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 e-documents of electronic trading applications with various and complicate functions. In this paper, we propose XKIS(XML Key Information Service) Model and implement E-XKISS(ETRI XKIS System) Component based on standard specification. Also describes the analysis and security of XKIS for Secure e-Trading, paying attention to the features of XML based security service. This model supported includes public key location by given identifier information, the binding of such keys to identifier information. The client service component controls the numbers of locate threads and validate threads to analyze the minimum requirements of real-time key retrievals. This reference model offers the security construction guideline for future domestic e-Business Frameworks.

Key-Words: XKMS. XKKISS, XML Security, XML. Key Management, Electronic Trading, PKI

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 e-Trading(Electronic Trading) as global standard for Internet and e-Trading. In this environment, a security mechanism for XML documents must be provided in the first place for secure e-Trading. 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 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. 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 XKIS(XML Key Information Service) Model and we explain our implementation, E-XKISS service 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 XKIS Service Model and explain implemented E-KISS service component. Finally, we explain function of service component and then we conclude this paper.
2 Related Work
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. 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.

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 Digital Signature Activities.

And, Verisign and Microsoft, Entrust, Baltimore, 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-XKISS Platform
In this section, the structure of ETRI (Electronics and Telecommunications Research Institute) XML Security Platform and E-XKISS Service Model will be introduced.

3.1 ETRI XML Security Platform
As is mentioned above, ETRI 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). ETRI 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 ETRI XML Security Platform.

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

[Figure 1] The Structure of ETRI XML Security Platform
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. 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. XML Security Platform uses X.509 certificates that are issued by certificate Authority for digital signature.

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

ETRI-XKMS System is comprised of the E-XKISS and the E-XKRSS (ETRI XKRSS System). E-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. E-XKRSS is an XML-based replacement for existing PKI file formats that are used when a user applies for a digital certificate. 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 E-XKRSS and E-XKISS protocols are expressed using the W3C's XML Schema Language. Figure 2 shows E-XKISS Service Model include X-KRSS Service of W3C.

As shown in the figure, a key owner registers his key with an ETRI-XKMS Service provider who makes use of an underlying ETRI-PKI to store and bind the keys with identification information. A commercial ETRI-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. Any Web service that wants to validate a <ds:KeyInfo> element it has received can invoke an E-XKISS Service that once again makes use of the underlying ETRI-PKI to complete the process.

4 Design of E-XKISS Component
In this section we explain our design of E-XKISS service.

4.1 Overview of XKISS Protocol
One of the major service of XKMS is X-KISS 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. 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.

The following [Figure 3] shows the Locate Service protocol. A client receives a signed XML document. The <KeyInfo> 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 3] 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.

4.2 Architecture of E-XKISS 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.

E-XKISS Service Platform is a Framework for the approaches about function of ETRI-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, E-XKISS Service Platform is composed. So, E-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 E-XKISS Service Platform.

[Figure 5] illustrates the architecture of E-XKISS Service Platform. Major components of E-XKISS Service Platform are Java Crypto Library, XML Security API, SOAP Security API, XML Signature API, XML Encryption API.

4.3 Processing Flow E-XKISS Component
E-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 <ds: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 6] 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 information 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 E-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.

[Figure 6] Locate and Validate service of E-XKISS

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 6] we have a situation similar to the one presented in Locate service of [Figure 6]. 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 an assertion regarding the binding between the name in the certificate and the public key.

[Figure 7] illustrates the processing flow of XML Signature and XML Encryption in E-XKISS
Processing Flow of Signature key information service is as follows.

1. Sender security registers his or her public signing key with an E-XKISS(or ETRI-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.

[Figure 7] Sig.&Enc. Service flow of E-XKISS

Processing Flow of Encryption key information service is as follows.
- a. Receiver security registers his or her encryption public key with an E-XKISS (or ETRI-XKMS).
- b. Sender retrieves the public key and encrypts the message to the receiver.
- c. Receiver receives and decrypts the message.
- d. In order cases, the trust service may obtain encryption key information from other XKMS Service or other types of servers.

