ICS 35.808
R 07
备案号:

JT

中华人民共和国交通运输行业标准

JT/T 1021—2016

交通运输信息系统
基于 XML 的数据交换通用规则

Transportation information system—XML based data exchange general rules

2016-02-02 发布　　2016-04-10 实施

中华人民共和国交通运输部　发布

目　　次

前言 …… Ⅲ
引言 …… Ⅳ
1　范围 …… 1
2　规范性引用文件 …… 1
3　术语和定义 …… 1
4　数据交换包数据类型 …… 2
5　数据交换包分类与组成 …… 3
6　数字签名与验证、数字加密与解密及数据压缩与解压 …… 5
附录 A(资料性附录)　通用规则中 W3C XML Schema 数据类型 …… 13
附录 B(规范性附录)　数据交换包的 XML Schema …… 21
附录 C(资料性附录)　数据交换包的 XML 格式应用实例 …… 27
附录 D(资料性附录)　基于 Web Service 方式的数据交换包的 XML 格式应用实例 …… 31
附录 E(资料性附录)　XML 加密粒度说明应用实例 …… 33
参考文献 …… 36

前　言

本标准按照 GB/T 1.1—2009 给出的规则起草。

本标准由交通运输信息通信及导航标准化技术委员会提出并归口。

本标准起草单位：长安大学、交通运输部公路科学研究院、北京中交国通智能交通系统技术有限公司。

本标准主要起草人：安毅生、张绍阳、赵怀鑫、尚龙华、曹金山、钱越、张建苍、王琪琳、雷甜。

引　言

在交通运输信息化中，存在大量需要进行信息共享和数据交换的业务系统，而各业务系统使用的数据库类型、共享交换的数据类型以及共享交换的模式（集中式、分布式等）也不尽相同。为了方便交通运输信息系统之间数据交换的实现，需要对基于 XML 的数据交换通用规则进行标准化。基于 XML 的数据交换通用规则，将为交通运输信息系统中各业务领域的数据交换接口的制定提供基础性规则，这对于减少接口规范的编写工作量、保证业务系统在大范围内进行信息交换都具有重要的作用。

针对数据交换过程中数据类型、格式的对应关系问题，在定义数据交换包及数据类型时分别参考了 GB/T 18793—2002《信息技术　可扩展置标语言（XML）1.0》和 GB/T 7408—2005《数据元和交换格式　信息交换　日期和时间表示法》。在定义数据交换包数字签名与加密文档结构时参考了 W3C XML 安全工作组发布的《XML　签名语法与处理》和《XML　加密语法与处理》。

交通运输信息系统
基于 XML 的数据交换通用规则

1 范围

本标准规定了交通运输信息系统基于 XML 的数据交换通用规则，包括数据交换包数据类型、数据交换包分类与组成、数字签名与验证、数字加密与解密、数字压缩与解压。

本标准适用于交通运输信息系统之间数据交换格式的制定。

本标准不适用于数据交换过程中的数据链路的建立、维护和撤销。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 20518 信息安全技术 公钥基础设施 数字证书格式

GB/T 25069 信息安全技术 术语

JT/T 697.1 交通信息基础数据元 第1部分：总则

XML 签名语法与处理(1.1 版)(XML Signature Syntax and Processing Version 1.1)

XML 加密语法与处理(1.1 版)(XML Encryption Syntax and Processing Version 1.1)

3 术语和定义

GB/T 20518 和 GB/T 25069 界定的以及下列术语和定义适用于本文件。为了便于使用，以下重复列出了 GB/T 20518 和 GB/T 25069 中的某些术语和定义。

3.1

数据交换包 data exchange packet

包含不同业务系统之间需要进行信息交换的数据及数据属性的文本，由数据包头、数据包体组成，其长度可变。

3.2

数据包头 packet header

数据包基本属性信息的文本，包括数据包标识、数据包功能、数据包参考号、发送方标识、接收方标识、数据包生成日期及时间、数据加密、数据压缩等属性。

3.3

数据包体 packet body

不同业务系统之间需要共享或交换的数据。

3.4

数字证书 digital certificate

由国家认可的，具有权威性、可信性和公正性的第三方证书认证机构(CA)进行数字签名的一个可信的数字化文件。

[GB/T 20518—2006，定义 3.7]

3.5

数字签名　digital signature

附加在数据单元上的数据，或是对数据单元所做的密码变换，这种数据或变换允许数据单元的接收者用以确认数据单元的来源和完整性，并防止数据被人（例如接收者）伪造或抵赖。

[GB/T 25069—2010，定义 2.2.2.176]

3.6

签名内容描述　signature content description

发送方对所签名的数据进行的说明。

3.7

加密　encipherment/encryption

对数据进行密码变换以产生密文的过程。一般包含一个变换集合，该变换使用一套算法和一套输入参量。输入参量通常被称为密钥。

[GB/T 25069—2010，定义 2.2.2.60]

3.8

加密内容描述　encryption content description

发送方对所加密的数据进行的必要说明。

3.9

数据压缩　data compress

在不丢失信息的前提下，缩减数据量以减少存储空间，提高其传输、存储和处理效率的一种技术方法。

3.10

压缩内容描述　compress content description

发送方对所压缩的数据进行的说明。

4　数据交换包数据类型

数据交换过程中的数据类型、格式等属性应遵照 JT/T 697.1 的规定，常用数据元的数据类型和数据格式与 W3C XML Schema 中的数据类型和数据格式的对应关系见表 1。在数据交换包生成过程中，应按照表 1 的规定进行数据类型转换。若表 1 中的 XML Schema 数据类型无法满足用户需求，则可采用 W3C 规定的 XML 数据类型。

完整的 XML Schema 数据类型及其层次结构说明参见附录 A。

表 1　基本数据类型对应关系

Schema 数据类型	Schema 数据格式	对应的 JT/T 697.1 中的数据类型和格式
boolean	合法的布尔值是 true、false、1（表示 true）以及 0（表示 false）	布尔型：True/False；0/1
date	"YYYY-MM-DD"	日期型：YYYY；YYYYMM；YYYYMMDD
time	"hh:mm:ss"	时间型：hhmmss
dateTime	"YYYY-MM-DDThh:mm:ss"	日期时间型：YYYYMMDDhhmmss
int	32 位的有符号整数	数字型：ns. . m
float	IEEE 单精度 32 位浮点数	数字型：ns. . m，k

表 1(续)

Schema 数据类型	Schema 数据格式	对应的 JT/T 697.1 中的数据类型和格式
double	64 位的双精度浮点数	数字型:ns. . m,k
string	字符串数据类型可包含字符、换行、回车以及制表符	字符型:a. . m;an. . m;am;anm;n. . m;nm
base64Binary	以 base64 编码保存的任意二进制数据	二进制型

5 数据交换包分类与组成

5.1 数据交换包分类

根据交通运输信息系统交换双方之间是否具有约定,可将数据交换包分为两种类型:

——通用数据交换包:若交换双方对交换内容事先无约定,则数据交换包除了要包含所要交换的数据外,还需要携带交换数据的描述信息,以供接收方理解和处理交换数据;

——简体数据交换包:若交换双方对交换内容事先有约定,则数据交换包只包含所需交换的数据内容,而不需要包含交换数据的描述信息。

5.2 通用数据交换包组成

通用数据交换包由数据包头、数据包体两部分组成:

——数据包头:由数据包标识、数据包功能、数据包参考号、发送方标识、接收方标识、数据包生成日期、数据包生成时间、数字签名标识、签名数据描述、数据加密标识、数据加密方法、加密数据描述、数据压缩标识、数据压缩方法、压缩数据描述、交换协议版本号、消息序列号以及数据包总数量组成;

