Journal of Southeast University (English Edition) June 2002 Vol.18 No.2 ISSN 1003—7985

CIT/XML Security Platform Syntax and Processing*

AL-Helali Adnan Hadi Mahdi** Zhang Shensheng

(Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: Today companies and organizations are using the Web as the main information dissemination means both at internal and external level. Information dissemination often takes the form of XML documents that are made available at Web servers, or that are actively broadcasted by Web servers to interested clients. These documents often contain information at different degrees of sensitivity, therefore a strong XML security platform and mechanism is needed. In this paper we developed CIT/XML security platform and take a close look to syntax and processing of CIT/digital signature model, CIT/encryption model, CIT/ smart card crypto and SPKI interface security models. Security services such as authentication, integrity and confidentiality to XML documents and non-XML documents, which exchanged among various servers, are provided.

Key words: electronic commerce, security, digital certificates, smart card, digital commerce, authentication, SPKI, XML

Today companies and organizations are using the Web as the main information dissemination means both at internal and external level. Information dissemination often takes the form of documents that are made available at Web servers, or that are actively broadcast by Web servers to interested clients. Such a widespread use of the Web has pushed the rapid development of suitable standards for information representation[1].

The extensible markup language (XML)[2,3] is a practical subset of standard generalized markup language (SGML) and it inherits the extensibility, structure and validation features of the SGML. It is the standard for the data exchange in Internet and widely accepted as the standard for electronic documents.

Therefore, a security models and mechanisms for XML documents must be provided. Such models and mechanisms are crucial to facilitate a selective dissemination of XML documents, containing information of different sensitivity levels, among (possibly large) subject communities.

In this paper, we have developed our laboratory XML security platform named computer integrated technique (CIT) XML security platform which consists CIT/digital signature model, CIT/ encryption model, smart card crypto model and simple public key infrastructure (SPKI) interface security model for securing XML documents and existing non-XML documents that are exchanged by Internet[4].

CIT/XML signature model provides integrity and authentication to XML documents and non-XML electronic documents. CIT/XML encryption model provides confidentiality to them. Signed or encrypted electronic documents are in the form of XML document and they can be integrated transparently to the existing XML technology and XML-based platforms. XML digital signature and XML encryption use Java-based crypto library and they can be used independent of the platform because they are developed in Java. CIT/XML security platform has the interface for certificate authority (CA) and it can process digital certificates[4] that are needed for the digital signature.

CIT/XML security platform can be applied to various services that require secure electronic documents exchange such as electronic documents interchange (EDI), E-government, E-commerce, B2B (business to business), and B2C (business to customer). Since it is developed in Java, and it can be ported easily to various platforms domestically and internationally because it supports Chinese standard encryption and signature algorithms in addition to famous standard algorithms that are used internationally. Since a Java Crypto library are conform to international standards.

1 Overview of the CIT/XML Security Platform Architecture

The major components of the CIT/XML security platform are XML document security processing subsystem that includes CIT/XML signature module and CIT/XML encryption module, Java crypto module and the interface for certificate authority. Fig. 1 illustrates

Received 2001-11-27.

* The project supported by the National Natural Science Foundation of China (59789502) and 863 High-Tech R&D program (863 – 511 – 030 – 006).

** Born in 1961, male, graduate.

the architecture and relationship among each component. CIT/XML signature module provides digital signature generation and verification that is in the form of XML document and CIT/XML encryption module encrypts and decrypts the electronic documents including XML documents. Java crypto module provides platform independent smart card crypto APIs and they are called by XML document security processing subsystem for digital signature or encryption. XML document security processing subsystem uses digital certificates that are issued by certificate authority for the digital signature.

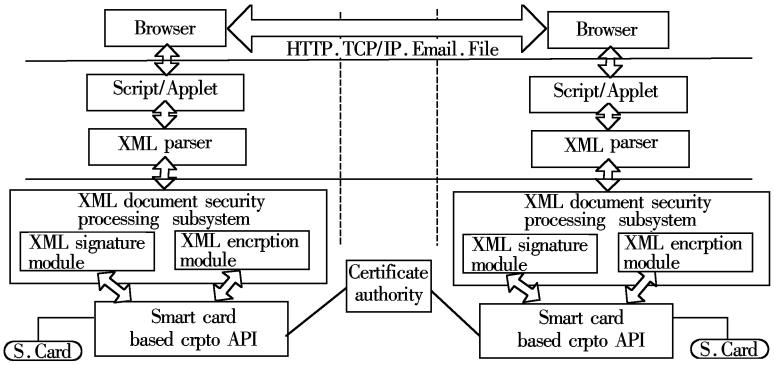

Fig. 1 Overview of the CIT/XML security platform architecture

2 CIT/XML Signature Module

To guarantee the authentication and integrity of the XML documents, XML signature is needed. In CIT/XML security platform design, we implemented the XML signature in the form of APIs that conform to the XML signature draft and support Chinese domestic standard for digital signature.