5 Implementation of E-XKISS Component

E-XKISS has been implemented based on the design described in previous section.

Package Library Architecture of E-XKISS based on CAPI (Cryptographic Application Programming Interface) is illustrated in [Figure 8]. Components of the E-XKISS are XML Security Platform Library, Service components API, Application Program. Although E-XKISS Service Component is intended to support XML applications, it can also be used in order environments where the same management and deployment benefits are achievable. E-XKISS has been implemented in Java and it runs on JDK ver 1.3 or more.

The manner in which the various E-XKISS Service builds upon each other and consumes each other’s services is shown in the following diagram.[Figure 9]

[Figure 8] Package Library Architecture of E-XKISS based on CAPI

[Figure 9] Interrelation of 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.

[Figure 10] shows Package structure of E-XKISS Service Component.

[Figure 10] Package structure of E-XKISS Component
The exkms.exkiss.xkis.message package is the implementation of classes for E-XKIS messages. The exkms.exkiss.xkis.request package is the implementation of classes for E-XKIS request message. The exkms.exkiss.xkis.result package is the implementation of classes for E-XKIS result message. The exkms.exkiss.interface.xkiscode package is the implementation of classes for the E-XKIS Status, Reason, Result message element.

The figure for representing Testbed Architecture of E-XKISS Service Component is as follows [Figure 11].

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 E-XKISS 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 2] summarizes algorithms supported by E-XKISS. E-XKISS provides all the algorithms recommended by XML Signature and XML Encryption of W3C Spec.

6 Conclusion
In this paper, we have proposed the Key Information Service model based on XML(ETRI-XKIS) for secure e-Trading. And we designed a Security Platform based on XML(ETRI XML Security Platform) that provides security services such as authentication, integrity and confidentiality for e-Trading. It provides XML Signature function, XML Encryption function, Java Crypto Library for securing XML document that are exchanged in the e-Trading. And then we implemented Service Component of E-XKISS(ETRI XKIS System) based on X-KISS Standard Specification of W3C. It provides function of Locate and Validate Service based on service protocol.

<table>
<thead>
<tr>
<th>Table 2. Algorithms Supported by E-XKISS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XML Signature</strong></td>
</tr>
<tr>
<td>Digest</td>
</tr>
<tr>
<td>Encoding</td>
</tr>
<tr>
<td>MAC</td>
</tr>
<tr>
<td><strong>Message Digest</strong></td>
</tr>
<tr>
<td><strong>Message Authentication</strong></td>
</tr>
<tr>
<td><strong>Canonicalization</strong></td>
</tr>
<tr>
<td><strong>XML Encryption</strong></td>
</tr>
<tr>
<td>Block Encryption</td>
</tr>
<tr>
<td>Enveloped Signature</td>
</tr>
<tr>
<td>3-DES</td>
</tr>
<tr>
<td>AES-128</td>
</tr>
<tr>
<td>AES-256</td>
</tr>
<tr>
<td>AES-192</td>
</tr>
<tr>
<td>RSA-OAEP</td>
</tr>
<tr>
<td>RSAOAEP-1.5</td>
</tr>
<tr>
<td>Diffie-Hellman</td>
</tr>
<tr>
<td>3DES KeyWrap</td>
</tr>
<tr>
<td>AES-128 KeyWrap</td>
</tr>
<tr>
<td>AES-256 KeyWrap</td>
</tr>
<tr>
<td>AES-192 KeyWrap</td>
</tr>
<tr>
<td><strong>Key Agreement</strong></td>
</tr>
<tr>
<td>Symmetric key Wrap</td>
</tr>
<tr>
<td><strong>XML Digital Signature</strong></td>
</tr>
</tbody>
</table>

References: