International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:10, 2008

A Study on the Secure ebXML Transaction Models

Dongkyoo Shin, Dongil Shin, Sukil Cha, and Seyoung Kim

***Abstract*—**ebXML (Electronic Business using eXtensible Markup Language) is an e-business standard, sponsored 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. While there is tremendous e-business value in the ebXML, security remains an unsolved problem and one of the largest barriers to adoption. XML 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 that allow trading partners to securely exchange XML based business transactions by employing XML security technologies. We show how each XML security technology meets the ebXML standard by constructing the test software and validating messages between the trading partners.

***Keywords*—**Electronic commerce, e-business standard, ebXML, XML security, secure business transaction.

I. Introduction

IN the last few years, XML (eXtensible Markup Language) [1] has rapidly become the first choice for defining data interchange formats in new e-business applications on the Internet and the basis for e-business framework such as ebXML, RosettaNet and Web Services [2]. ebXML (Electronic Business using eXtensible Markup Language) is a set of specifications for XML-based global infrastructure for e-business transactions, being driven by OASIS (the Organization for the Advancement of Structured Information Standards) and UN/CEFACT (the United Nations' Center for Trade Facilitation and E-business), which enables enterprises of any size and in any geographical location to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes [3,8,9]. Nowadays, ebXML is regarded as an e-business Web Service, where Web Services are a standard proposed by the W3C (World Wide Web Consortium). In Web services, great interoperability and extensibility are offered thanks to the use of XML, and each Web Service can be combined in a loosely coupled way in order to achieve complex operations [9]. Components providing simple services can interact with each other in order to achieve business goals.

While there is tremendous e-business value in the ebXML, security remains an unsolved problem and one of the largest barriers to adoption. To ensure trust between business entities, a model for security is needed. The ebXML security challenge [3, 4, 8, 9] is to understand and assess the risk involved in securing this new web framework based on our 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 in order for the entity to perform business functions. List of key risks for ebXML is identified as follows [4].

- *Unauthorized transactions and fraud* – businesses might be more at risk because of the increased automation of transactions that could allow unauthorized access or fraud to be perpetrated.
- *Loss of confidentiality* – transactions or specific entity knowledge may be carelessly or deliberately opened on the network
- *Error detection (application, network/transport, platform)* – application errors can result in the transmission of incorrect trading information.
- *Potential loss of management and audit* – There is the potential for the loss of data if appropriate management and auditing are not implemented.
- *Potential legal liability* - Without the legislation for the legality of electronic transactions, the presentation and admissibility of electronic evidence is still immature and inconsistent between jurisdictions.

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) [21] and token can provide user identification and authentication to solve the unauthorized

Dongkyoo Shin (Phone: +82-2-3408-3242, e-mail: shindk@sejong.a.c.kr) to whom corresponding should be addressed, and Dongil Shin (e-mail: dshin@sejong.ac.kr) are with the Department of Computer Engineering, Sejong University, 98 Kunja-Dong, Kwangjin-Gu, Seoul 143-747, Korea.

Sukil Cha (e-mail: chasi@krf.or.kr) is with Korea Research Foundation, 25 HeunReung-Ro, Seocho-Gu, Seoul 137-748, Korea.

Seyoung Kim (e-mail:seykim@khidi.or.kr) is with the Center for Global Business, Korea Health Industry Development Institute, 57-1 Noryangjin-Dong, Dongjak-Gu, Seoul 156-050, Korea.

This study was supported by a grant of the Korea Health 21 R&D Project, Ministry for Health, Welfare and Family Affairs, Republic of Korea. (0412-MI01-0416-0002).

transactions and fraud problems in electronic business systems. For the loss of confidentiality problem, SSL (Secure Socket Layer) [6] and S/MIME (Secure Multi-Purpose Internet Mail Extensions) [7] are used to provide confidentiality and authentication of endpoints. Typical tools such as anti-virus software and intrusion detection software can be used to resolve error detection problems and PKI can be exploited to resolve potential loss of management and audit problems. The potential legal liability problem is resolved by policies and procedures including audits and controls.

XML 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 [11] and SAML (Security Assertion Markup Language) [14] 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) to allow or deny access to an XML resource. XML Encryption [10] 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 based on ebXML that allow trading partners to securely exchange XML based business transactions by employing XML security technologies. We have also developed the test software, which shows how each XML security technology meets the ebXML standard by checking messages between the modules.

