第 32 卷 第 14 期 计 算 机 工 程 2006 年 7 月
Vol.32 № 14 Computer Engineering July 2006
·安全技术· 文章编号：1000—3428(2006)14—0141—02 文献标识码：A 中图分类号：TP 311

# 基于加密过程控制语言的 XML 数据加密方案

陈 祥[1]，左洪福[2]
（1. 南京航空航天大学信息科学与技术学院，南京 210016；2. 南京航空航天大学民航学院，南京 210016）

**摘 要：**目前国内外的 XML 数据加密方法只针对某一特定的应用，缺乏对加密过程的描述，通用性不佳，数据加密过程效率不高。该文在 XML 文档对称加密与非对称加密的方案基础上，提出了一种加密过程控制语言，用以控制 XML 文档的加密过程，从而提高了 XML 数据加密的效率和通用性，同时给出了这种加密方案的实现。

**关键词：**XML；加密；加密过程控制语言

# Design and Implementation of XML Data Encryption with Encryption Process Control Language

CHEN Xiang[1], ZUO Hongfu[2]
(1. College of Information Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016;
2. College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016)

**【Abstract】** Now XML data encryption method is focus on special application, lacks of description for encryption process, it is not common and efficient. A new language which controls the encrypting process is designed based on symmetric cryptography and asymmetric cryptography. With this language, encrypting process could be universal and more efficient. Furthermore, implementation of this method is provided.

**【Key words】** XML; Encryption; Language which controls the encrypting process

可扩展标记语言(eXtensible Markup Language，XML)是由 W3C 制定的一种数据标准，XML 因其结构化、可扩展、与平台无关的特性，被广泛地应用于不同平台、不同应用系统、不同语言之间数据交换的标准，逐步成为 Internet 交换数据的一种机制。然而 XML 作为一种数据载体，并没有实现对数据的保护。因此 XML 数据的安全问题就成为了一个迫切需要解决的问题。传统的数据安全方案是通过对整个数据进行加密以便在网络上进行传输(如 SSL、TSL)，它们能够保证数据传输的充分安全，然而这种方案只能实现文件级别的加密，对于文件中某个元素加密则无能为力，加密粒度显得过粗。XML 加密可以对 XML 文件任意内容加密，包括 XML 文档、元素以及元素内容等。

## 1 XML 加密规范

XML 加密生成的文件仍然是格式正确的 XML 文件，使用<EncryptedData>元素来构建加密数据，其中 Id 属性是直接的任意标识符，可以应用于具体的应用，Type 属性用于描述明文的类型。<EncryptionMethod>元素用来表示加密算法，<CipherData>元素用来以某种形式描述加密后的数据，<ds:KeyInfo>元素描述了如何获得密钥来对<CipherData>中的加密数据解密。

〈EncryptedData Id ? Type ?〉
〈EncryptionMethod/ 〉?
〈ds :KeyInfo〉
〈EncryptedKey〉?
〈AgreementMethod〉?
〈ds :KeyName〉?
〈ds :RetrievalMethod〉?
〈ds : 3 〉?
〈/ ds :KeyInfo〉?
〈CipherData〉
〈CipherValue〉?
〈CipherReference URI ?〉?
〈EncryptionProperties〉?
〈/ EncryptedData〉

## 2 XML 文档加密过程控制语言

为了加密 XML 文档，需要描述关于加密文档的若干信息以及加密步骤的信息，如加密元素的位置、加密算法、加密密钥等，经过分析，这些信息可以归纳为以下两类：

(1)加密对象描述信息：包括加密粒度信息以及加密对象位置信息，其中加密粒度信息表示加密的内容是整个文档、元素、文本或者是其它加密算法的密钥。

(2)加密的描述信息：包括加密算法和加密密钥的描述信息。

### 2.1 加密对象描述

本文设计了一个 EncryptionTarget 复杂类型描述加密对象信息，该复杂类型包括两个元素：EncryptionGranularity 和 TargetObject。EncryptionGranularity 用以描述加密粒度，取值为：“ All ”, “ Node ”, “ Text ”, “ Key ”，分别表示加密整个文档、元素、文本和密钥。TargetObject 元素用以描述加密元素信息，它包括 TargetElement 和 KeyURI 两个元素中的一个，TargetElement 元素描述加密元素的位置信息，KeyURI 元素表示需要加密的密钥的 URI。EncryptionTarget 的模式定义如下(xs 表示命名空间 xmlns="http://www.w3.org/2001/XMLSch

