안전한 XML 웹 서비스를 위한 키 정보 서비스 프로토콜에 관한 연구

박남제[†]·문기영[††]·손승원[†††]

요 약

XML 보안 명세 중 하나인 XML 키 관리 명세(XKMS)는 다양하고 복잡한 기능의 웹 서비스 애플리케이션에서 XML 문서의 서명을 검증하거나 암호화하는 공개키의 관리를 위한 프로토콜을 정의한다. 본 논문에서는 XML표준 명세를 준수하는 XML 키 정보 프로토콜 서비스 모델을 제시하고, 표준에 근거한 프로토콜 컴포넌트의 참조 모델을 구현하였다. 또한, XML 기반 보안서비스 특성에 착안하여 안전한 XML 웹 서비스를 위한 키 정보 서비스에 대한 분석과 보안 방안에 대해 기술한다. 프로토콜 컴포넌트는 식별정보가 주어졌을 때, 필요로 하는 공개키 위치와 식별자 정보, 공개키 연결 기능을 제공한다. 구현된 참조 모델은 향후 국내 e-비즈니스 프레임워크 구성 시 표준적인 보안 모델을 구현할 수 있는 지침을 제공할 것이다.

A Study on Key Information Service Protocol for Secure XML Web Service

Nam Je Park[†]·Young Ki Moon[††]·Sung Won Sohn[†††]

ABSTRACT

XKMS(XML Key Management Specification), one of XML Security specification, defines the protocol for distributing and registering public keys for verifying digital signatures and enciphering XML documents of web service applications with various and complicate functions. In this paper, we propose XML Key Information protocol service model and implement reference model of protocol component based on standard specification. Also describes the analysis and security method of Key Information Service(XKIS) for Secure XML Web Service, paying attention to the features of XML based security service. This protocol component supported includes public key location by given identifier information, the binding of such keys to identifier information. This reference model offers the security construction guideline for future domestic e-Business Frameworks.

키워드 : XML 키 관리(XML Key Management), XML 보안(XML Security), XKMS, XKISS, XML 웹 서비스(XML Web Service)

1. Introduction

The XML(eXtensible Markup Language) is a promising standard for describing semi-structured information and contents on the Internet. Some of the well-recognized benefits of using XML as data container are its simplicity, richness of the data structure, and excellent handling of international characters. The practical use of XML is increasing in proportion to spread speed of XML Web Service as global standard for Internet and Web Service. In this environment, a security mechanism for XML documents must be provided in the first place for secure XML Web Service. The security mechanism also has to support security function for the existing non-XML documents, too.

The XML Security standards define XML vocabularies and processing rules in order to meet security requirements. These standards use legacy cryptographic and security technologies, as well as emerging XML technologies, to provide a flexible, extensible and practical solution toward meeting security requirements.

The Industry is therefore eager for XML and PKI(Public Key Infrastructure) to work together in fulfilling the widely held expectations for cryptographically secure, XML-coupled business applications. The best-known simplicity of XML is to provide portability of data between disparate

[†] 정 회 원 : 한국전자통신연구원 정보보호연구본부 능동보안기술연구팀
[††] 정 회 원 : 한국전자통신연구원 정보보호연구본부 능동보안기술연구팀 과제책임/선임연구원
[†††] 정 회 원 : 한국전자통신연구원 네트워크보안연구부장/책임연구원

논문접수 : 2003년 7월 22일, 심사완료 : 2003년 9월 22일

business systems contrasts with the complexity of traditional PKI implementation. Therefore, a key architectural goal in the XML Key Management Specification(XKMS) is to shield XML application developers from the complexity of traditional PKI implementation. It permits delegation of trust processing decisions to one or more specialized trust processors. It enables XML-based systems to rely on complex trust relationships without the need for complex or specialized end-entity PKI application logic on the client platforms where XML processing is taking place.

The world recently, by way to offer certification about important transaction of this XML environment, is researching about XML key management to integration of PKI and public key certificate and XML application. At the same time Setting a reference systems that embody this are developed. But, R&D for actually system that domestic can construct XKMS offer of Trust Service based on XML are insufficient. Therefore, further research activity is needed for the progress and standardization of the XML key management technology, and it is necessary to develop XML key management system for the activation of the Secure XML Web Service.

E-XKISS(ETRI XKIS System) which will be introduced in this paper, is a subsystem of XKMS that has been implemented to support the processing, by a relying party, of Key Information associated with a XML digital signature, XML encrypted data, or other public key usage in an XML web application.

In this paper, we propose a design for XML Key Information Service(XKIS) Model and we explain our implementation, service protocol component based on standard specification. First we investigate related work on XKMS and then we explain overview of the service system structure. Then we propose a design for service model and explain implemented service protocol component. Finally, we explain function of protocol component and then we conclude this paper.