——数据包体:主要由若干数据项组成。

完整的通用数据交换包结构见图 1。通用数据交换包的 Schema 文档要求见附录 B;应用实例参见附录 C。

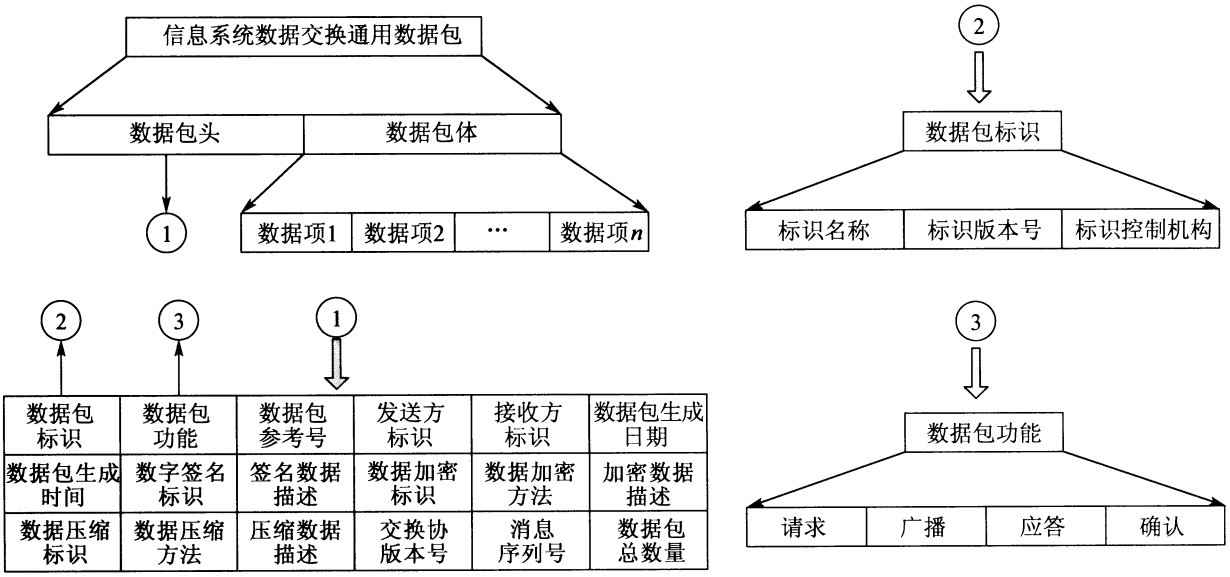

图 1 数据交换包的组成

数据交换包组成元素及属性见表2。

表2 数据交换包组成元素及属性

索 引	XML 标签名	说 明	约 束	出现次数	数据格式	数据类型
01	DataPacketHeader 数据包头	标识数据包头的开始	M	1..1		
02	DataPacketID 数据包标识	标识数据包基本信息的节点	M	1..1		
03	NameOfDataPacketID 标识名称	数据包的标识的名称	M	1..1	an..70	string
04	DataPacketVersion 标识版本号	数据包标识的版本号信息	O	0..1	an..9	string
05	ControllingAgency 标识控制机构	给出该数据包标识的控制机构的名称或代码	O	0..1	an..70	string
06	DataPacketFunction 数据包功能	给出数据包的功能,例如请求、广播、应答及确认等	M	1..1	an..35	string
07	DataPacketRefID 数据包参考号	该数据包的唯一参考号	M	1..1	an..35	string
08	SenderID 发送方标识	数据包发送方的唯一标识,由网络服务提供商分配	M	1..1	an..70	string
09	RecipientID 接收方标识	数据包接收方的唯一标识,由网络服务提供商分配	M	1..n	an..70	string
10	DataPacketGenDate 数据包生成日期	该数据包生成的日期	M	1..1	YYYY-MM-DD	date
11	DataPacketGenTime 数据包生成时间	该数据包生成的时间	M	1..1	hh:mm:ss	dateTime
12	DataSigID 数字签名标识	标识是否有数字签名	O	0..1	true/false	boolean
13	DataSigDesc 签名数据描述	对被签名数据进行的简要描述	O	0..1	an..100	string
14	DataEncID 数据加密标识	该数据包中数据是否有被加密	O	0..1	true/false	boolean
15	DataEncMethod 数据加密方法	加密方法的名称	O	0..1	an..16	string
16	EncDataDesc 加密数据描述	该数据包中被加密数据的简要描述	O	0..1	an..100	string

表 2(续)

索　引	XML 标签名	说　　明	约　束	出现次数	数据格式	数据类型
17	ComDataID 数据压缩标识	该数据包中数据是否有被压缩	O	0..1	true/false	boolean
18	ComDataMethod 数据压缩方法	数据压缩方法的名称	O	0..1	an..16	string
19	ComDataDesc 压缩数据描述	该数据包中被压缩的数据的简要描述	O	0..1	an..100	string
20	ExProtocolVersionNum 交换协议版本号	数据交换双方采用的标准协议版本编号	M	1..1	an..70	string
21	MessageSequeNum 数据包序列号	发送包的顺序序列号	M	1..1	an..70	string
22	DataPacketsAmount 数据包总数量	数据交换包的数量	M	1..1	n1..9	int
23	DataPacketBody 数据包体	数据包中涉及业务系统之间需要交换的数据的部分	M	1..1		
24	DataItem 数据项	标识每一条需要交换的数据	M	1..n		

注 1:M——必选;O——可选。
注 2:"0..1"——不出现或出现一次;"1..1"——出现且仅出现一次;"1..n"——必选且可出现多次。

5.3　简体数据交换包组成

简体数据交换包不含数据包头部分,数据包体部分的要求与通用数据交换包相同。附录 D 给出了简体数据交换包的实例。

6　数字签名与验证、数字加密与解密及数据压缩与解压

6.1　概述

交通运输数据在以 XML 格式进行交换时,可对数据包体进行数字签名及验证、数据加密及解密、数据压缩及解压等处理。

交通运输数据交换处理流程见图 2。

6.2　数字签名与验证

6.2.1　概述

本标准采用《XML　签名语法与处理(1.1 版)》中规定的 XML 数字签名规范。

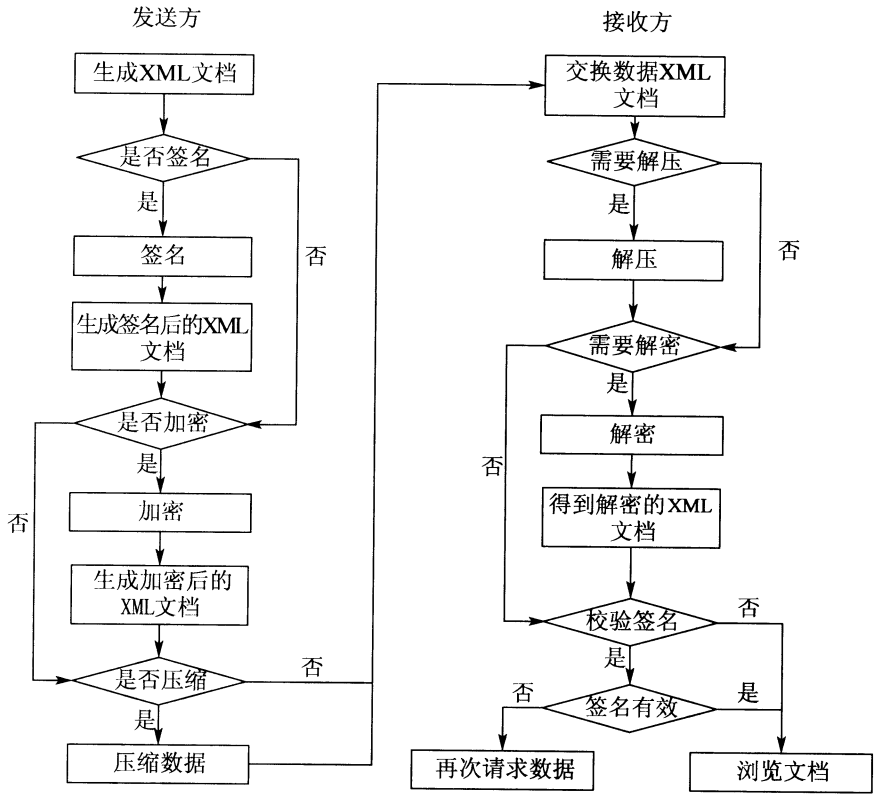

图 2　数据交换处理流程示意图

6.2.2　XML 数字签名规范文档

6.2.2.1　XML 数字签名规范文档结构

XML 数字签名规范文档由 Signature 元素表示，具有以下结构：

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

注：? ——表示出现 0 或 1 次，+ ——表示至少出现 1 次，* ——表示出现 0 或多次。

6.2.2.2　XML 数字签名规范文档元素

XML 数字签名规范文档元素说明见表 3。

表 3　XML 数字签名规范文档元素说明

元 素 名 称	元 素 说 明
< Signature >	XML 数字签名的根元素，包括 < SignedInfo > 、< SignatureValue > 、< KeyInfo > 、< Object > 4 个子元素，其中 < SignedInfo > 和 < SignatureValue > 元素应出现且仅能出现 1 次
< SignedInfo >	包含了被签名数据的相关信息，是 XML 数字签名的核心元素，包括 < CanonicalizationMethod > 、< SignatureMethod > 、< Reference > 3 个子元素
< Canonicalization-Method >	描述了规范化算法。数字签名对被签名对象的计算过程是基于位运算的，而规范化算法即为在被签元素做签名运算之前对其做规范化处理，使它们在物理表现上相同，应出现且仅能出现 1 次
< SignatureMethod >	描述了将已规范化过的 < SignedInfo > 元素转换为数字签名值的算法，应出现且仅能出现 1 次
< Reference >	描述了被签名对象的相关信息，如位置信息、摘要算法、摘要值等。其下包括 < Transforms > 、< DigestMethod > 、< DigestValue > 3 个子元素，至少出现 1 次
< Transforms >	由若干有序的 < Transform > 子元素组成，为可选元素。它们描述了签名者如何获取签名对象。第一个元素的输入参数由 < Reference > 元素的 URI 指定，最后一个 < Transform > 的输出是摘要算法 < DigestMethod > 的输入参数
< DigestMethod >	描述了对相应数字签名对象使用的摘要算法。本标准建议使用 SHA-256 算法
< SignatureValue >	记录了数字签名的值
< KeyInfo >	用于描述密钥信息并主要用于签名的验证，是一个可选元素
< Object >	可以包括任意数据对象及 < Manifest > 、< SignatureProperities > 等子元素，为可选元素