**作者简介：**陈 祥(1981—)，男，硕士生，主研方向：软件工程；左洪福，教授
**收稿日期：**2005-11-15 **E-mail：**tallman@163.com

ema")：

```
<xs:complexType name="EncryptionTarget">
<xs:sequence>
<xs:element name="EncryptionGranularity">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="All"/>
<xs:enumeration value="Node"/>
<xs:enumeration value="Text"/>
<xs:enumeration value="Key"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="TargetObject" minOccurs="0">
<xs:complexType>
<xs:choice>
<xs:element name="TargetElement" type="xs:string"/>
<xs:element name="KeyURI" type="xs:anyURI"/>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
```

**2.2 加密描述信息**

本文设计了 Encryption 复杂类型描述加密描述信息，该复杂类型包括 EncryptionMethod 元素和 KeyInfo 元素。EncryptionMethod 描述加密算法信息(限于篇幅，本文只列出 DES 和 RSA 两个加密算法)。KeyInfo 元素描述密钥信息，它包括 DES 元素或者 RSA 元素，前者描述了使用 DES 加密算法所涉及到的秘钥，包括 KeyURI 和 InitValueURI 两个元素，分别指向 DES 加密算法中密钥和初始向量值文件的 URI。后者描述了使用 RSA 加密算法所涉及到的公钥和私钥的文件的 URI，分别对应 PublicKeyURI 和 PrivateKeyURI 元素：

```
<xs:complexType name="Encryption">
<xs:sequence>
<xs:element name="EncryptionMethod">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="DES"/>
<xs:enumeration value="RSA"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="KeyInfo">
<xs:complexType>
<xs:choice>
<xs:element name="DES">
<xs:complexType>
<xs:sequence>
<xs:element name="KeyURI" type="xs:anyURI"/>
<xs:element name="InitValueURI" type="xs:anyURI"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="RSA">
<xs:complexType>
<xs:sequence>
<xs:element name="PublicKeyURI" type="xs:anyURI"/>
<xs:element name="PrivateKeyURI" type="xs:anyURI"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:choice>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
```

**2.3 加密步骤信息**

XML 加密过程可能涉及到许多步骤，例如对某个元素进行对称加密，再对对称加密的密钥进行非对称加密等，因此本文设计了 EncryptionStep 复杂类型描述加密步骤，它包括：Step，EncryptionTarget 和 Encryption 元素，分别对应加密步骤(描述该步骤是加密过程中的第几个步骤)、加密对象描述和加密描述信息。

```
<xs:complexType name="EncryptionStep">
<xs:sequence>
<xs:element name="Step" type="xs:integer"/>
<xs:element name="EncryptionTarget" type="EncryptionTarget"/>
<xs:element name="Encryption" type="Encryption"/>
</xs:sequence>
</xs:complexType>
```

**2.4 加密过程**

本文设计 EncryptionProcess 元素来表示整个加密的过程，它包括若干个加密步骤元素，即 EncryptionStep，这样 EncryptionProcess 就可以表示整个加密过程.

## 3 XML 加密实现

本部分介绍在.NET 框架下，使用 C#.NET 语言实现 XML 文件的加密。为表示加密过程描述语言，本文设计如下所示的结构体数组来表示整个加密过程：

```
struct TargetObject{//确定需要加密的明文信息
public string targetElement; //加密元素的路径
public string keyURI;}//需要加密的密钥的文件路径
struct DES{
public string keyURI;//DES 加密算法密钥文件的路径
public string initValue;}//DES 加密算法初始向量文件的路径
struct RSA{
public string publicKeyURI;//RSA 公钥文件路径
public string privateKeyURI;}//RSA 私钥文件路径
struct KeyInfo{//密钥信息
public DES desInfo;
public RSA rsaInfo;}
struct EncryptionTarget{//需要加密的对象信息
public string encryptionGranularity;//加密粒度信息
public TargetObject targetObject;}
struct Encryption{
public string encryptionMethod;//加密算法描述信息
public KeyInfo keyInfo;}//加密算法相关密钥信息
struct EncryptionStep{//加密步骤描述信息
public EncryptionTarget encryptionTarget;
public Encryption encryption;}
EncryptionStep[] encryptionProcess;//加密过程的数组
```

通过 DOM 编程接口技术读取加密过程控制文件来初始化该数组，然后根据数组中每个元素的信息来加密明文。这里采用 DES 算法来加密 XML 数据明文，再使用 RSA 算法加

(下转第 145 页)