This paper is composed of six sections. Section II includes overview of ebXML, XML security standards and single sign-on scheme. In section III, two ebXML business transaction models are proposed to securely exchange XML based business transactions among trading partners by employing XML security technologies. Section IV includes the design and implementation of the test software to validate messages between trading partners and section V includes the assay of the messages. Finally we conclude in section VI.

II. Background

XML Security technologies are recommended by the ebXML security team to be used in ebXML implementation [3]. Currently there are many XML security standards. We will briefly summarize the ebXML standard and related XML security standards. Especially, single sign-on feature using one of the XML security standards, SAML, is elucidated to assist the concept in the business transaction models.

A. Overview of ebXML

ebXML is a modular suite of specifications for the XML-based global infrastructure for e-business transactions, that enables enterprises of any size and in any geographical location to conduct business over the Internet. ebXML aims to provide a standard method to exchange business messages, conduct trading relationships, communicate data in common terms and define and register business processes. The direct sponsors of ebXML are OASIS (Organization for the Advancement of Structured Information Standards) and UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) [3, 9]. The vision of ebXML is to create a single set of agreed upon technical specifications that consist of common XML semantics and related document structures to facilitate global trade.

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 must be used (such as HTTP or SMTP). SOAP (Simple Object Access Protocol) 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. How applications interact with the registry (registry service interfaces) and the minimum information model (the types of information that are stored about registry items) that the registry must support is specified. Examples of items in the registry might be XML schemas of business documents, definitions of library components for business process modeling, and trading partner agreements.
- *Trading Partner Information*: It consists of two specifications: CPP (Collaboration Protocol Profile) and CPA (Collaboration Protocol Agreement) [18]. The CPP provides the definition (DTD and 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 (BPSS)*: The Specification Schema provides the definition (in the form of a DTD and W3C XML schema) 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 such things as the overall business process, the roles, transactions, identification of the business documents used, document flow, legal aspects, security aspects, business level acknowledgments, and status. It is used to configure the business details of conducting business electronically with another organization.
- *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.

Currently, development of the ebXML specifications is an on-going effort sponsored by OASIS and UN/CEFACT. Technical committees for the ebXML Registry, Messaging, Collaborative Partner, and Implementation are hosted by OASIS, and Business Process and Core Component work continues at UN/CEFACT.

The ebXML infrastructure is modular and with few exceptions these infrastructure components may be used somewhat independently. An illustration based on the ebXML technical architecture specification [3], as shown in Fig. 1, explains a high-level use case scenario for two trading partners. *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.

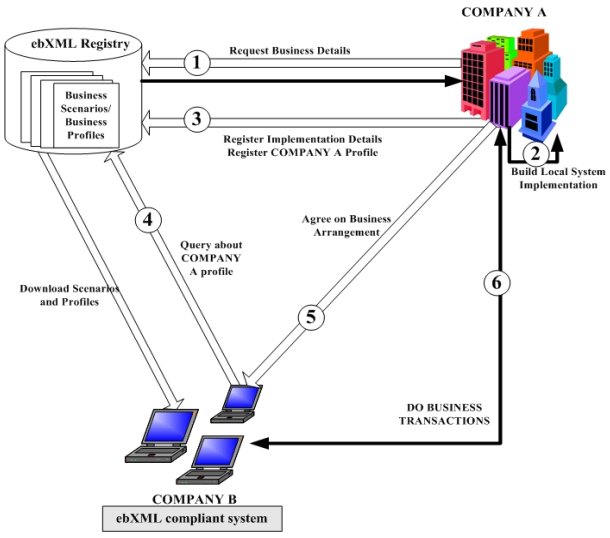

Fig. 1 Overview of ebXML interaction between two companies [3]

B. XML Security Standards

When a standard is deployed as openly as XML, businesses are bound to have security concerns. This section introduces and explains five proposed XML standards that deal with security issues.