2. Related Work

2.1 Standardization Activities regarding XKMS

To simplify the integration of PKI and digital certificates with XML applications, an industry consortium of VeriSign, Microsoft, Ariba and webMethods have created the open XKMS[2, 28, 31]. XKMS 1.0 was submitted to the W3C (World Wide Web Consortium) as a technical note in March 2001 and a working group formed to develop a standard. And it's getting more and more support from the industry. Later XKMS efforts were joined by Citigroup, HP, IBM, IONA, Netegrity, Entrust, Baltimore Technologies, Reuters and more. Although a number of vendors released products and prototypes based on the 1.0 specification, a number of minor variations were made during interoperability testing[1, 2].

The W3C has announced the launch of its XML Key Management Activity, tasked with the development of an XML application/protocol that allows a simple client to obtain key information(values, certificates, management or trust data) from a Web Service. Based upon the XKMS, the Activity is chartered to produce a companion Recommendation for the IETF/W3C XML Encryption and XML Signature Activities.

2.2 Products related to XKMS

Verisign and Microsoft, Entrust, Baltimore[4], RSA Security has its own XKMS reference solutions. Verisign is one of the original authors of XKMS. Microsoft maintains client and server sample code(ASP.NET) at Internet web site. Entrust maintains a java XKMS reference implementation as a service on the Internet web site. RSA Security's BSAFE Cert-J SDK supports XML-DSIG and XKMS.

3. The Structure of the E-XKIS Service Model

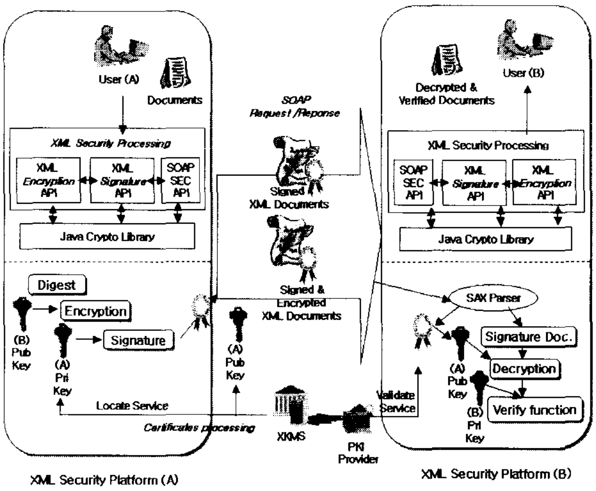
In this section, the structure of XKIS platform based on XML Security and XML Key Information Protocol Service Model will be introduced.

3.1 XML Security Platform

As is mentioned above, XML Security Platform has XML Signature API, XML Encryption API and Java Crypto Library as its subsystem. Java Crypto Library is compatible with Sun JCE(Java Cryptography Extension). XML Security Platform also includes XKMS Client service component for certificate processing and retrieve private keys and certificates for users. The following (Figure 1) shows the structure of XML Security Platform[33, 34].

XML Security Platform processes input document to make them secure and it is composed of XML Signature API's and XML Encryption API's.

XML Signature API provides digital signature generation and verification that is in the form of XML document and XML Encryption API encrypts and decrypts the e-documents including XML documents[5, 6]. The encrypted document is also in the form of XML. Encryption API provides platform independent Java Crypto Library and they are called by XML Security Platform Subsystem for digital signature or encryption[7, 8]. XML Security Platform uses X.509 certificates that are issued by certificate Authority for digital signature.



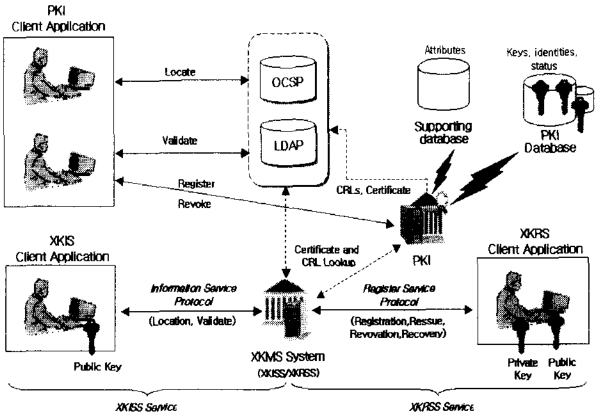
(Figure 1) The Structure of XML Security Platform

When XML Security Platform subsystem processes e-document, XML Signature API signs it and it is sent to the destination system. XML Security Platform subsystem of the destination system verifies the signature using XML Signature API's. In this case, authentication, integrity and confidentiality of the document are guaranteed.

3.2 E-XKIS Service Model

XKMS defines protocols for the registration and distribution of public keys[1-3]. The keys may be used with XML Signatures, a future XML Encryption specification, or other public key applications for secure messaging.

