (Certificate Authority).
- Messaging between business entities is based on HTTP-SOAP protocol and XML Signatures and XML Encryption are applied for secure messaging.

![Sequence Diagram – Model 2: Exchange of Business Transactions](image)

**Figure 2. Sequence Diagram – Model 2: Exchange of Business Transactions**

The procedures for Model 2 are presented in the form of a sequence diagram in Figure 2, where Client 1 belong to *Company A*, and Client 2 belongs to *Company B*. An arrow denotes each step and number in the diagram and is explained as follows:

(1) **Message generation after CPA analysis**: Each client completes the generation of CPA for a business transaction, and Client 1 creates a transaction document.

(2) **Transfer of transaction document**: The transaction document is transferred from Client 1 to Client 2.

(3) **Request of key verification information for digital signature**: Client 2 requests Client 1’s public key information to XKMS Web Service to verify the received message.
(4) **Response of key verification information**: The extracted client’s public key information is transferred to Client 2 using response protocol.

(5) **Message Authentication and transaction processing**: Authentication on the transaction message is performed using Client 1’s public key information and the transaction is processed.

4. **Design and Implementation of the Experimentation Software**

We designed and implemented the experimentation software, which focuses on security for registry/repository and messaging, and then targets system performance for the two business scenarios mentioned in the previous section under a secure and reliable environment. XML Signature and XML Encryption are applied to the business transactions in the MSH (Message Service Handler) of ebXML client applications, registry, XKMS and SAML Web Services. Major security modules are shown in Figure 3.

![Figure 3. Major Security Modules in the Experimentation Software](image)

We tested two models by analyzing the messages in each step from Figure 1 and 2. In Model 1, an assertion message is generated in the ebXML client. And then, this attribute assertion message is included in the body of a SOAP message and sent to the SAML Web Service, where the XML signature and XML encryption was applied to this SOAP message body. To verify digital signature in the received SOAP message, the SAML Web Service extracts a public key from <ds:KeyInfo> within this message and transfers this key value to the XKMS Web Service. The SAML Web Service performed message authentication using the key validation results.

The CPP update messages now include 3 payloads: the first one is the information of updated CPPs. The second one is a new CPP, and third one is a SAML attribute and
authentication assertion. These payloads are transferred in “SOAP with Attachment” formats with XML encryption applied.

The MSH of the registry analyzes the request of ebXML clients after validating and decrypting the received message. If the message is a CPP updated message, the MSH defines the user authorization to decide if the user is an appropriate user for the update by checking XACML policy documents. If he is an appropriate user, the authorization decision assertion is issued from a XACML Web Service and the CPP is updated. The resulting message is sent to the ebXML Client.

For Model 2, similar messages are generated according to the steps described in Figure 2.

5. Conclusion

In this paper, we proposed two business Web service models for trading game contents based on ebXML that allow trading partners to securely exchange business transactions by employing Web service security technologies. We have shown how each XML security technology meets the ebXML standard and solves the risks by designing and implementing experimentation software, and checking the messages.

We will continue research on the ebXML application system focusing on trading digital game contents and the advanced security model using Web service security.

Acknowledgement

This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2011.

References


Authors

Dongkyoo Shin received a B.S. in Computer Science & Statistics from Seoul National University, Korea, in 1986, an M.S. in Computer Science from Illinois Institute of Technology, Chicago, Illinois, in 1992, and a Ph.D. in Computer Science from Texas A&M University, College Station, Texas, in 1997. He is currently a Professor in the Department of Computer Science & Engineering at Sejong University in Korea. From 1986 to 1991, he worked in Korea Institute of Defense Analyses, where he developed database application software. From 1997 to 1998, he worked in the Multimedia Research Institute of Hyundai Electronics Co., Korea as a Principal Researcher. His research interests include XML Security, XML based middleware, multimedia application, biological database, mobile Internet and ubiquitous computing.

Dongil Shin received a B.S. in Computer Science from Yonsei University, Seoul, Korea, in 1988. He received an M.S. in Computer Science from Washington State University, Pullman, Washington, U.S.A., in 1993, and a Ph.D. from University of North Texas, Denton Texas, U.S.A., in 1997. He was a senior researcher at System Engineering Research Institute, Deajun, Korea, in 1997. Since 1998, he has been with the Department of Computer Science & Engineering at Sejong University in Korea where he is currently a Professor. His research interests include Mobile Internet, Computer Supported Cooperative Work, Object-Oriented Database, Distributed Database, Data Mining and Machine Learning.