XML signatures can be applied to any digital content (data object), including XML documents via an indirection. Data objects are digested, the resulting value is placed in an element (with other information) and that element is then digested and cryptographically signed. An XML signature may be applied to the content of one or more resources. XML signature can sign multiple resources at once using just a single signature document. In addition to signing complete XML and non-XML documents, the XML signature allows applications to sign parts of XML documents.

2.1 Syntax of XML signature

The syntax of XML signature is based on XML signature draft and the processing flow[5]. It supports signing complete XML documents, parts of XML documents and even non-XML documents. The resulting signature is a well-formed XML fragment that can either be a standalone XML document or embedded within a more complex XML document. In our model, we support three different kinds of XML signature, which differ from the location of the data object.

- Detached signature The data object is either an external data object, or a local data object included as sibling element in the XML document containing the signature element.
- Enveloped signature The data object encloses the signature element. Thus, the signature element is inserted into the XML document containing the data object being signed.
- Enveloping signature The data object is contained in the object element. Thus, it is part of the signature element.

2.2 Processing flow of XML signature generation

Fig.2 illustrates the structure of the XML signature and the processing flow of XML signature.

1) Resources that are to be signed are accessed and appropriate transform on them are performed. Transforms specified in the XML signature draft are canonicalization (XML-C14N, XML-C14N-a, minimal canonicalization), Base64 encoding, XSLT, XPath transformation and enveloped signature transform. Enveloped signature transform removes the signature structure from the document prior to digesting. Thus, the signing and verification operations will both be performed on a document consisting of everything but the XML signature[5].

2) Message digests are computed over each entity that is to be signed using SHA1.

3) Reference elements that contain URI, transform method, digest method and digest value are generated.

4) If it is necessary to include additional information such as timestamp to the signature, the information can be placed in the signature properties element. To protect the information, a reference for the signature properties are generated and included in the signedInfo and signed together.