XKMS system is comprised of the XKISS and the XKRSS. XKISS allows a client application to delegate part or all of the tasks required to process an XML Signature to a trust service. This is useful for developers who don't want to implement the signature checking them, or who want to delegate this functionality to an application service provider that may be optimized for signature checking. XKRSS is an XML-based replacement for existing PKI file formats that are used when a user applies for a digital certificate[10, 11]. XML brings the same advantages to PKI as it brings to other industries-open standards, platform independence, and human readability. Both protocols utilize SOAP(Simple Object Access Protocol), and WSDL(Web Services Definition Language) is used to define message relationships. The XKRSS and XKISS protocols are expressed using the W3C's XML Schema Language[1]. (Figure 2) shows XKISS service model include XKRSS service of W3C.



(Figure 2) Architecture of E-XKIS Service Model

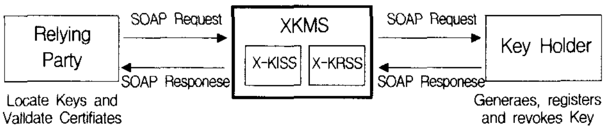
As shown in the figure, a key owner registers his key with an XKMS service provider who makes use of an underlying PKI to store and bind the keys with identification information. A commercial PKI typically contains a key registration authority, a certification authority, a key validation authority, and a secure keys directory in which all information is stored[12-14]. Any Web service that wants to validate a <ds:KeyInfo> element it has received can invoke an XKISS service that once again makes use of the underlying PKI to complete the process[29, 32].

4. Design of E-XKIS Service Protocol Component

In this section we explain our design of XKIS service protocol component.

4.1 Overview of XKMS Protocol in W3C

The XKMS protocol is essentially a request response protocol layers on SOAP, with optional embellishments described at the end of the chapter[1-3].



(Figure 3) XKMS Protocol

The request and result messages used in the individual XKMS operations share a common format. These common members are defines in <Table 1> [32].

〈Table 1〉 Members Common to Request and Result Elements

Item	Description
Id@	A Unique identifier for the message
Service@	The service URI of the XKMS service
Nonce@	Randomly generated information that is used in the extended protocol processing options to defeat replay and denial of service attacks
ds : Signature	An enveloped XML Signature that authenticates the XKMS messages
OpaqueClient Data	Optional information supplied by the client in a request that is returned unmodified in the response

Additional members are defined for request messages, allows the client to specify the protocol options it supports, the types of and maximum quantity of information to be provided in the response, and additional information used in the extended protocol options. These additional members are described in <Table 2>[32]. Additional members are defined for request messages, allowing the service to specify the result of the operation(success, failure, etc) and binding the request to the response by means of the request Id. These additional members are described in <Table 2>.

4.2 Analysis of XKIS Protocol

One of the major service of XKMS is XKISS defines protocols to support the processing by a relying party of key information associated with a XML digital signature, XML encrypted data, or other public key usage in an XML aware application[2]. Functions supported include locating required public keys given identifier information, and binding of such keys to identifier information.

XKISS defines three levels of key information service that is Retrieval Method, Locate Service, and Validate Service. It mentions the possibility of higher-level services, such as those dealing with long term trust relationships or the status of trust assertions. <Table 3> shows Tiered service mode of W3C spec[2].

〈Table 3〉 Tiered Service Mode

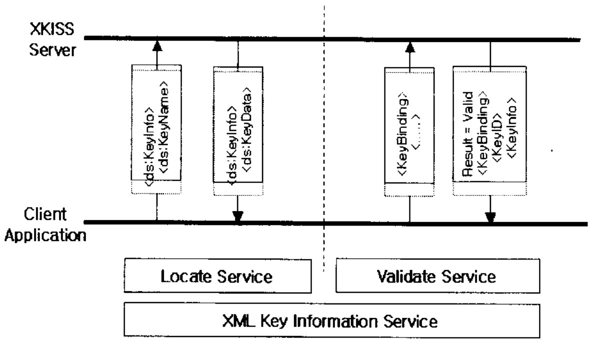
Level	Service Name	Comments	
Tier 0	-	Client performs the location and validation itself	M
Tier 1	Location	Client delegates location to assertion server, but performs validation itself	M
Tier 2	Validate	Client delegates both location and validation to the assertion server	M
Tier 3	Assertion	Establishment and management of long term trust relationships	O
Tier 4	Assertion Status	Management of the status of assertions	O

(M : Mandatory, O : Optional)

The following (Figure 4) shows the Locate Service protocol. A client receives a signed XML document. The <Key-Info> element in the signature specifies a retrieval method for an X.509 certificate. The client lacking the means to either resolve the URL or parse the X.509 certificate to obtain the public key parameters delegates these tasks to the trust service. The following (Figure 4) shows the Validate Service protocol. The client sends to the XKMS service a prototype containing some or all of the elements for which the status of the key binding is required. If the information in the prototype is incomplete, the XKMS service may obtain additional data required from an underlying PKI service. Once the validity of the key binding has been determined the XKMS service returns the status result to the client.