1) XML Signature

```
<Signature>
  <SignedInfo>
    (CanonicalizationMethod)
    (SignatureMethod)
    (<Reference(URI=)?>
      (Transforms)?
      (DigestMethod)
      (DigestValue)
    </Reference>)+
  </SignedInfo>
  (SignatureValue)
  (KeyInfo)?
  (Object)*
</Signature>

1. "?" = Zero or one occurrence
2. "+" = One or more occurrences
3. "*" = Zero or more occurrences
```

Fig. 2 Syntax of XML digital signature

XML signature XML signatures are used to ensure that the content within an XML document hasn't changed [11, 12, 19]. 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. Anything encrypted after signing is decrypted, and data integrity is verified by comparing the result to the signature

included in the XML document. The syntax of XML signature is shown in Fig. 2.

2) XML Encryption

Besides being able to use standard methods of encryption when transmitting XML documents, the W3C and IETF estabilished a standard for encrypting the XML data and tags within a document [10]. This would let you encrypt portions of a document, with the idea that only sensitive information needs to be protected. Encrypting portions of a document with different keys would allow you to distribute the same XML document to various recipients, but the recipients would only be able to decrypt the parts relevant to them.

3) XKMS (XML Key Management Specification)

The XKMS protocol [13, 21] is a standard maintained by the W3C. It defines a way to distribute and register the public keys used by the XML-SIG 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.

4) SAML (Security Assertion Markup Language)

SAML [14, 20], managed by OASIS, is the counterpart to XACML that handles the actual exchange of authentication and authorization requests and responses. An SAML request is sent, via SOAP over HTTP, to a system with the appropriate means for processing the request. 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 with the intended goal of using XACML to allow or deny access to an XML resource. SAML uses an assertion schema maintained by OASIS. 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.

5) XACML (XML Access Control Markup Language)

XACML [15, 16, 17, 20] is a specification from OASIS. It's 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 true or false value to indicate whether or not access is granted, an SAML authorization decision assertion is returned, which is then processed accordingly.

C. Single Sign-On

The basic idea of single sign-on (SSO) is to shift the complexity of the security architecture to the SSO service and release other parts of the system from certain security obligations. The SSO service acts as the wrapper around the existing security infrastructure that exports various security features like authentication and authorization [23]. To support single sign-on, the system collects all the identification and user credential information from the user as a part of the primary sign-on. This information is used by SSO Services within the primary domain to support the authentication of the user to each of the secondary domains with which the user may interact.

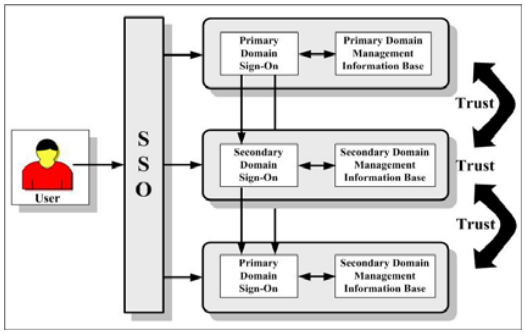

Fig. 3 Single sign-on to multiple services

For an approach for SSO implementation, token-based protocols such as cookies or SAML are used [24]. SAML has advantage over the cookie approach for SSO solutions since it is a standard suitable for facilitating site access among trusted security domains after single authentication. Artifacts, which have a role of tokens, are created within a security domain and sent to other security domains for user authentication. Since the artifacts sent to the other domains are returned to the original security domain and removed after user authentication, this resolves the problems of session keys being revealed and stolen tokens in the browser. In addition, artifact destination control is fully achieved since artifact identification is attached to the Uniform Resource Locator (URL) and redirects the message sent to the destination [14].

III. XML Business Transaction Models

We propose two ebXML transaction models for business scenarios ensuring the trust relationship within the real trading partners. The first scenario performs a user authentication and updates the CPP in the repository. The second scenario performs business transactions within the trading partners. In these scenarios, each XML security is constructed as a Web Service, which follows the Web Services standards proposed by the W3C (Word Wide Web Consortium) and OASIS [10,11,13,14,15].

1) Scenario 1: Update of CPP

In Scenario 1, an ebXML client performs an update for its own CPP in the ebXML registry, where applying security modules to implement business processes satisfies security requirements. In this scenario, a business partner already been authenticated can do 1-to-N businesses (from one partner to multiple partners) as well as 1-to-1 business (from one partner to one partner), because he can search and access to various CPPs registered in the ebXML registry. To offer more flexible access to multiple business partners, distributed registries need to be integrated, but it causes the problems of user authentication and security vulnerability. By applying single sign-on scheme, we can simplify user authentication and overcome the problems.