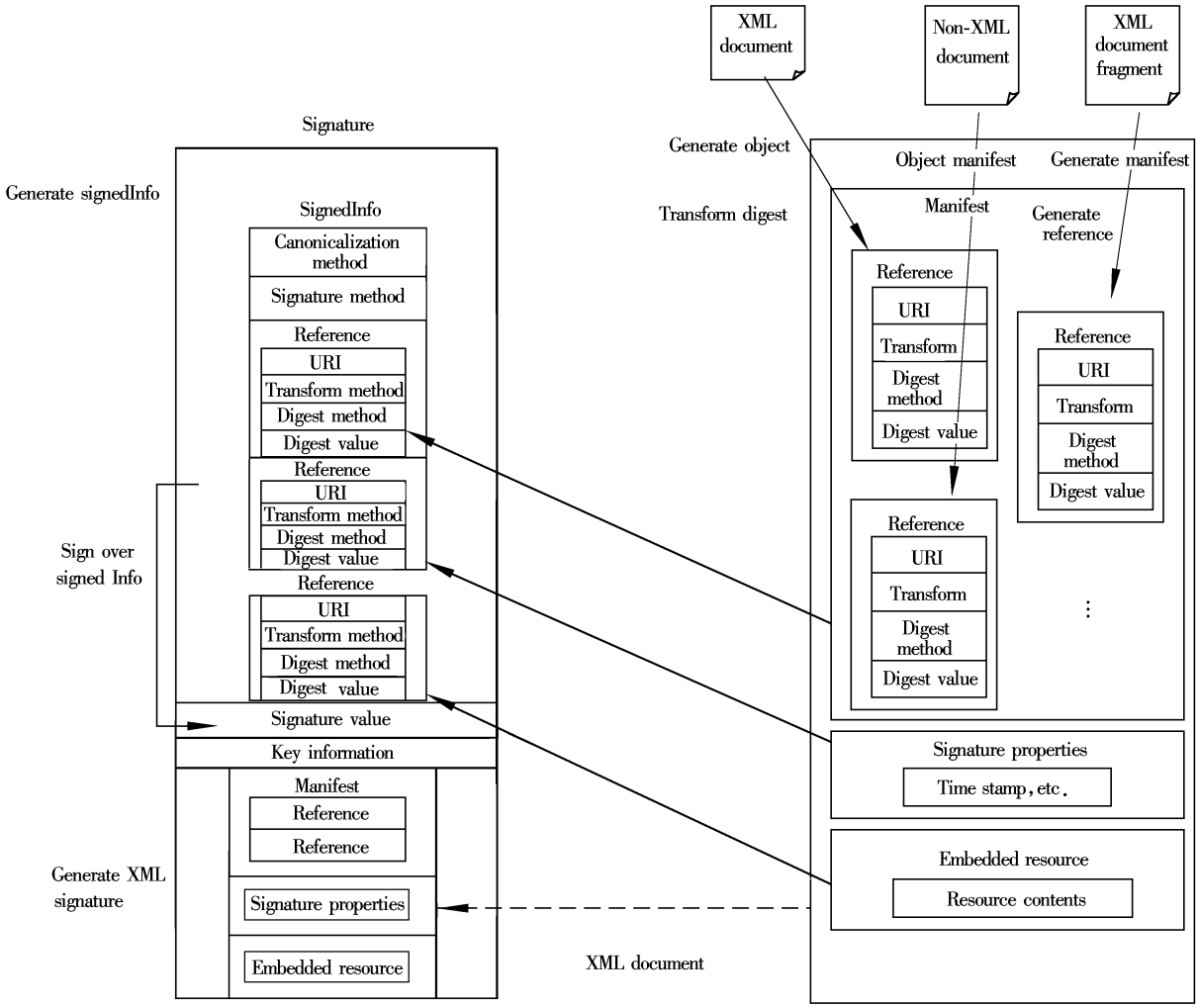

Fig.2 Structure of the XML signature

5) When the enveloping signature is used, the signed data is actually embedded within the XML signature structure. Signed data have to be included in the object element of the signature document.

6) If manifest or signature properties are used, embed them in the object element and generate a reference for it.

7) SignedInfo element is generated which contains canonicalization method information (XML-C14N, XML-C14N-a or minimal canonicalization) for signedInfo itself, signature method information (DSAwithSHA1, RSAwithSHA1), manifest, signature properties, object and references for other resource. This element allows a signature recipient to ensure that no signed entity has been modified. The location information allows the recipient to locate the signed entity. A new message digest can then be computed over this entity and compared with the message digest in the signedInfo. By the properties of cryptographic message digest algorithms, if the document has changed, the message digest will have changed. Therefore, if the two match, the document has not changed from the time of signing.

8) A digital signature is computed over this signedInfo fragment using the signature method indicated by signature method element in the signed info. XML security platform provides DSA with SHA1, RSA and SHA1 for a digital signature. Canonicalization has to be performed before the digital signature computation using the canonicalization method indicated by canonicalizaton method element in the signed info (XML-C14N, XML-C14N-a or minimal canonicalization). The digital signature allows the recipient to ensure first that the signed info fragment has not changed and second that the document was signed by a particular person. The reason for the two phases of signing is that message digest operations are relatively fast, whereas digital signature operations are relatively slow. Computing a digital signature of each entity to be signed would be extremely inefficient; using several message digest operations and a single digital signature is far more efficient.

9) An XML signature element is produced, containing the SignedInfo element, the digital signature value and various additional pieces of information such as the signer's key information, object element, etc. Verification of a signature thus involves both checking the digital signature on the signed info fragment and checking the message digest of each entity listed in the signed info.

2.3 Processing flow of XML signature verification

1) Resources are accessed for verification using the URI information in the corresponding reference element. Then they are transformed using the transform algorithm specified in the transform method element in the reference.

2) Digest values of the resources are computed using the digest algorithm specified in the digest method element in the corresponding reference.

3) Computed digest values are compared with the signature value in the signed Info element. All the references are verified in this way.

4) Signed Info is canonicalized using the algorithm specified in the canonicalization method element in the signedInfo.