〈Table 2〉 Members of the Request & Response Element

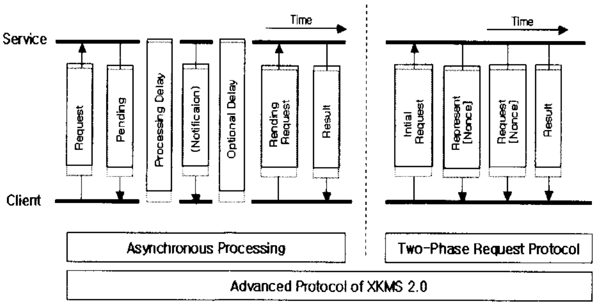
	Item	DescriptionItem
Request Element	ResponseMechanism	Specifies ant extended protocol options supported by the client for this request, such as asynchronous processing or the two-phase protocol. Multiple ResponseMechanism values may be specified
	ResponseWith	Specifies a data type that the client requests be present in the response, such as a key value, an X.509 certificate, or a certificate chain. Multiple ResponseWith values may be specified
	PendingNotification	Optionally specifies a means of notifying completion of the operation when asynchronous processing is used
	OriginalRequestID@	This attribute is used in the extended protocol to specify the ID attribute of the initial request in a multistage request
	ResponseLimit	The maximum number of key binding elements that the service should return in a response
Response Element	ResultMajor	The principal result code of the XKMS operation
	ResultMinor	The secondary result code of the XKMS operation, giving additional information such as reason for the result
	RequestID	The ID attribute of the corresponding request


XKISS Server
Client Application
<ds:KeyInfo> <ds:KeyName>
<ds:KeyInfo> <ds:KeyData>
<KeyBinding> <......>
Result = Valid <KeyBinding> <KeyID> <KeyInfo>
Locate Service
Validate Service
XML Key Information Service

(Figure 4) Protocol of Locate and Validate Service

In XKMS 1.1, all operations consisted of a single request message followed by a single response. XKMS 2.0 specifies additional protocol options that allow a client to make multiple XKMS requests simultaneously, allow an XKMS service to queue XKMS requests for later processing, and make it possible to defend against denial of service attacks [32].

First, Asynchronous processing may be required because some form of operator intervention is required to complete an operation. Asynchronous processing is also desirable in cases where the request may take a long time to complete. Asynchronous processing involves two separate request/response pairs. Second, Two Phase Request Protocol providers protection against denial of service attacks by checking that the requestor can read IP packets sent to the purported source of the request.



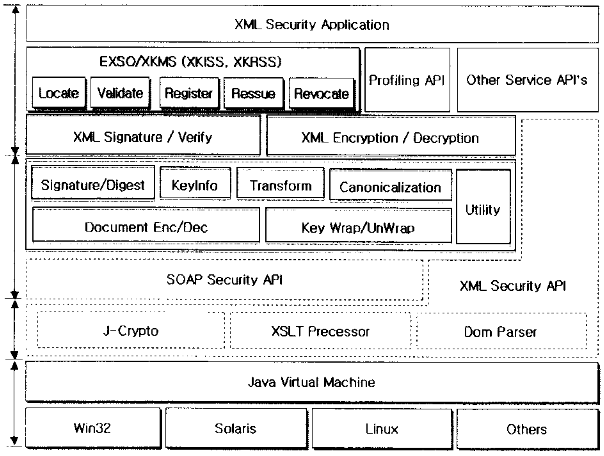
(Figure 5) Advanced Protocol features of XKMS 2.0

The following (Figure 5) shows the Asynchronous processing. The Client makes the first request specifying the Response mechanism type <xkms:Asynchronous>. The service may return the actual response immediately or signal that the response will be returned asynchronously using the ResultMajor code <xkms:Pending>. Once the service has completed processing the request, the client obtains the result by issuing a pending request message[32]. The following (Figure 5) shows two-phase request protocol. The client sends an initial request to the service. Unless the service has reason to believe that the request is part of a denial of service attack, the service may respond with an immediate result. If the service has determined that it is under a denial of service attack and the request may be a part of that attack, it returns response with the ResultMajor code <xkms : Represent> that contains a nonce value. In order for the service to act on the request, the client must represent the request together with the previously issued none value.