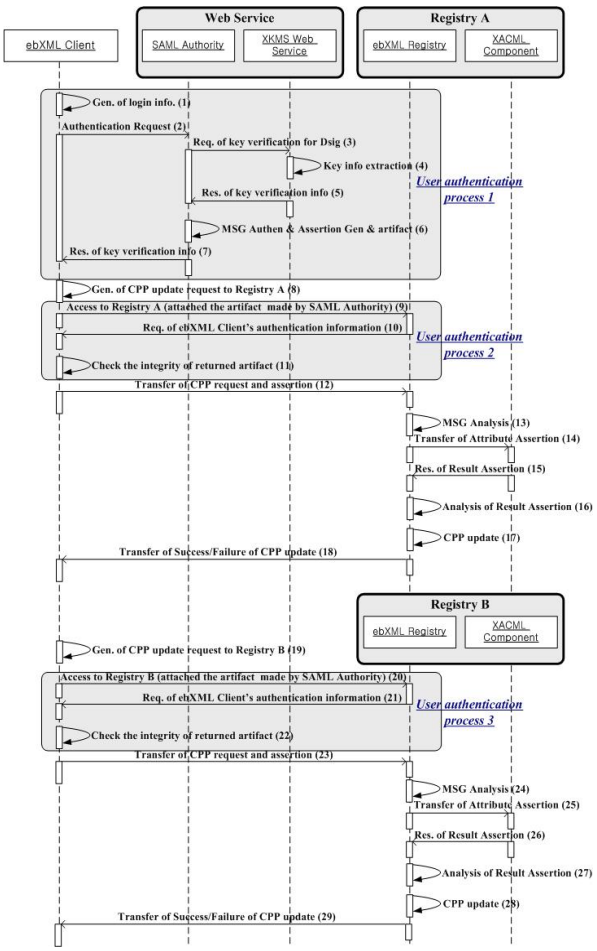

Fig. 4 Sequence Diagram – Scenario 1

The premises for Scenario 1 are as follows:

- User registration of Company A is completed in the registry, SAML and the XKMS Web Services.
- Company A and B and 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 Scenario 1 is presented in the form of a sequence diagram in Fig. 4, 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 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).

2) Scenario 2: Exchange of Business Transactions

In Scenario 2, two ebXML client exchange business transactions, where security requirements are satisfied by applying security modules to implement business processes. The premises for Scenario 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.

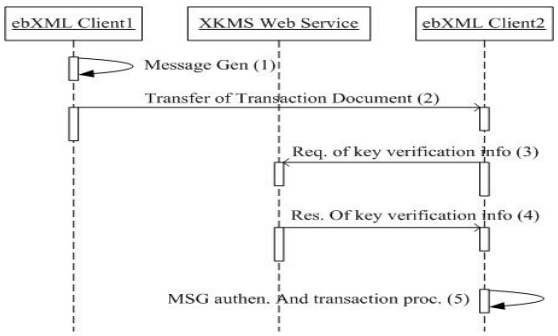

Fig. 5 Sequence Diagram – Scenario 2

The procedures for Scenario 2 are presented in the form of a sequence diagram in Fig. 5, where each box in the diagram denotes a Web Service or an application program. 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.

IV. Design and Implementation of the Test Software

We designed and implemented a test 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. The architecture for the test software is shown in Fig. 6.

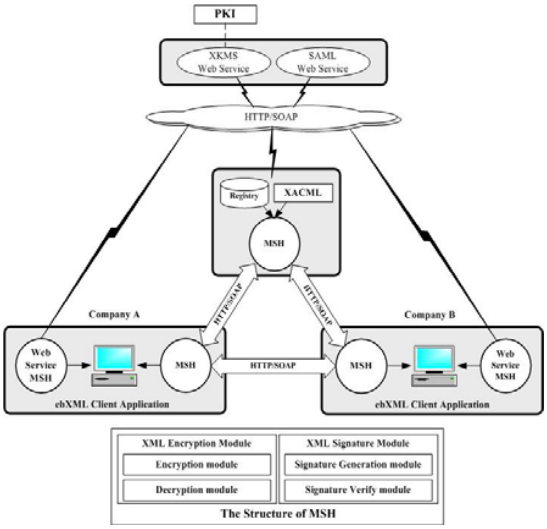

Fig. 6 Architecture of the ebXML test software

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 Fig. 7.