6.2.3　XML 数字签名生成及校验过程

6.2.3.1　XML 数字签名生成

XML 数字签名的生成过程包括引用生成(Reference Generation)和签名生成(Signature Generation)，详细步骤见图 3。

其中被签名的数据对象由 < Reference > 元素的 URI(Uniform Resource Identifier)指出，其 URI 属性不仅可指定 XML 文档内部的某个元素，还可指定本地或网络上的文本或二进制数据。采用多个 < Reference > 元素分别指定不同的签名对象可实现对多个数据进行签名。

根据 XML 签名元素和被签名对象之间的关系，XML 的签名可使用下列 3 种方式之一：

a)　封装式签名(Enveloping Signature)：签名数据被封装在 XML 签名元素 < Signature > 内部；

b)　嵌入式签名(Enveloped Signature)：XML 签名元素 < Signature > 被嵌入到被签名数据中；

c)　分离式签名(Detached Signature)：XML 签名元素和被签名数据是彼此分离的，两者之间不存在包含与被包含的关系。被签名的数据可是独立的外部文档，也可是与 Signature 元素位于同一 XML 内部的并列的兄弟元素。

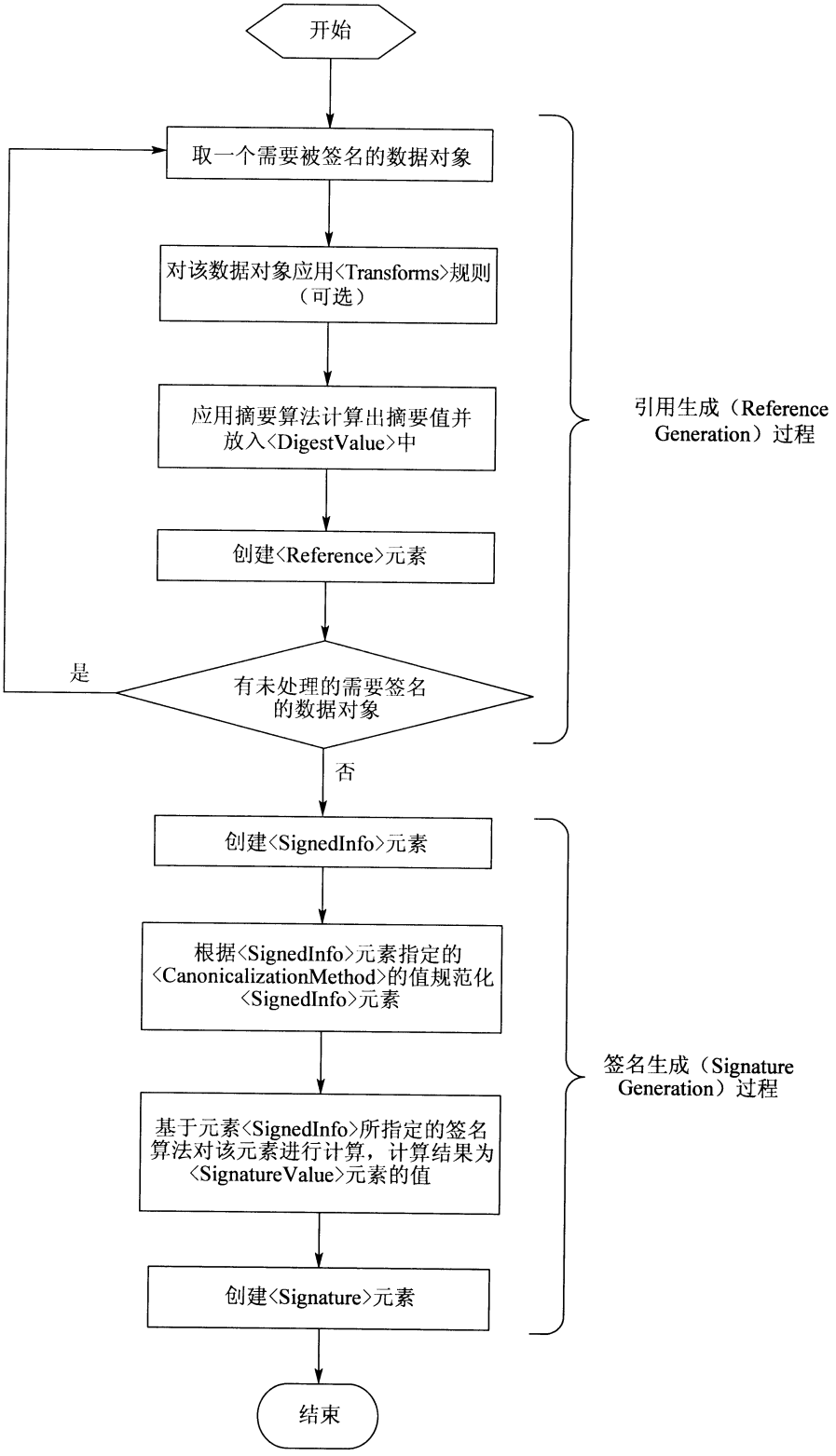

图3　XML 数字签名生成过程

6.2.3.2　XML 数字签名的校验

XML 数字签名校验过程包括引用确认(Reference Validation)和签名确认(Signature Validation)。引用确认是对 < SignedInfo > 元素中每个 < Reference > 元素所包含的摘要值进行校验,为了保证被签名对象没有被做任何修改。签名确认则是对元素 < SignedInfo > 中的元素 < SignatureValue > 值进行校验,确保签名人身份的真实性及数据的完整性。详细步骤如图4 所示。

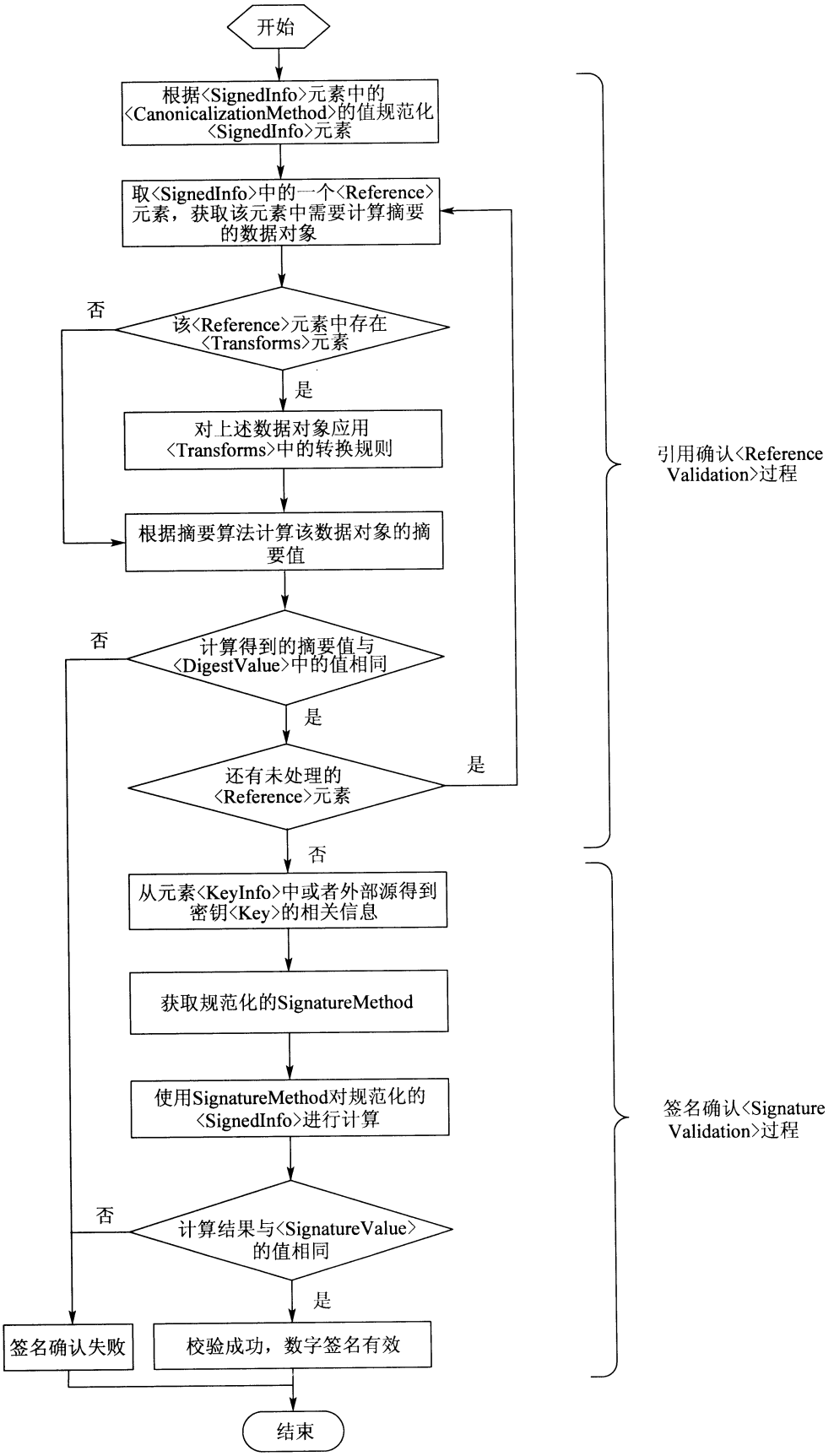

图 4　XML 数字签名校验过程

关键处理过程描述如下：

a） 引用确认过程：

1） 基于 < SignedInfo > 元素中的 < CanonicalizationMethod > 指定的规范化算法对 < SignedInfo > 元素进行规范化；