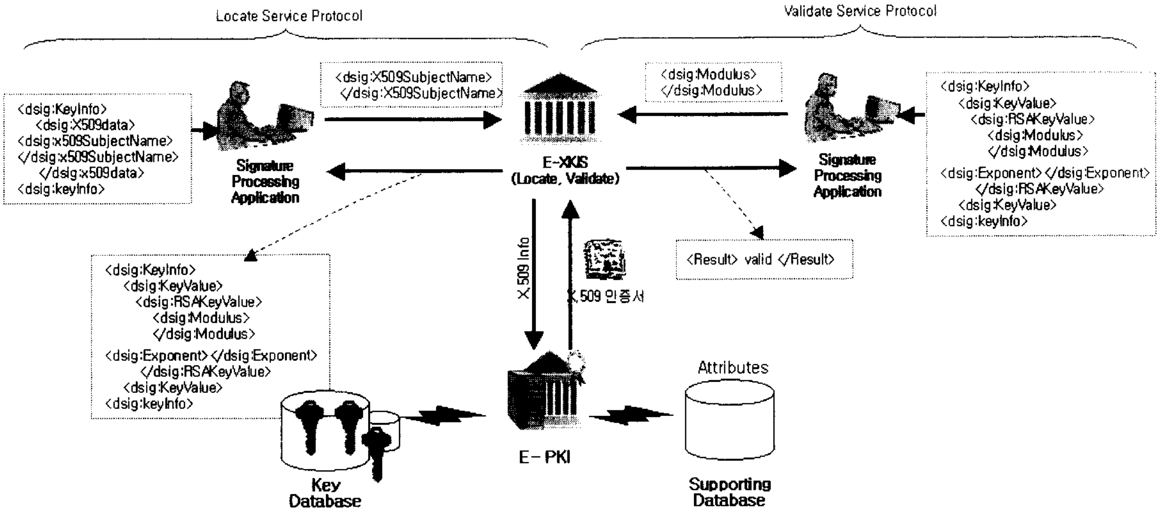
4.3 Design of E-XKIS Service Platform

In case tools that is based on Java these advantage that can bring much gain economically, without porting process between kinds of machine. When develop program of multiplex platform environment. Specially, When develop client/server program. These can use same module, just as, it is in spite of that development environment can be different.

XKIS Service Platform is a Framework for the approaches about function of XKMS System and work for development based on Java platform. XML Security API is expressed by structure of Java Crypto Library and XML Paser, XSLT processor. And It includes service provide mechanism. SOAP security API supplies XML Web service security. And XML security API and SOAP security API supports key exchange and encryption. It supports XML Signature and XML Encryption function. Based on this, XKISS service platform is composed. So, XKISS service application program are achieved by component of service platform that is constructed by each function. Other than system application, Many XML web application security can be provided using the XML security API and Library that is provided from the XKIS Service Platform.



(Figure 6) Architecture of E-XKIS Service Platform



(Figure 7) Locate and Validate Service Model of E-XKISS

(Figure 6) illustrates the architecture of E-XKIS(ETRI XKIS) Service Platform. Major components of XKISS Service Platform are Java Crypto Library, XML Security API, SOAP Security API, XML Signature API, XML Encryption API.

4.4 Processing Flow of E-XKIS Service Protocol Component

XKISS supports two services. Locate Service resolves a <ds:Keyinfo> element but does not require the service to make an assertion concerning the validity of the binding between the data in the <Keyinfo> element.

The Validate Service provides all the functions of locate but returns a trusted key binding that has been validated in accordance with the policy of validate service.

Locate Service retrieves and provides information concerning keys. In Locate Service of (Figure 7) begins with an incoming XML Signature. The <ds:Signature> element is parsed for the <ds:KeyInfo> element that contains a <ds:KeyName> element including the odd key identifier. We are assuming the signature processing application doesn't understand this identifier and must delegate the processing to a key location service. This key locate service processes the key identifier and makes a database query that matches it to an X.509 certificate. This certificate is then formatted as a <ds:KeyInfo> element and passed back to the signature processing application. At this point the signature processing application has enough information to perform cryptographic validation of the signature processing application. At this point the signature processing application has enough in formation to perform cryptographic validation of the signature. It now has a public key, whereas before it only had a single key identifier. The signature processing application may now choose to perform path validation on its own, or it may decide to delegate this action to a service as well. The key location service is the first tier of XKISS, which is called the locate service. In addition to passing off <ds:KeyInfo> element, the signature processing application may also pass off a <ds:RetrievalMethod> element if the signature processing application doesn't have access to the necessary network or server location.

The second tier is called the Validate Service and is responsible for asserting trust over the binding of a name and a public key. The Validate Service is a superset of the Locate Service. This means that in addition to providing name key assertions, it can also locate public key values. In Validate Service of (Figure 7) we have a situation similar to the one presented in Locate Service of (Figure 7). In Validate Service of (Figure 6) we are passing a <ds:X509Data> element to the Validate Service with the expectation of a status result and an indication of the key binding. Validate Service gives us the name and public key from the queried certificate as well as make en assertion regarding the binding between the name in the certificate and the public key.

(Figure 8) illustrates the processing flow of XML Signature and XML Encryption in XKISS[29, 35, 36].

Processing Flow of Signature key information service is as follows.

① Sender security registers his or her public signing key with an XKISS (or XKMS)

② Signed message is sent to the receiver.