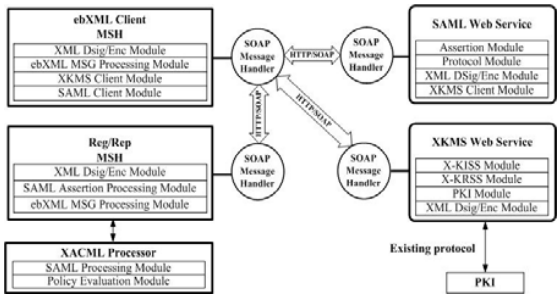

Fig. 7 Major Security Modules in the Test Software

V. Test Results

By analyzing the messages in each step from Figs. 4 and 5 two scenarios tested. In Scenario 1, the assertion message generated in the ebXML client is shown in Fig. 8.

```
<?xml version="1.0" encoding="UTF-8" ?>
<samlp:Request IssueInstant="2004-10-22T08:53:58Z"
               MajorVersion="1" MinorVersion="0"
               RequestID="ef153af0-02ae-11d7-b2e6-2398b88b0e62"
               xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
  <samlp:AttributeQuery>
    <saml:Subject
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
      <saml:NameIdentifier Format=""
NameQualifier="samlauthority.com/>
        companyA@companyA.com
      </saml:NameIdentifier>
      <saml:SubjectConfirmation>
        <saml:ConfirmationMethod>
          urn:oasis:names:tc:SAML:1.0:am:password
        </saml:ConfirmationMethod>
        <saml:SubjectConfirmationData>
          password
        </saml:SubjectConfirmationData>
      </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:AttributeDesignator
        AttributeName="//samlauthority.com/ebxml/regrep/role"
        AttributeNamespace="samlauthority.com/namespace/"
        xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion" />
  </samlp:AttributeQuery>
</samlp:Request>
```

Fig. 8 Request of Attribute Assertion

```
<?xml version="1.0" encoding="UTF-8"?>
<k:ValidateResult xmlns:k="http://www.xkms.org/schema/xkms-2001-01-20">
    <k:Result>Success</k:Result>
    <k:Answer>
      <k:KeyBinding>
        <k:Status>Valid</k:Status>
        <d:KeyInfo xmlns:d="http://www.w3.org/2000/09/xmldsig#">
          <d:KeyValue xmlns:d="http://www.w3.org/2000/09/xmldsig#">
            <d:RSAKeyValue>
              <d:Modulus>
              ovig7pVPyrVlhp3BS/TsO/+Fvoa3I8mtzr0aPXVJ
              EgIBrMuP8HUF+1F41EerIwTkoJS7cJX9KqrzY
              okmLS/NA09V/3LNwO74tzQRNQ71V7Ifi7DBQ
              WKuJKGBPRvSSIgYBA9KA5xRLBhXPr92Yrj
              gUcJKZ2dM6TNxBzE7JGj0U=
              </d:Modulus>
              <d:Exponent xmlns:d="http://www.w3.org/2000/09/xmldsig#">
                AQAB
              </d:Exponent>
            </d:RSAKeyValue>
          </d:KeyValue>
          <d:KeyName>companyA@companyA.com</d:KeyName>
        </d:KeyInfo>
      </k:KeyBinding>
    </k:Answer>
</k:ValidateResult>
```

Fig. 9 XKMS Validate Service Request

And then, this attribute assertion message was included in the body of a SOAP message and was 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 requested message including this key value is shown in Fig. 9.

The response message from the XKMS Web Service is shown in Fig. 10. This response message was included in the body of SOAP message. Also XML signatures and XML encryptions were applied to this SOAP message body.