2） 对 < SignedInfo > 元素中的每一个 < Reference > 元素进行摘要值校验；

3） 若引用确认成功，则进行签名确认，否则失败。

b） 签名确认过程：

1） 从 < KeyInfo > 元素或者一个外部源中获取密钥信息；

2） 使用 < CanonicalizationMethod > 指定的规范化算法获取 < SignatureMethod > 的规范化形式，对 < SignedInfo > 元素的签名值进行确认；

3） 若签名确认成功则整个校验过程成功，否则失败。

6.3 数据加密与解密

6.3.1 概述

本标准采用《XML 加密语法与处理(1.1 版)》中规定的 XML 加密解密规范。

6.3.2 XML 数据加密规范文档

6.3.2.1 XML 数据加密规范文档结构

XML 数字加密规范文档由 EncrypedData 元素表示，其结构如下：

```
< EncryptedData Id? Type? MimeType? Encoding? >
        < EncryptionMethod/ > ?
        < ds:KeyInfo >
                < EncryptedKey > ?
                < AgreementMethod > ?
                < ds:KeyName > ?
                < ds:RetrievalMethod > ?
                < ds: * > ?
        </ds:KeyInfo > ?
        < CipherData >
                < CipherValue > | < CipherReferenceURI? >
        </CipherData >
        < EncryptionProperties > ?
</EncryptedData >
```

注：? ——表示出现 0 或 1 次，+ ——表示至少出现 1 次，* ——表示出现 0 或多次，| ——表示出现其中之一。

6.3.2.2 XML 数据加密规范文档元素

XML 数字加密规范文档元素说明见表 4。

表 4 XML 数字加密规范文档元素说明

元素名称	元素说明
EncryptedData	对加密的粒度、编码方式等进行说明
EncryptionMethod	说明加密所使用的算法

表4(续)

元素名称	元素说明
KeyInfo	说明加密所使用的密钥信息
EncryptedKey	描述一个被加密密钥
AgreementMethod	描述通信双方的密钥协商算法
ds:KeyName	描述被加密密钥的名字
ds:RetrievalMethod	描述获得被加密密钥的方法
CipherData	封装或关联生成的加密数据,且CipherData应包含子元素CipherValue和CipherReference之一
CipherValue	描述实际密文的值
CipherReference	描述密文数据的引用
EncryptionProperties	对与加密相关的其他信息进行说明

6.3.3 XML数字加密及解密处理过程

XML数字加密及解密处理的简要流程见图5。

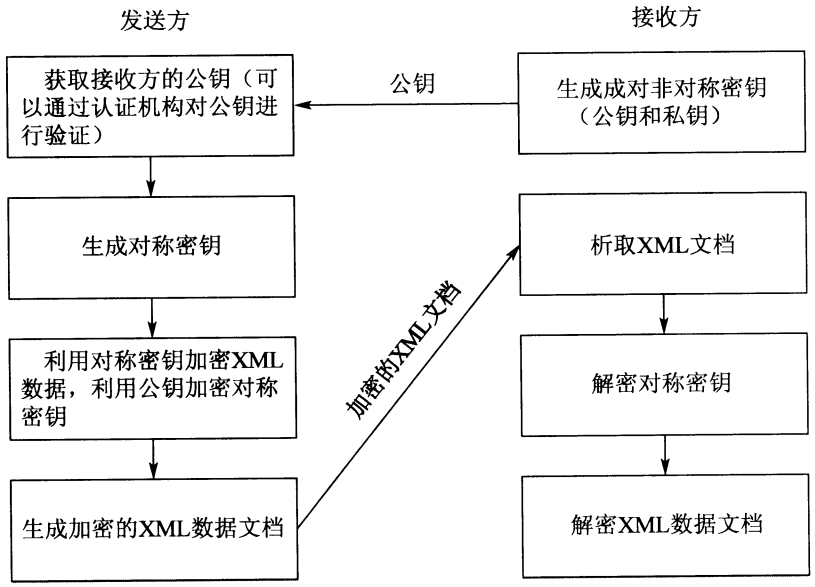

图5 XML数字加密及解密处理流程

6.3.4 XML数字加密类型

XML数字加密规范按照加密粒度的不同进行了分类,本标准中采用如下两种加密粒度:

a) 数据包体加密:将XML文档中元素的子元素或者元素中的字符内容作为加密对象,用<EncryptedData>元素替换相应被加密对象;

b) 多重加密:将XML文档中的一个或多个<EncryptedData>元素作为加密对象,并用新生成的<EncryptedData>元素替换相应被加密对象。

有关 XML 加密粒度详细说明的示例文档参见附录 E。

6.4 数据压缩与解压

针对 XML 文件的压缩方法包括 XMill、XMLPPM 及 XGrind 等，用户可根据不同的情况选择不同的压缩和解压方法。

附 录 A
（资料性附录）
通用规则中 W3C XML Schema 数据类型

A.1 Schema 内置的数据类型的层次结构

Schema 内置的数据类型的层次结构，见图 A.1。

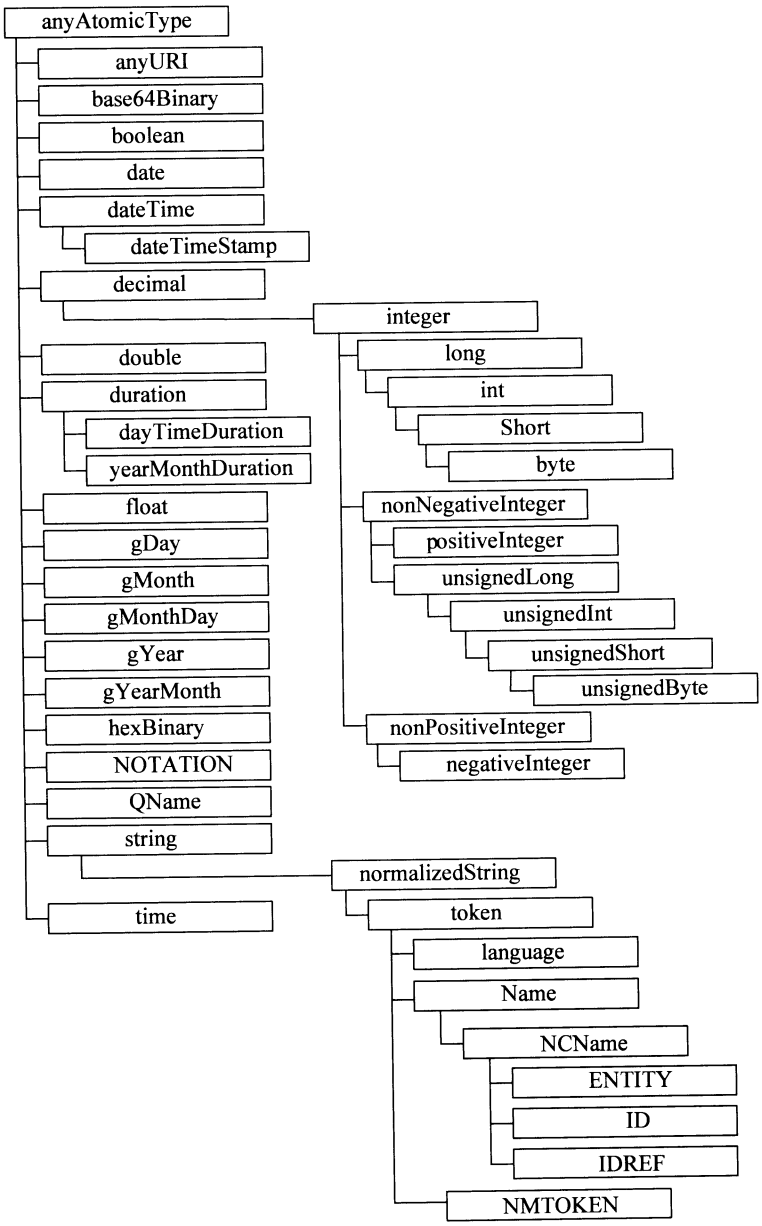

图 A.1 Schema 内置的数据类型层次结构示意图

A.2 数据类型

图 A.1 中出现的数据类型的简单定义与使用示例：

a） anyURI 类型：

- 定义:用于规定 URI。即表示一个统一资源标识符(URI)的引用。anyURI 值可以是绝对的,也可以是相对的;
- Schema 中 anyURI 声明示例:<xs:attribute name = "src" type = "xs:anyURI"/>;
- 文档中的元素声明示例:< pic src = " http://www. w3school. com. cn/images/smiley. gif" />。

b) base64Binary 类型:

- 定义:以 base64 编码保存的任意二进制数据。其值只能由 a ~ z、A ~ Z、0 ~ 9 和加号(+)等字符组成,其长度是 4 的倍数;
- Schema 中 base64Binary 声明示例:<xs:element name = "hehe" type = "xs: base64Binary " />;
- 文档中 base64Binary 的元素声明示例:<hehe > ab46c9d + </hehe >。

c) boolean 类型:

- 定义:用于规定 true 或 false 值;
- 注释:合法的布尔值是 true、false、1(表示 true)以及 0(表示 false);
- Schema 中逻辑声明示例:<xs:attribute name = "disabled" type = "xs:boolean"/>;
- 文档中的元素声明示例:<prize disabled = "true" >1 </prize >。

d) date 类型:

- 定义:用于定义日期,其格式为:"YYYY - MM - DD",其中:YYYY 表示年份(必需),MM 表示月份(必需),DD 表示日(必需);
- Schema 中日期声明示例:<xs:element name = "start" type = "xs:date"/>;
- 文档中的元素的声明示例:<start >2002 - 09 - 24 </start >。

e) dateTime 类型:

- 定义:用于定义日期和时间,其格式为:"YYYY - MM - DDThh:mm:ss. sss",其中:YYYY 表示年份(必需),MM 表示月份(必需),DD 表示日(必需),T 表示必需的时间部分的起始(必需),hh 表示小时(必需),mm 表示分钟(必需),ss 表示秒(必需),sss 表示毫秒数(可选);
- Schema 中日期时间声明示例:<xs:element name = "startdate" type = "xs:dateTime"/>;
- 文档中的元素声明示例:<startdate >2002 - 05 - 30T09:00:00 </startdate >。

f) dateTimeStamp 类型:

- 定义:派生自 dateTime,表示一个特定的日期时间类型,与 dateTime 的不同之处是:其表达式要求添加时区偏移表达式,合法时区偏移值为 - 14:00 ~ + 14:00 或者字符"Z";
- Schema 中日期时间声明示例:
  <xs:element name = "starttime" type = "xs: dateTimeStamp"/>;
- 文档中的 dateTimeStamp 元素声明示例:
  <starttime >2004 - 04 - 12T13:20:00 - 05:00 </starttime >。

g) decimal 类型:

- 定义:用于表示任意精度的小数。其格式是有限长度的十进制数字(0 ~ 9)序列,以点号(.)作为小数分隔符。数字前面可以添加" + "或" - "号,以表示正负;
- Schema 中 decimal 类型声明示例:<xs:element name = "price" type = "xs:decimal"/>;
- 文档中的元素示例:<price >999. 50 </price >。

h) integer 类型:

- 定义:派生自 decimal 类型,表示无限制的整数类型,即用于规定无小数部分的数值;
- Schema 中整数声明示例:<xs:element name = "prize" type = "xs:integer"/>;

- 文档中的元素示例：<prize>999</prize>。

i） long 类型：

- 定义：代表 64 位的有符号整数。最大值为 $2^{63}-1$，最小值为 -2^{63}；
- Schema 中整数声明示例：<xs:element name="price" type="xs:long"/>；
- 文档中的元素示例：<price>99999</price>。

j） int 类型：

- 定义：代表 32 位的有符号整数。最大值为 $2^{31}-1$，最小值为 -2^{31}；
- Schema 中整数声明示例：<xs:element name="price" type="xs:int"/>；
- 文档中的元素示例：<price>99954</price>。

k） short 类型：

- 定义：代表 16 位的有符号整数。最大值为 $2^{15}-1$，最小值为 -2^{15}；
- Schema 中整数声明示例：<xs:element name="price" type="xs:short"/>；
- 文档中的元素示例：<price>32767</price>。

l） byte 类型：

- 定义：代表 8 位的有符号整数。最大值为 $2^{7}-1$，最小值为 -2^{7}；
- Schema 中整数声明示例：<xs:element name="byteValue" type="xs:byte"/>；
- 文档中的元素示例：<byteValue>100</byteValue>。

m） nonNegativeInteger 类型：

- 定义：表示无限制的非负整数，即仅包含非负值的整数(0,1,2,...)，最小值为 0；
- Schema 中 nonNegativeInteger 声明示例：
  <xs:element name="age" type="xs:nonNegativeInteger"/>；
- 文档中的元素示例：<age>38</age>。

n） positiveInteger 类型：

- 定义：表示无限制的正整数，即仅包含正值的整数(1，2，..)，最小值为 1；
- Schema 中 positiveInteger 声明示例：
  <xs:element name="posValue" type="xs:positiveInteger"/>；
- 文档中的元素示例：<posValue>38</posValue>。

o） unsignedLong 类型：

- 定义：表示 64 位的无符号整数，即无正负的 64 位整数，最大值为 $2^{64}-1$，最小值为 0；
- Schema 中 unsignedLong 声明示例：
  <xs:element name="unsignedLongValue" type="xs:unsignedLong"/>；
- 文档中的元素示例：<unsignedLongValue>381</unsignedLongValue>。

p） unsignedInt 类型：

- 定义：表示 32 位的无符号整数，即无正负的 32 位整数，最大值为 $2^{32}-1$，最小值为 0；
- Schema 中 unsignedInt 声明示例：
  <xs:element name="unsignedIntValue" type="xs:unsignedInt"/>；
- 文档中的元素示例：<unsignedIntValue>381</unsignedIntValue>。

q） unsignedShort 类型：

- 定义：表示 16 位的无符号整数，即无正负的 16 位整数，最大值为 $2^{16}-1$，最小值为 0；
- Schema 中 unsignedShort 声明示例：
  <xs:element name="unsignedShortValue" type="xs:unsignedShort"/>；
- 文档中的元素示例：<unsignedShortValue>381</unsignedShortValue>。

r） unsignedByte 类型：

- 定义：表示 8 位的无符号整数，即无正负的 8 位整数，最大值为 2^8-1，最小值为 0；
- Schema 中 unsignedByte 声明示例：
  <xs:element name="unsignedByteValue" type="xs:unsignedByte"/>；
- 文档中的元素示例：<unsignedByteValue>181</unsignedByteValue>。

s) nonPositiveInteger 类型：

- 定义：表示无限制的非正整数，即仅包含非正值的整数(...，-2，-1，0)，最大值为 0；
- Schema 中 nonPositiveInteger 声明示例：
  <xs:element name="nonPosValue" type="xs:nonPositiveInteger"/>；
- 文档中的元素示例：<nonPosValue>-181</nonPosValue>。

t) negativeInteger 类型：

- 定义：表示无限制的负整数，即仅包含负值的整数 (...，-2，-1)，最大值为 -1；
- Schema 中 negativeInteger 声明示例：
  <xs:element name="negValue" type="xs:negativeInteger"/>；
- 文档中的元素示例：<negValue>-181</negValue>。

u) double 类型：

- 定义：用于代表 64 位的双精度浮点数，double 值的格式是一个尾数(应是 decimal 类型的数字)，其后可跟字符“e”(大小写均可)，“e”后面跟一个指数(应是整数)；
- Schema 中双精度浮点数的声明示例：<xs:element name="double" type="xs:double"/>；
- 文档中的元素示例：<double>5.6e3</double>。

v) duration 类型：

- 定义：用于表示持续时间。其格式为：PnYnMnDTnHnMnS，其中：P 表示周期(必需)，nY 表示年数，nM 表示月数，nD 表示天数，T 表示时间部分的起始(如果要规定小时、分钟和秒，则此选项为必需)，nH 表示小时数，nM 表示分钟数，nS 表示秒数；
- Schema 中持续时间声明示例：<xs:element name="period" type="xs:duration"/>；
- 文档中的元素示例：<period>P5Y</period>。

w) dayTimeDuration 类型：

- 定义：派生自 Duration，其格式为：PnDTnHnMnS，其中：P 是固定的，D、H、M、S 分别代表日、时、分、秒，且 D、H、M 前面的 n 是整数，S 前面的 n 可以有小数部分；
- Schema 中 dayTimeDuration 类型声明示例：
  <xs:element name="period" type="xs:dayTimeduration"/>；
- 文档中的 dayTimeDuration 元素声明示例：<period>P1DT2H</period><!—1 天 2 小时- - >。

x) yearMonthDuration 类型：

- 定义：派生自 Duration，其格式为：PnYnM，其中：P 是固定的，Y、M 分别代表年、月，且其前面的 n 是整数；
- Schema 中 yearMonthDuration 声明示例：
  <xs:element name="period" type="xs:yearMonthDuration"/>；
- 文档中的 yearMonthDuration 元素声明示例：<period>P2Y6M</period><!—2 年 6 个月- - >。

y) float 类型：

- 定义：表示 IEEE 单精度 32 位浮点数；
- Schema 中 float 数据类型的声明示例：

<xs:element name="planetQuality" type="xs:float"/>;