③ Receiver retrieves the public key and verifies the signature

④ In order cases, the trust service may obtain signing key information from other XKMS Service or other types of servers.

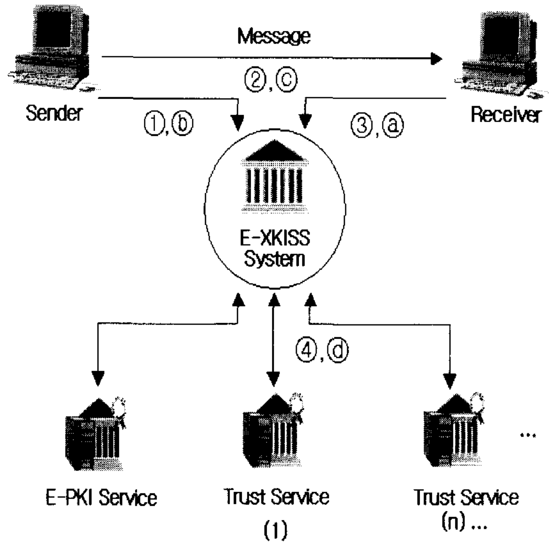
Processing Flow of Encryption key information service is as follows.

ⓐ Receiver security registers his or her encryption public key with an XKISS (or XKMS)

ⓑ Sender retrieves the public key and encrypts the message to the receiver.

ⓒ Receiver receives and decrypts the message retrieves the public key and verifies the signature

ⓓ In order cases, the trust service may obtain encryption key information from other XKMS Service or other types of servers.

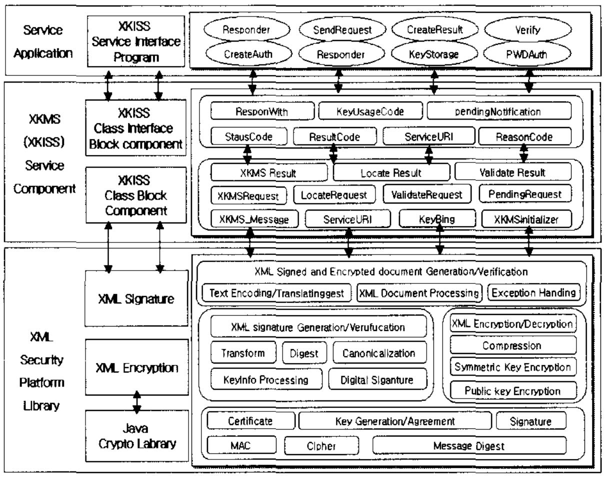


(Figure 8) Signature and Encryption service flow of XKISS

5. Implementation of E-XKIS Service Protocol Component

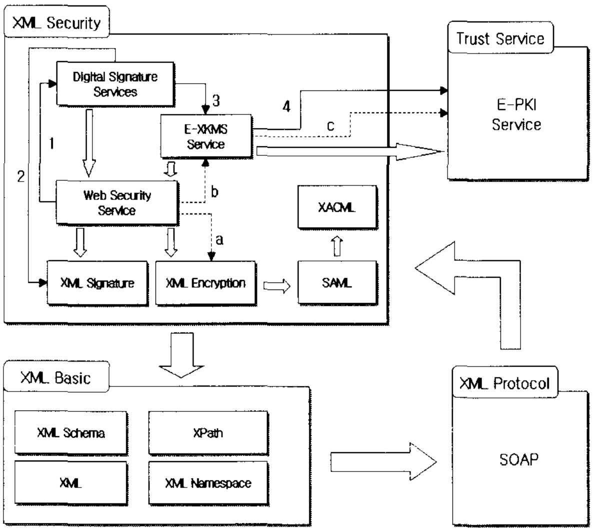
5.1 Implementation of protocol component

XKISS has been implemented based on the design described in previous section. Package library architecture of XKISS based on CAPI(Cryptographic Application Programming Interface) is illustrated in (Figure 9). Components of the XKISS are XML Security Platform library, Service components API, Application program. Although XKIS service protocol component is intended to support XML applications, it can also be used in order environments where the same management and deployment benefits are achievable. XKISS has been implemented in Java and it runs on JDK ver 1.3 or more.



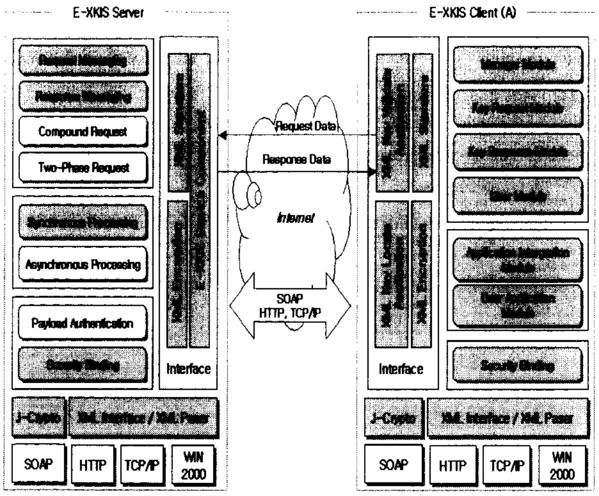
(Figure 9) Package library architecture of E-XKISS based on CAPI

The manner in which the various XKIS service builds upon each other and consumes each other's services is shown in the following diagram.