The SAML Web Service performed message authentication using the key validation results. Attribute and authentication assertions in the response message are shown in Fig. 11. In this figure, “ContentOwner” as a value of the <AttributeValue> element has a role in the ebXML registry.

```
<?xml version="1.0" encoding="UTF-8"?>
<k:Validate xmlns:k="http://www.xkms.org/schema/xkms-2001-01-20">
  <k:Query>
    <k:TransactionID>41139320-029a-11d7-b2e6-2398b88b0c62</k:TransactionID>
    <k:Status>Indeterminate</k:Status>
    <d:KeyInfo xmlns:d="http://www.w3.org/2000/09/xmldsig#">
      <d:KeyValue xmlns:d="http://www.w3.org/2000/09/xmldsig#">
        <d:RSAKeyValue>
          <d:Modulus>ovig7pVPyrVlhp3BS/TsO/voa3I8mtzr0aPXVJ
            EgIBrMuP8HUF+1F41EerIwTkoJS7cJX9KqrzYokmLS/N
            A09V/Jw3LNwO74tzQRNQ71V7Ifi7DBQWKuJKGBPRv
            SSIgYBA9KA5xRLBhXPr92YrjgUcJKZ2dM6TNxBzE7JG
            j0U=
          </d:Modulus>
          <d:Exponent xmlns:d="http://www.w3.org/2000/09/xmldsig#">
            AQAB
          </d:Exponent>
        </d:RSAKeyValue>
      </d:KeyValue>
      <d:KeyName>companyA@companya.com</d:KeyName>
    </d:KeyInfo>
  </k:Query>
  <k:Respond>
    <k:string>KeyName</k:string>
    <k:string>KeyValue</k:string>
  </k:Respond>
</k:Validate>
```