- 文档中的元素示例：<planetQuality>1.888e29</planetQuality>。

z) gDay 类型：

- 定义：表示每月都出现的特定的一天，其格式为：---DD，前面的三个连字符是必需的；
- Schema 中 gDay 类型的声明示例：<xs:element name="myDay" type="xs:gDay"/>；
- 在文档中 gDay 元素的声明示例：<myDay>---06</myDay><!--每月的6日-->。

aa) gMonth 类型：

- 定义：表示每年都出现的特定月，其格式为：--MM，前面的两个连字符是必需的；
- Schema 中 gMonth 类型的声明示例：<xs:element name="myMonth" type="xs:gMonth"/>；
- 在文档中 gMonth 元素的声明示例：<myMonth>--05<myMonth/><!--5月-->。

bb) gMonthDay 类型：

- 定义：表示每年都出现的特定的一天其格式为：--MM-DD，前面的两个连字符是必需的；
- Schema 中 gMonthDay 类型的声明示例：
  <xs:element name="myMonth" type="xs:gMonthDay"/>；
- 在文档中 gMonthDay 元素的声明示例：<myMonthDay>--06-04</myMonthDay><!--每年的6月4日-->。

cc) gYear 类型：

- 定义：表示一个特定的阳历(gregorian calender)年，其格式为 YYYY，要表示 9999 之后的年，可以在年值左边添加数字；要表示公元前的年，可以在年值前面添加减号(-)；
- Schema 中 gYear 类型的声明示例：<xs:element name="myYear" type="xs:gYear"/>；
- 文档中 gYear 元素的声明示例：<myYear>-0050</myYear><!--公元前50年-->。

dd) gYearMonth 类型：

- 定义：表示特定的年中的特定月其格式为 YYYY-MM，其中"YYYY"和"MM"都是必需的。要表示 9999 之后的年，可以在年值左边添加数字，要表示公元前的年，可以在年值前面添加减号(-)；
- Schema 中 gYearMonth 类型的声明示例：
  <xs:element name="myYearMonth" type="xs:gYearMonth"/>；
- 在文档中 gYearMonth 元素的声明示例：<myYearMonth>1988-05</myYearMonth><!--1988年5月-->。

ee) hexBinary 类型：

- 定义：表示的是以十六进制数字保存的二进制数据，其值只能由 0~9、a~f 和 A~F(代表 10~15)组成。其长度为偶数；
- Schema 中 hexBinary 声明示例：<xs:element name="blobsrc" type="xs:hexBinary"/>；
- 文档中 hexBinary 的元素声明示例：<blobsrc>1f12fa12aefa</blobsrc>。

ff) NOTATION 类型：

- 定义：描述 XML 文档中非 XML 数据的格式，其父元素为 Schema；

- 语法：

```
<notationid = IDname = NCNamepublic = anyURIsystem = anyURIany attributes >
                    (annotation?)
</notation >
```

- Schema 中 NOTATION 的声明示例：

```
<? xml version = "1.0"? >
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema" >
    <xs:notation name = "gif" public = "image/gif" system = "view.exe"/ >
    <xs:notation name = "jpeg" public = "image/jpeg" system = "view.exe"/ >
    <xs:element name = "image" >
        <xs:complexType >
            <xs:attribute name = "type" use = "required" >
                <xs:simpleType >
                    <xs:restriction base = "xs:NOTATION" >
                        <xs:enumeration value = "gif"/ >
                        <xs:enumeration value = "jpeg"/ >
                    </xs:restriction >
                </xs:simpleType >
            </xs:attribute >
        </xs:complexType >
    </xs:element >
</xs:schema >
```

- 文档中 NOTATION 的声明示例：<image type = "gif" > </image >。

gg） QName 类型：

- 定义：表示 XML 名称空间限定名，其值包括一个名称空间前缀和一个本地部分，中间以冒号(:)分隔，这两部分都是 NCName 类型。QName 类型的值不能以冒号开头；
- Schema 中 QName 的声明示例：

  <myQName xmlns:bks = "http://www.xml.org/books" >bks:book </myQName >；
- 文档中 QName 元素的声明示例：<myQName >book </myQName >。

hh） string 类型：

- 定义：字符串数据类型可包含字符、换行、回车以及制表符；
- Schema 中字符串声明示例：<xs:element name = "customer" type = "xs:string"/ >；
- 文档中的元素示例：<customer >John Smith </customer >。

ii） normalizedString 类型：

- 定义：规格化字符串数据类型源自于字符串数据类型。规格化字符串数据类型同样可包含字符，但是 XML 处理器会移除换行，回车以及制表符；
- Schema 中规格化字符串数据类型声明示例：

  <xs:element name = "customer" type = "xs:normalizedString"/ >；
- 文档中的元素声明示例：<customer >John Smith </customer >。

jj） token 类型：

- 定义：源自于字符串数据类型。Token 数据类型同样可包含字符，但是 XML 处理器会移除换行符、回车、制表符、开头和结尾的空格以及(连续的)空格；
- Schema 中有关 token 声明示例：<xs:element name = "customer" type = "xs:token"/ >；

- 文档中的元素声明示例：<customer>John Smith</customer>。

kk） language 类型：

- 定义：符合[a-zA-Z]{1,8}(-[a-zA-Z0-9]{1,8})*样式的自然语言标识符，通常用于表示语言的文档或文档的一部分，如 zh，en 等；
- Schema 中 language 类型的声明示例：
  <xs:element name="speaking" type="xs:language"/>；
- 文档中 language 元素的声明示例：<speaking>en</speaking><!--English-->。

ll） Name 类型（名称字符串类型）：

- 定义：派生自 token，它表示一个 XML 名称，即该类型的值应以字母、下划线（_）或者冒号（:）开头，而且只能包含字母、数字、连字符（-）、下划线（_）、冒号（:）或者句号（.）；
- Schema 中有关 Name 声明示例：<xs:element name="name" type="xs:Name"/>；
- 文档中元素声明示例：<name>_book</name>。

mm） NCName 类型（无冒号名称字符串类型）：

- 定义：继承自 Name，表示一个 XML 无冒号的名称，即该名称中不能使用冒号。NCName 类型的值应以字母或者下划线（_）开头，而且只能包含字母、数字、连字符（-）、下划线（_）和句号（.）。NCName 常用于带名称空间限定的限定名的本地部分；
- Schema 中有关 NCName 声明示例：<xs:element name="name" type="xs:NCName"/>；
- 文档中元素声明示例：<name>_computer.book</name>。

nn） ENTITY 类型：

- 定义：该属性值是一个未解析实体，例如图片文件，声音文件。（该类型有部分浏览器无法解析）。

oo） ID 类型：

- 定义：ID 类型的属性应包含一个有效的 XML 名称，而且该名称在文档中是独一无二的，即 ID 属性可为元素分配一个唯一的标识符。它应以字符或者下划线开头，并且只能包含字母、数字、下划线、连字符等；
- Schema 中 ID 类型的声明示例：
  <xs:attribute name="shipID" type="xs:ID" use="required" />；
- 文档中 ID 元素的声明示例：<shipping shipID="ID_1"/>。

pp） IDREF 类型：

- 定义：该属性值应引用自另一个已有的 ID 属性值；
- Schema 中 IDREF 类型的声明示例：<xs:attribute name="shippedBy" type="xs:IDREF" use="required"/>；
- 文档中 IDREF 元素的声明示例：<book shippedBy="ID_1"/><!--其中 ID_1 是已声明过的 ID 类型的值-->。

qq） NMTOKEN 类型（名称记号字符串类型）：

- 定义：该属性值应是一个合法的 XML 名称，同时指定了该属性值是字符串数据。限制 token 类型中只能包含数字，字母，下划线，冒号，及其他名字字符（可以由数字开头）；
- Schema 中 NMTOKEN 类型的声明示例：<xs:element name="NMTOKEN" type="xs:NMTOKEN" />；
- 文档中 NMTOKEN 元素的声明示例：<NMTOKEN>:1234</NMTOKEN>。

rr） time 类型：

- 定义：用于定义时间。其格式为："hh：mm：ss"，其中 hh 表示小时，mm 表示分钟，ss 表示秒；
- 注释："hh"、"mm"和"ss"都是必需的；
- Schema 中时间声明示例：<xs：element name = "start" type = "xs：time"/>；
- 文档中的元素声明示例：<start>09：00：00</start>。