5) The signature is verified. First the public key information is obtained from the KeyInfo element and the signature value of the signedInfo is calculated using the signature algorithm specified in the signature method element. The value is compared with the value in the signature value element.

6) Manifest is verified. Digest values of each reference in the manifest are verified. The verification processing is up to application program.

3 CIT/XML Encryption Module

An XML document with private information should be encrypted for security when it is transmitted. The XML document is transformed into byte stream, compressed (optional) and encrypted. An encrypted XML document is encoded in an XML node. Secret key and ancillary information are transmitted with the encrypted XML document. The secret key that is in the form of byte stream is transformed into an XML node after it is encrypted using public key crypto system, and ancillary information (such as the encryption algorithm used, etc.) is also transformed into an XML node.

When we design XML encryption module, first we have to consider confidentiality, a fundamental requirement for securing information, so that the XML document transmitted or stored cannot be revealed. In addition, we should use the cryptographic algorithm secure enough and recognized by some domestic standard authority or national certification authority.

In this design, we provide an option for selecting symmetric key cryptographic algorithms (such as DES, Triple-DES (DESede), RC2, RC4, IDEA etc.) and public key cryptographic algorithms (such as RSA[6,7], ElGamal[8], ECC, etc.). Optional compression of the encoded XML plaintext, prior to encryption, is also supported. This is useful because it reduces the size of the resulting ciphertext. It also greatly reduces the plaintext knowledge to which the attacker has access, in order to mount an attack on the ciphertext.

3.1 Processing flow of XML encryption

Fig.3 illustrates the processing flow of XML encryption. The XML document encryption consists of XML document encryption step, a secret key encryption step and ancillary information-encoding step.

1) A secret key, which will be used in encrypting XML document, is generated using pseudo-random number generator (for symmetric encryption).

2) XML document is encoded in a stream of bytes. It is simply encoded in textual form and translated to a stream of bytes.

3) The encoded byte stream is compressed (optional). It reduces the size of the resulting ciphertext and the plaintext knowledge to which the attacker has access.

4) The stream of bytes is encrypted with a symmetric encryption algorithm (DESede, DES, RC2, RC4, IDEA, etc.) using the secret key generated in step 1).

5) The resulting ciphertext bytes are transformed to a textual encoding. The transformed byte stream is encoded in an XML node.

6) The secret key, used in encrypting an XML document, is encrypted using a particular recipient's public key. In this case, a public key encryption algorithm is used. (RSA, ElGamal, ECC, etc.)

7) The encrypted secret key is transformed to a textual encoding and encoded in an XML node.

8) Ancillary information such as the encryption algorithm used, is encoded as further XML nodes.

9) The resulting XML nodes, generated in step 5), 7) and 8), are organized into a DTD-defined XML structure and returned to the caller.