Fig. 10 XKMS Validate Service Response

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:Response IssueInstant="2004-10-22T08:54:00Z"
                MajorVersion="1" MinorVersion="0"
                ResponseID="eff98070-02ae-11d7-b2e6-2398b88b0e62"
                xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
  <samlp:Status>
    <samlp:StatusCode Value="samlp:Success"/>
  </samlp:Status>
  <saml:Assertion AssertionID="eff93250-02ae-11d7-b2e6-2398b88b0c62"
                  IssueInstant="2004-10-22T08:54:00Z"
                  Issuer="SamlAuthority" MajorVersion="1" MinorVersion="0"
                  xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
    <saml:Conditions NotBefore="2004-10-22T08:54:00Z"
                     NotOnOrAfter="2004-10-22T09:04:00Z"/>
    <saml:AuthenticationStatement AuthenticationInstant="2004-10-22T08:54:00Z"
                                  AuthenticationMethod="Password">
      <saml:Subject>
        <saml:NameIdentifier Format="#emailAddress" NameQualifier="samlauthority.com/">
          companyA@companyA.com
        </saml:NameIdentifier>
        <saml:SubjectConfirmation>
          <saml:ConfirmationMethod>
            urn:oasis:names:tc:SAML:1.0:am:password
          </saml:ConfirmationMethod>
          <saml:SubjectConfirmationData>password</saml:SubjectConfirmationData>
        </saml:SubjectConfirmation>
      </saml:Subject>
    </saml:AuthenticationStatement>
    <saml:AttributeStatement>
      <saml:Subject>
        <saml:NameIdentifier Format="#emailAddress" NameQualifier="samlauthority.com/">
          companyA@companyA.com
        </saml:NameIdentifier>
        <saml:SubjectConfirmation>
          <saml:ConfirmationMethod>
            urn:oasis:names:tc:SAML:1.0:am:password
          </saml:ConfirmationMethod>
          <saml:SubjectConfirmationData>password</saml:SubjectConfirmationData>
        </saml:SubjectConfirmation>
      </saml:Subject>
      <saml:Attribute AttributeName="//samlauthority.com/ebxml/regrep/role"
                      AttributeNamespace="samlauthority.com/namespace/">
        <saml:AttributeValue>ContentOwner</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
</samlp:Response>
```

Fig. 11 Response of Attribute and Authentication Assertion

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 Scenario 2, similar messages are generated according to the steps described in Fig. 4.

VI. Conclusion

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

Recently, many business systems have adopted Web Services standards that were proposed by W3C, and XML security technologies which are suitable security standards for Web Services. XML security technologies will become widely used as XML-based business applications become popular. We will further apply XML security technologies to real word business systems such as the Electronic Document Management Systems (EDMS) and the groupware systems. We will also continue research on the advanced security model using XML security.

References

[1] W3C Recommendation, *Extensible Markup Language (XML) 1.0 (Second Edition)*, W3C, T. Bray, J. Paoli, C.M. Sperberg-McQueen, E. Maler, 2000.
[2] Web Services Architecture Working Group Working Draft, *Web Services Architecture*, Web Services Architecture Working Group, D. Booth, H. Hass, F. McCabe, et. Al., 2003
[3] UN/CEFACT and OASIS Technical Specifications, *ebXML Technical Architecture Specification*, UN/CEFACT and OASIS, B., C. Barham, 2001.
[4] UN/CEFACT and OASIS Technical Reports, *ebXML Technical Architecture Risk Assessment V1.0*, *UN/CEFACT and OASIS*, ebXML Security Team, 2001.
[5] R. Conrad, D. Scheffner, and J. Freytag, "XML conceptual Modeling using UML", 19th International Conference on Conceptual Modeling, Salt Lake City, Utah, U.S.A., 2000.
[6] Transport Layer Security Working Group Internet Draft, *The SSL Protocol*, Transprot Layer Security Working Group, A.O. Freier, P. Karlton, P.C. Kocher, 1996
[7] IETF RFC. 2311, *S/MIME Version 2 Message Specification*, Network Working Group, 1998.
[8] ebXML, "Creating a Single Global Electronic Market," http://www.ebxml.org
[9] S. Patil, E. Newcomer, "ebXML and Web Services, Internet Computing", IEEE, Vol. 7, No. 3, May-June.2003, pp. 74-82.
[10] W3C Recommendation, *XML Encryption Syntax and Processing*, W3C, T. Imamura, B. Dillaway, E. Simon, 2002.
[11] W3C Recommendation, *XML Signature Syntax and Processing*, W3C, M. Bartel, J. Boyer, B. Fox, B. LaMacchia and E. Simon, 2002.
[12] W. Y. Han, C. S. Park, S. Y. lim, J. H. Kang, "An XML digital signature for Internet e-business applications", International Conferences on Info-tech and Info-net, Beijing China, Vol. 6, No. 29, Oct.2001, pp. 23-29.
[13] W3C Working Draft, XML Key Management Specification (XKMS) Version 2.0, W3C, W. Ford, P. Baker H., B. Fox, B. Dillaway, B. LaMacchia, J. Epstein and J. Lapp., 2003.
[14] OASIS Committee Specification, *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1*, OASIS, E. Maler, P. Mishra, R. Philpott R, 2003.
[15] OASIS Std., *eXtensible Access Control Markup Language (XACML) Version 1.0 OASIS Standard*, OASIS, S. Godik, T. Moses, 2003.
[16] E. Bertino, E. Ferrari, "Secure and selective dissemination of XML documents", ACM Transactions on Information and System Security (TISSEC), Vol. 5, No. 3, Aug.2002.
[17] E. Damiani, S. Vimercati, S. Paraboschi, P. Samarati, "A fine-grained access control system for XML documents", ACM Transactions on Information and System Security (TISSEC), Vol. 5, No. 2, May.2002.
[18] OASIS Technical Committee, Collaboration-Protocol Profile and Agreement Specification Version 2.0, OASIS, S. Aissi, A. Chan. *et. al.*, 2002.
[19] D. J. Polivy, R. Tamassia, "Authenticating Distributed Data using Web Services And XML Signatures", Dynamic Coalitions Program of the Defense Advanced Research Projects Agency under grant F30602–00–2–0509 (2002)
[20] P. Devanbu, M. Gertz, A. Kwong, C. Martel, G. Nuckolls, S. G. Stubblebine, "Flexible authentication of XML documents", ACM Conference on Computer and Communications Security, 2001, pp. 136-145.
[21] IETF RFC. 2459, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, Network Working Group, 1999.
[22] Junseok Lee, O.H. Sung, S.-W Jung, K. S. Yoon, C.S. Park and J.-C. Ryou, "A DRM Framework for Distributing Digital Contents through the Internet," ETRI Journal, vol. 25, no. 6, Dec 2003, pp. 423-436
[23] B. Pfitzmann, B. Waidner, "Token-based web Single Signon with Enabled Clients", IBM Research Report RZ 3458 (#93844), Nov.2002.
[24] J. Jeong, D. Shin, D. Shin, K. Moon., "Java-Based Single Sign-On Library Supporting SAML (Security Markup Language) for Distributed Web Services", Lecture Notes in Computer Science, Vol. 3007, 2004.