附 录 B
（规范性附录）
数据交换包的 XML Schema

```
<? xml version = "1.0" encoding = "UTF -8" ? >
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema" elementFormDefault = "qualified" attributeFormDefault = "unqualified" >
    <xs:element name = "DataSharingPacket" >
        <xs:annotation >
            <xs:documentation >共享数据包 </xs:documentation >
        </xs:annotation >
        <xs:complexType >
            <xs:sequence >
              <xs:element ref = "DataPacketHeader" minOccurs = "0" / >
              <xs:element ref = "DataPacketBody" / >
            </xs:sequence >
        </xs:complexType >
    </xs:element >
    <xs:element name = "DataPacketHeader" >
        <xs:annotation >
            <xs:documentation >数据包头 </xs:documentation >
        </xs:annotation >
        <xs:complexType >
            <xs:sequence >
                <xs:element ref = "DataPacketID" / >
                <xs:element ref = "DataPacketFunction" / >
                <xs:element ref = "DataPacketRefID" / >
                <xs:element ref = "SenderID" / >
                <xs:element ref = "RecipientID" maxOccurs = "unbounded" / >
                <xs:element ref = "DataPacketGenDate" / >
                <xs:element ref = "DataPacketGenTime" / >
                <xs:element ref = "DataSigID" minOccurs = "0" / >
                <xs:element ref = "DataSigDesc" minOccurs = "0" / >
                <xs:element ref = "DataEncID" minOccurs = "0" / >
                <xs:element ref = "DataEncMethod" minOccurs = "0" / >
                <xs:element ref = "EncDataDesc" minOccurs = "0" / >
                <xs:element ref = "ComDataID" minOccurs = "0" / >
                <xs:element ref = "ComDataMethod" minOccurs = "0" / >
                <xs:element ref = "ComDataDesc" minOccurs = "0" / >
                <xs:element ref = "ExProtocolVersionNum" / >
                <xs:element ref = "MessageSequeNum" / >
```

```
                <xs:element ref="DataPacketsAmount" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="DataPacketID">
        <xs:annotation>
            <xs:documentation>数据包标识</xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="NameOfDataPacketID"/>
                <xs:element ref="DataPacketVersion" minOccurs="0"/>
                <xs:element ref="ControllingAgency" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="NameOfDataPacketID">
        <xs:annotation>
            <xs:documentation>数据包标识名称</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="70"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="DataPacketVersion">
        <xs:annotation>
            <xs:documentation>数据包版本</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="9"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="ControllingAgency">
        <xs:annotation>
            <xs:documentation>控制机构</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="70"/>
```

```
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="DataPacketFunction">
        <xs:annotation>
            <xs:documentation>数据包功能</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="DataPacketRefID">
        <xs:annotation>
            <xs:documentation>数据包参考号</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="35"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="SenderID">
        <xs:annotation>
            <xs:documentation>发送方标识</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="70"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="RecipientID">
        <xs:annotation>
            <xs:documentation>接收方标识</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="70"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
```

```
<xs:element name="DataPacketGenDate" type="xs:date">
    <xs:annotation>
        <xs:documentation>数据包生成日期</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="DataPacketGenTime" type="xs:time">
    <xs:annotation>
        <xs:documentation>数据包生成时间</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="DataSigID" type="xs:boolean">
    <xs:annotation>
        <xs:documentation>数字签名标识</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="DataSigDesc">
    <xs:annotation>
        <xs:documentation>签名数据描述</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="100"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="DataEncID" type="xs:boolean">
    <xs:annotation>
        <xs:documentation>数据加密标识</xs:documentation>
    </xs:annotation>
</xs:element>
<xs:element name="DataEncMethod">
    <xs:annotation>
        <xs:documentation>数据加密方法</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="16"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="EncDataDesc">
    <xs:annotation>
        <xs:documentation>加密数据描述</xs:documentation>
```

```
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="100"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="ComDataID" type="xs:boolean">
        <xs:annotation>
            <xs:documentation>压缩数据标识</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="ComDataMethod">
        <xs:annotation>
            <xs:documentation>数据压缩方法</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="16"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="ComDataDesc">
        <xs:annotation>
            <xs:documentation>压缩数据描述</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="100"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="ExProtocolVersionNum">
        <xs:annotation>
            <xs:documentation>交换协议版本号</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="70"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="MessageSequeNum">
```

```
            <xs:annotation>
                <xs:documentation>消息序列号</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:maxLength value="70"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="DataPacketsAmount">
            <xs:annotation>
                <xs:documentation>数据包总数量</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
                <xs:restriction base="xs:int">
                    <xs:minInclusive value="1" />
                    <xs:maxInclusive value="999999999" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="DataPacketBody">
            <xs:annotation>
                <xs:documentation>数据包体</xs:documentation>
            </xs:annotation>
            <xs:complexType>
                <xs:sequence>
                    <xs:element ref="DataItem" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
        <xs:element name="DataItem" type="xs:anyType">
            <xs:annotation>
                <xs:documentation>数据项</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:schema>
```

附 录 C
（资料性附录）
数据交换包的 XML 格式应用实例

C.1 应用实例说明

应用 XML 格式数据交换包的集装箱堆场“运输设备细目”数据信息示例说明，见图 C.1。

报文开始：
发送方：Qingdao Qianwan Container Terminal Co. . Ltd（青岛前湾集装箱码头有限责任公司）；
接收方：KMT（高丽海运）；
报文生成日期：2012 年 10 月 1 日；
报文生成时间：14：20：00。

设备细目：
箱号：KMTU789456；
空重：重箱；
尺寸类型：20G0；
进场日期：2012 年 9 月 19 日 2：23：15；
堆场位置：E1735E1；
提单号：KMTCHIJ003398。

图 C.1 应用 XML 格式数据交换包示例

C.2 应用实例

```
<? xml version = "1.0" encoding = "UTF - 8" ? >
< DataSharingPacket xmlns:xsi = "http://www. w3. org/2001/XMLSchema - instance"
                    xsi:noNamespaceSchemaLocation = "DataSharingPacket. xsd" >
    < DataPacketHeader >
        < DataPacketID >
            < NameOfDataPacketID > 数据交换包 </NameOfDataPacketID >
        </DataPacketID >
        <! - - 数据包功能描述 - - >
        < DataPacketFunction > 应答 </DataPacketFunction >
        < DataPacketRefID > 1 </DataPacketRefID >
        <! - - 发送方标识 - - >
        < SenderID > QQCT </SenderID >
        <! - - 接收方标识 - - >
        < RecipientID > KMT </RecipientID >
        <! - - 数据包生成日期/时间 - - >
        < DataPacketGenDate > 2012 - 10 - 01 </DataPacketGenDate >
        < DataPacketGenTime > 14:20:00 </DataPacketGenTime >
        <! - - 数字签名标识，这里为 true，表示对当中的数据进行了签名处理。 - - >
        < DataSigID > true </DataSigID >
```

```
        <！ - -对被签名数据进行简要描述 - - >
        < DataSigDesc > 对 XML 格式的设备细目数据进行了数字签名。 </DataSigDesc >
        <！ - -数字签名标识,这里为 true,表示对当中的数据进行了签名处理。 - - >
        < DataEncID > true </DataEncID >
        <！ - -采用的加密方法。 - - >
        < DataEncMethod > 3DES </DataEncMethod >
        <！ - -对所加密的数据进行的简要描述。 - - >
        < EncDataDesc > 对已签名的 XML 格式的设备细目数据进行了加密。 </EncDataDesc >
        <！ - -数字压缩标识,这里为 false,表示对当中的数据不进行压缩处理。 - - >
        < ComDataID > false </ComDataID >
        <！ - -对交换协议版本号进行的简要描述 - - >
        < ExProtocolVersionNum > 0x01 0x02 0x0F </ExProtocolVersionNum >
        <！ - -对消息序列号进行的简要描述 - - >
        < MessageSequeNum > 1 </MessageSequeNum >
        <！ - -数据包总数量 - - >
        < DataPacketsAmount > 1 </DataPacketsAmount >
    </DataPacketHeader >
    < DataPacketBody >
        < DataItem >
            < EquipmentIDNumber > KMTU789456 </EquipmentIDNumber >
            < FullOrEmptyDescription > 重箱 </FullOrEmptyDescription >
            < EquipmentSizeAndTypeCode > 20G0 </EquipmentSizeAndTypeCode >
            < DateValue > 20120919 2:23:15 </DateValue >
            < LocationCode > E1735E1 </LocationCode >
            < ReferenceNumber > KMTCHIJ003398 </ReferenceNumber >
        </DataItem >
    </DataPacketBody >
</DataSharingPacket >
```

C.3 签名及加密应用实例

```
<？ xml version = "1.0" encoding = "UTF - 8" ？ >
< DataSharingPacket xmlns:xsi = "http://www. w3. org/2001/XMLSchema - instance"
                    xsi:noNamespaceSchemaLocation = "DataSharingPacket. xsd" >
    < DataPacketHeader >
        < DataPacketID >
          < NameOfDataPacketID > 数据交换包 </NameOfDataPacketID >
        </DataPacketID >
        <！ - -数据包功能描述 - - >
        < DataPacketFunction > 应答 </DataPacketFunction >
        < DataPacketRefID > 1 </DataPacketRefID >
        <！ - -发送方标识 - - >
        < SenderID > QQCT </SenderID >
        <！ - -接收方标识 - - >
```

```
    <RecipientID>KMT</RecipientID>
    <!--数据包生成日期/时间-->
    <DataPacketGenDate>2012-10-01</DataPacketGenDate>
    <DataPacketGenTime>14:20:00</DataPacketGenTime>
    <!--数字签名标识,这里为true,表示对当中的数据进行了签名处理。-->
    <DataSigID>true</DataSigID>
    <!--对被签名数据进行简要描述-->
    <DataSigDesc>对XML格式的设备细目数据进行了数字签名。</DataSigDesc>
    <!--数字签名标识,这里为true,表示对当中的数据进行了签名处理。-->
    <DataEncID>true</DataEncID>
    <!--采用的加密方法。-->
    <DataEncMethod>3DES</DataEncMethod>
    <!--对所加密的数据进行的简要描述。-->
    <EncDataDesc>对已签名的XML格式的设备细目数据进行了加密。</EncDataDesc>
    <!--数字压缩标识,这里为false,表示发送之前不对数据进行压缩处理。-->
    <ComDataID>fasle</ComDataID>
    <!--对交换协议版本号进行的简要描述-->
    <ExProtocolVersionNum>0x01 0x02 0x0F</ExProtocolVersionNum>
    <!--对消息序列号进行的简要描述-->
    <MessageSequeNum>1</MessageSequeNum>
    <!--数据包总数量-->
    <DataPacketsAmount>1</DataPacketsAmount>
</DataPacketHeader>
<DataPacketBody>
        <!--以下是数据加密后生成的XML片段-->
        <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
                            Type="http://www.w3.org/2001/04/xmlenc#Content">
            <xenc:EncryptionMethod
                Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
            <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <xenc:EncryptedKey
                        xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
                    <xenc:EncryptionMethod
                Algorithm="http://www.w3.org/2001/04/xmlenc#kw-tripledes"/>
                    <xenc:CipherData>
                    <xenc:CipherValue>
                        b5xGBUgfB3VGfXk5HptVT3X5a/hzJ+i…
                    </xenc:CipherValue>
                    </xenc:CipherData>
                        </xenc:EncryptedKey>
                    </ds:KeyInfo>
                    <xenc:CipherData>
                        <xenc:CipherValue>
```

```
                                UdUgTfWtW6i4CYfVJ4hpaGDWLpB9zB8xtLIPA1Zkn…
                            </xenc:CipherValue>
                        </xenc:CipherData>
                    </xenc:EncryptedData>
                </DataPacketBody>
            </DataSharingPacket>
```