(Figure 10) Interrelation of E-XKMS(E-XKIS) service

The Arrow reflects the primary relationship between the security services. The First(Alphabet) path shows alternative ways of checking the security of a SOAP message secured using Web Security Service. Second(Number) path is the same except Web Service Security delegates signature checking in its entirety to a Digital Signature Service. The figure for representing Testbed Architecture of XKIS service protocol component is as follows (Figure 11).



(Figure 11) Testbed Architecture of E-XKIS Service Protocol Component

We use Testbed system of windows PC environment to simulate the processing of various service protocols. The protocols have been tested on Pentium 3 and Pentium 4 PCs. It has been tested on Windows 2000 server, Windows XP. The E-XKIS server is composed Server service component of XKIS platform package. The communication protocol between the Server and Client follows the standardized SOAP protocol illustrated in (figure 11). And the message format is based on specification of W3C. <Table 4> summarizes function of XKIS service protocol component.

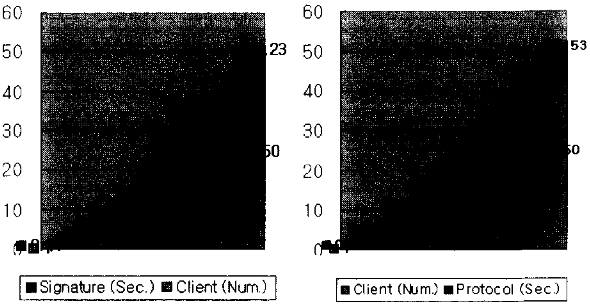
<Table 4> Function of E-XKIS service protocol component

Service & Protocol	E-XKIS					
	Tier 0	Tier 1	Tier 2	Tier 3	Tier 4	
Register Service	*	*	*	*	*	KRSS
Locate Service	M	M	*	O	O	-
Validate Service	M	M	M	O	O	-
Recovery/ Revoke Service	*	*	*	*	*	KRSS
Compound Request Protocol	O	O	O			-
Synchronous Processing	*	M	M	*	*	-
Asynchronous Processing	*	O	O	*	*	-
Two-Phase Request Protocol	*	O	O	*	*	-
Payload Authentication	*	O	O	*	*	-
HTTP Transport	M	M	M	M	M	-
SOAP 1.1 Transport	M	M	M	M	M	-

(M : Mandatory, O : Optional, * : No Recommendation)

5.2 Performance Evaluation

(Figure 12) (a) showed difference for 0.2 seconds that compare average transfer time between client and server of XML Encryption&Decryption by XML Signature base on XML Security Platform. According as increase client number on the whole, showed phenomenon that increase until 0.3 seconds



(a)

(b)

(Figure 12) Performance Evaluation

(Figure 12) (b) is change of average transmission time according as increase client number in whole protocol environment. If client number increases, we can see that average transfer time increases on the whole. And average transfer time increases rapidly in case of client number is more than 45. Therefore, client number that can process stably in computer on Testbed environment grasped about 40. When compare difference of (Figure 12) (a) and (Figure 12) (b). Time of XML Signature module is occupying and shows the importance of signature module about 60% of whole protocol time.

6. Conclusion

In this paper, we have proposed the Key Information Service Protocol Model based on XML(E-XKIS Model) for secure XML Web Service. And we designed a Security Platform based on XML(XML Security Platform) that provides security services such as authentication, integrity and confidentiality for XML Web Service. It provides XML Signature function, XML Encryption function, Java Crypto Library for securing XML document that are exchanged in the XML Web Service. And then we implemented service protocol component of XKISS(E-XKIS System) based on X-KISS standard specification of W3C. It provides function of Locate and Validate Service based on service protocol. XKIS Platform of this paper can be applied to various services that require secure exchange of e-document such as B2C, B2B and XML/EDI. Since it is developed in Java, it can be ported easily to various platforms. And Since XML Signature, XML Encryption, Java Crypto Library is conforming to international standards, XKIS Platform is compatible with many security platforms that conform to the standards.

Further research and development are needed on the integration between two system that XKISS and PKI System. And Need continuous research for integration of XML Signature&Encryption technical development in mobile platform and XKISS based on wire/wireless system for XML Web Service of next generation Web business environment.

References

[1] W3C Note, "XML Key Management(XKMS 2.0) Requirements," May, 2003.

[2] W3C Working Draft, "XML Key Management Specification Version 2.0," April, 2003.

[3] W3C Working Draft, "XML Key Management Specification Bindings," April, 2003.