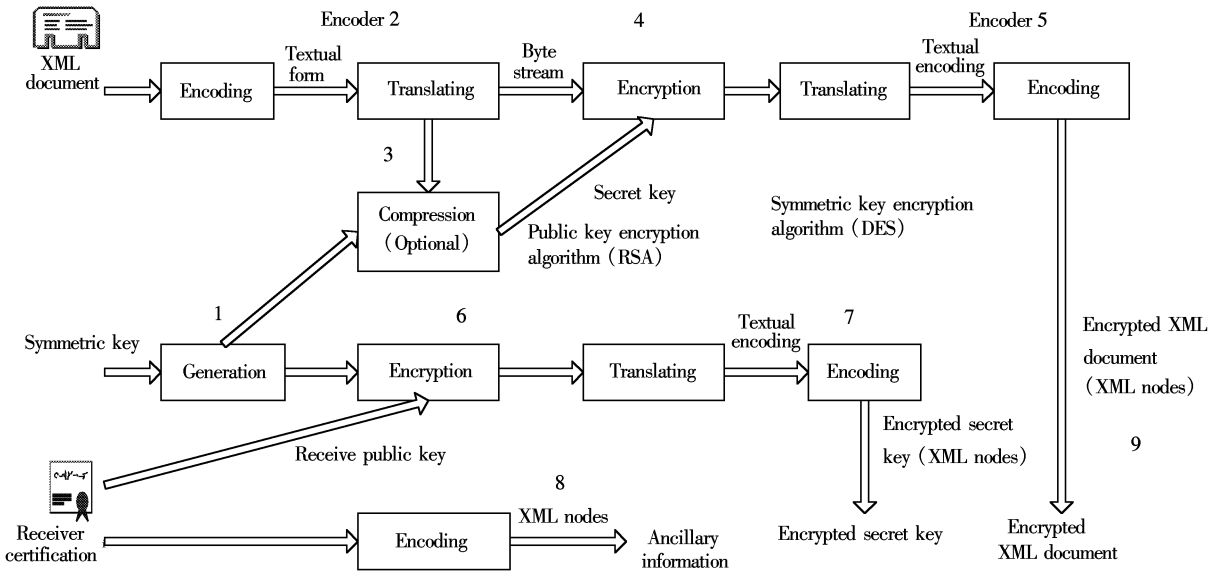

Fig.3 Processing flow of XML encryption

3.2 Processing flow of XML decryption

1) The ancillary information is decoded to check what algorithm was used to encrypt the document.

2) The recipient's private key is used to decrypt the embedded secret key contained within the XML document. (Using RSA, ElGamal[8], ECC, etc.)

3) The embedded XML ciphertext is decoded from its textual encoding and is decrypted using the selected symmetric encryption algorithm and secret key. (Using DESede, DES, RC2, RC4, IDEA, etc.)

4) If the decrypted stream of bytes is compressed, it is decompressed.

5) The resulting stream of bytes is decoded back into an XML structure to form the nodes of the hierarchy.

4 Conclusion

In this paper, we designed a CIT/XML security platform that provides security services such as authentication, integrity and confidentiality for XML-based electronic document. It provides digital signature function, encryption function, Java Crypto library and PKI-based security functions for securing XML documents and existing non-XML documents that are exchanged among the various servers. Signed or encrypted electronic documents are in the form of XML document and they can be integrated transparently to the existing XML technology and XML-based electronic commerce platforms. Existing XML applications can be easily extended to use XML signature and XML encryption by adding XML signature and XML encryption DTD to the existing DTD of the applications, and adding CIT/XML signature and CIT/encryption modules to them. XML digital signature and XML encryption use java-based crypto library and they can be used independent of the platform because they are developed in Java.

References

[1] Bertino E, Castano S, Ferrari E. On specifying security policies for Web documents with an XML-based, language [J]. *ACM Symposium on Access Control Models and Technologies*, Fairfax, VA, May 2001.

[2] W3C. XML 1.0 Recommendation[EB/OL]. http://www.w3.org/TR/1998/REC-xml-19980210, February 1998.

[3] W3C. XML-Signature Syntax and Processing[EB/OL]. http://www.w3.org/TR/2001/, August 2001.

[4] Frank Boumphrey. *Professional XML applications* [M]. WROX, 1999.

[5] Elisa bertino, Barbara carminat. XML security [EB/OL]. Report http://www.elsevier.com, 2001.

[6] Dr. Shimshon Berkovits. Public key infrastructure study [EB/OL]. Final Report, 1994.

[7] Rivest R L, Shamir A, Adleman L M. A method for obtaining digital signatures and public-key cryptosystems [J]. *Communications of the ACM*, 1978.

[8] ElGamal T. A public-key cryptosystem and a signature scheme based on discrete [J]. *Communications of the ACM*, April 1985.

[9] Microsoft. SDK cryptography, smart card cryptography [EB/OL]. http://www.micosoft.com. August 1999.

[10] Schneier B. *Applied cryptography* [M]. Wiley, 1996.

CIT/XML 安全平台语义与处理

安 南 张申生

(上海交通大学计算机科学与工程系, 上海 200030)

摘 要 Web 正成为公司和组织内部以及和外界传播信息的主要方式.信息发布通常在 Web 服务器端采用 XML 文档的形式,或者通过 Web 服务器将 XML 文档主动发送给感兴趣的客户端.这些文档通常含有程度不同的敏感信息,所以必须有一个强大的 XML 安全平台和机制.在本文中我们提出了 CIT/XML 安全平台,并详细介绍了 CIT 数字签名,CIT 加密模型,CIT 智能卡加密和 SPKI 接口安全模型的语义和处理.提供了对在各种服务器间交换的 XML 文档以及非 XML 文档的安全服务,如认证、完整性以及机密性.

关键词 电子商务, 安全, 数字证书, 智能卡, 数字商务, 认证, SPKI, XML

中图分类号 TP309.7