附 录 D
（资料性附录）
基于 Web Service 方式的数据交换包的 XML 格式应用实例

D.1 应用实例说明

数据发送方针对数据请求方返回的数据信息示例说明，见图 D.1。

设备细目：
箱号：KMTU789456；
空重：重箱；
尺寸类型：20G0；
进场日期：2012 年 9 月 19 日 2：23：15；
堆场位置：E1735E1；
提单号：KMTCHIJ003398。

图 D.1 数据发送方针对数据请求方返回的数据信息示例

D.2 应用实例

```
< DataSharingPacket xmlns:xsi = " http://www. w3. org/2001/XMLSchema - instance"
                    xsi:noNamespaceSchemaLocation = " DataSharingPacket. xsd" >
    < DataPacketBody >
        < DataItem >
            < EquipmentIDNumber > KMTU789456 </EquipmentIDNumber >
            < FullOrEmptyDescription > 重箱 </FullOrEmptyDescription >
            < EquipmentSizeAndTypeCode > 20G0 </EquipmentSizeAndTypeCode >
            < DateValue > 20120919 2:23:15 </DateValue >
            < LocationCode > E1735E1 </LocationCode >
            < ReferenceNumber > KMTCHIJ003398 </ReferenceNumber >
        </DataItem >
    </DataPacketBody >
</DataSharingPacket >
```

D.3 签名及加密应用实例

```
< DataSharingPacket xmlns:xsi = " http://www. w3. org/2001/XMLSchema - instance"
    xsi:noNamespaceSchemaLocation = " DataSharingPacket. xsd" >
    < DataPacketBody >
        < xenc:EncryptedData xmlns:xenc = " http://www. w3. org/2001/04/xmlenc#"
                             Type = " http://www. w3. org/2001/04/xmlenc#Content" >
        < xenc:EncryptionMethod
            Algorithm = " http://www. w3. org/2001/04/xmlenc#aes128 - cbc"
            xmlns:xenc = " http://www. w3. org/2001/04/xmlenc#" / >
    < ds:KeyInfo xmlns:ds = " http://www. w3. org/2000/09/xmldsig#" >
```

```
            <xenc:EncryptedKey xmlns:xenc = "http://www.w3.org/2001/04/xmlenc#" >
                <xenc:EncryptionMethod
                        Algorithm = "http://www.w3.org/2001/04/xmlenc#kw - tripledes"
                        xmlns:xenc = "http://www.w3.org/2001/04/xmlenc#" / >
                <xenc:CipherData xmlns:xenc = "http://www.w3.org/2001/04/xmlenc#" >
                        <xenc:CipherValue
                            xmlns:xenc = "http://www.w3.org/2001/04/xmlenc#" >
                            5Tsf3AAGNENGyDpSeDQrsI…
                        </xenc:CipherValue >
                </xenc:CipherData >
            </xenc:EncryptedKey >
        </ds:KeyInfo >
        <xenc:CipherData xmlns:xenc = "http://www.w3.org/2001/04/xmlenc#" >
            <xenc:CipherValue xmlns:xenc = "http://www.w3.org/2001/04/xmlenc#" >
                sdrD4/5HMOXOxPPELU7Wt6Ilu4mluBga9nI/J7lgtBo7ZgXhO6gvysv6yft3…
            </xenc:CipherValue >
          </xenc:CipherData >
      </xenc:EncryptedData >
      </DataPacketBody >
  </DataSharingPacket >
```

附 录 E
（资料性附录）
XML 加密粒度说明应用实例

E.1 应用实例说明

XML 加密规范按照加密粒度的不同进行了分类，使用户根据需要选择不同的加密粒度，下面给出了一个 XML 示例文档，该文档描述了支付信息，通过选择不同的加密密度对该文档进行加密，对几种不同的加密粒度进行说明。

```
<? xml version = '1.0'? >
<PaymentInfo xmlns = 'http://example.org/paymentv2' >
    <Name > John Smith </Name >
    <CreditCard Limit = '5,000' Currency = 'USD' >
        <Number > 4019 2445 0277 5567 </Number >
        <Issuer > Example Bank </Issuer >
        <Expiration > 04/02 </Expiration >
    </CreditCard >
</PaymentInfo >
```

E.2 应用实例

E.2.1 加密 XML 元素

```
<? xml version = '1.0'? >
<PaymentInfo xmlns = 'http://example.org/paymentv2' >
    <Name > John Smith </Name >
    <EncryptedData Type = 'http://www.w3.org/2001/04/xmlenc#Element'
            xmlns = 'http://www.w3.org/2001/04/xmlenc#' >
        <CipherData >
            <CipherValue > A23B45C56 </CipherValue >
        </CipherData >
    </EncryptedData >
</PaymentInfo >
```

E.2.2 加密 XML 元素的内容——<CreditCard > 的子元素

```
<? xml version = '1.0'? >
<PaymentInfo xmlns = 'http://example.org/paymentv2' >
    <Name > John Smith </Name >
    <CreditCard Limit = '5,000' Currency = 'USD' >
        <EncryptedData xmlns = 'http://www.w3.org/2001/04/xmlenc#'
            Type = 'http://www.w3.org/2001/04/xmlenc#Content' >
        <CipherData >
            <CipherValue > A23B45C56 </CipherValue >
```

```
            </CipherData>
            </EncryptedData>
        </CreditCard>
    </PaymentInfo>
```

E.2.3 加密 XML 元素的内容——<Number>的字符数据

```
<?xml version='1.0'?>
<PaymentInfo xmlns='http://example.org/paymentv2'>
    <Name>John Smith</Name>
    <CreditCard Limit='5,000' Currency='USD'>
        <Number>
            <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
                Type='http://www.w3.org/2001/04/xmlenc#Content'>
                <CipherData>
                    <CipherValue>A23B45C56</CipherValue>
                </CipherData>
            </EncryptedData>
        </Number>
        <Issuer>Example Bank</Issuer>
        <Expiration>04/02</Expiration>
    </CreditCard>
</PaymentInfo>
```

E.2.4 加密 XML 文档

```
<?xml version='1.0'?>
<EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'MimeType='text/xml'>
    <CipherData>
        <CipherValue>A23B45C56</CipherValue>
    </CipherData>
</EncryptedData>
```

E.2.5 多重加密

E.2.5.1 加密过的文件

```
<pay:PaymentInfo xmlns:pay='http://example.org/paymentv2'>
  <EncryptedData Id='ED1' xmlns='http://www.w3.org/2001/04/xmlenc#'
      Type='http://www.w3.org/2001/04/xmlenc#Element'>
      <CipherData>
          <CipherValue>originalEncryptedData</CipherValue>
      </CipherData>
  </EncryptedData>
</pay:PaymentInfo>
```

E.2.5.2 加密过的文件的再加密

```
<pay:PaymentInfo xmlns:pay='http://example.org/paymentv2'>
```

```
    <EncryptedData Id = 'ED2' xmlns = 'http://www.w3.org/2001/04/xmlenc#'
            Type = 'http://www.w3.org/2001/04/xmlenc#Element' >
        <CipherData >
            <CipherValue > newEncryptedData </CipherValue >
        </CipherData >
    </EncryptedData >
</pay:PaymentInfo >
```

参 考 文 献

[1] GB/T 7408—2005 数据元和交换格式 信息交换 日期和时间表示法.
[2] GB/T 18793—2002 信息技术可扩展置标语言(XML)1.0.

中华人民共和国
交通运输行业标准
交通运输信息系统基于XML的数据交换通用规则
JT/T 1021—2016
*
人民交通出版社股份有限公司出版发行
（100011 北京市朝阳区安定门外外馆斜街3号）
各地新华书店经销
北京市密东印刷有限公司印刷
*
开本：880×1230 1/16 印张：2.5 字数：68千
2016年4月 第1版
2016年4月 第1次印刷
*
统一书号：15114·2367 定价：30.00元