[4] W3C Working Draft, "XKMS Bulk Operation," August, 2002.

[5] W3C/IETF Draft, "XML-Signature Requirements," October, 1999.

[6] W3C/IETF Recommendation, "XML-Signature Syntax and Processing," Feburary, 2002.

[7] W3C Recommendation, "XML Encryption Syntax and Processing," 2003.

[8] W3C Recommendation, "Decryption Transformation for XML Signature," 2003.

[9] IETF, "X.509 Certificate and CRL Profile," RFC2459, Januanry, 1999.

[10] IETF, "Certificate Management Protocol, RFC2510," March, 1999.

[11] IETF, "Certificate Request Message Format," RFC2511, March, 1999.[12] RSA Encryption Standard, PKCS #1.

[13] "Password-Based Encryption Standard," PKCS #5.

[14] "Public-Key Cryptography Standard," PKCS #7.

[15] "ASN.1 Specification of Basic Notation," ITU-T X.680.

[16] "ASN.1 Encoding Rules DER," ITU-T X.690.

[17] W3C, "XML 1.0 Recommendation," Feburary, 1998.

[18] W3C, "Document Object Model (DOM) Level 1 Specification," October, 1998.

[19] W3C Working Draft, "SOAP Version 1.2 (1) : Messaging Framework," June, 2002.

[20] W3C Note, "SOAP : Simple Object Access Protocol 1.1," May, 2000.

[23] W3C Note, "SOAP Security Extensions : Digital Signature," Feb., 2001.

[24] IETF, "The TLS Protocol Version, 1.0," RFC 2246, January, 1999.

[25] NIST, "Key Management Guideline, Part 1 : General Guideline," 2002.

[26] Mark Bartel, Bard Fox, Brian LaMacchia and Ed Simon, "XML Signature Syntax and Pro cessing," http://www.w3.org/TR/xmldsig-core/.

[27] Takeshi Imamura, Blair Dillaway and Ed Simon, "XML Encryption Syntax and processing," http://www.w3.rg/TR/xmlenccore/, 2002.

[28] Phillip Hallam-Baker, "W3C XKMS Workshop position paper," Proceedings of XKMS Workshop, Redwood City, CA, July, 2001.

[29] Blake Dournaee, "XML Security," RSA Press, 2002.

[30] Donald E, Eastlake, Kitty Niles, "Secure XML, Pearson addison wesley," 2003.

[31] OASIS, "Web Service Security," http://www-106.ibm.com/, Apr, 2002.

[32] Mark ONeill, et al., "Web Service Security," Osborne, 2003.

[33] Jae Seung Lee, Young Soo Kim, Joo young Lee, Ju Han Kim, KyungBum Kim and Seung Won Sohn, "A Design of the XML Security Platform for Secure Electronic Commerce," WISA 2000, Seoul, Korea, 2000.

[34] Joo Young Lee, Ju Han Kim, Jae Seung Lee, Ki Young Moon, and Hyun-Sook Cho, "ESES : XML Security for Secure Electronic Commerce," Proceedings of WISA 2001, Sep. 2001.

[35] Nam Je Park et. Al., "XML Key Management of Secure Electronic Trading," KIISC Review, ISSN 1598-3978, 13 (3), June, 2003.

[36] Nam Je Park, Ki Young Moon, "EXSO/XKMS Service Platform Infrastructure," CISC 2003, pp.212-216, 2003.

박 남 제

e-mail : namjepark@etri.re.kr

2000년 동국대학교 정보산업학과(학사)

2003년 성균관대학교 정보보호학과(공학석사)

2003년~현재 한국전자통신연구원 정보보호연구본부 능동보안기술연구팀

관심분야 : XML 정보보호, EC 보안, 무선인터넷 보안, 전자지불, 컨텐츠 보호 등

문 기 영

e-mail : kymoon@etri.re.kr

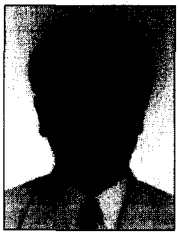
1986년 경북대학교 전자공학과(학사)

1989년 경북대학교 대학원 전자공학과
(공학석사)

1992년~1994년 (주)대우정보시스템 기술
연구소

1994년~현재 한국전자통신연구원 정보보호연구본부 능동보안
기술연구팀 과제책임/선임연구원

관심분야 : XML 정보보호, 응용보안, 분산시스템, 트랜잭션 등



손 승 원

e-mail : swsohn@etri.re.kr

1984년 경북대학교 전자공학과(학사)

1994년 연세대학교 대학원 전자공학과
(공학석사)

1999년 충북대학교 대학원 컴퓨터공학과
(공학박사)

1991년~현재 한국전자통신연구원 네트워크보안연구부장/책임
연구원

관심분야 : 네트워크보안, 라우팅 알고리즘, 생체인식기술 등