ICS 11.020
CCS C 07

WS

中 华 人 民 共 和 国 卫 生 行 业 标 准

WS/T 790.3—2021

区域卫生信息平台交互标准
第 3 部分：节点验证服务

Regional health information platform interactive standard—
Part 3: Node authentication service

2021-10-27 发布　　2022-04-01 实施

中华人民共和国国家卫生健康委员会　发布

前　言

本标准是WS/T 790《区域卫生信息平台交互标准》的第3部分。WS/T 790已经发布以下部分：

——第1部分：总则；
——第2部分：时间一致性服务；
——第3部分：节点验证服务；
——第4部分：安全审计服务；
——第5部分：基础通知服务；
——第6部分：居民注册服务；
——第7部分：医疗卫生机构注册服务；
——第8部分：医疗卫生人员注册服务；
——第9部分：术语注册服务；
——第10部分：健康档案存储服务；
——第11部分：健康档案管理服务；
——第12部分：健康档案采集服务；
——第13部分：健康档案调阅服务；
——第14部分：文档订阅发布服务；
——第15部分：预约挂号服务；
——第16部分：双向转诊服务；
——第17部分：签约服务；
——第18部分：提醒服务。

本标准由国家卫生健康标准委员会卫生健康信息标准专业委员会负责技术审查和技术咨询，由国家卫生健康委统计信息中心负责协调性和格式审查，由国家卫生健康委规划发展与信息化司负责业务管理、法规司负责统筹管理。

本标准起草单位：国家卫生健康委统计信息中心、湖南省卫生计生委信息统计中心、国家电子计算机质量监督检验中心。

本标准主要起草人：胡建平、李岳峰、许德俊、叶彦波、雷永贵、郑良。

区域卫生信息平台交互标准
第3部分：节点验证服务

1 范围

本标准规定了基于健康档案的区域卫生信息平台的交互信息的节点验证规则。

本标准适用于对基于健康档案的区域卫生信息平台的服务访问和消息传输。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本标准必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本标准；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本标准。

ITU-T X.509(03/00) 信息技术-开放系统互联-目录：公共密钥和属性凭证框架

ISO/ITU-T ASN.1抽象语法标记（Abstract Syntax Notation One）

WS/T 448 基于居民健康档案的区域卫生信息平台技术规范

WS/T 482 卫生信息共享文档编制规范

3 术语和略缩语

3.1 术语和定义

WS/T 448、WS/T 482界定的及下列术语和定义适用于本标准。

3.1.1

安全节点 Security node

节点指拥有自己唯一网络地址的设备或服务模块,具有传送或接收数据功能。包括工作站、客户端、网络用户、个人计算机、服务器、打印机和其他网络连接的设备。安全节点指被授权且经过验证的节点。

3.1.2

网络访问点 Access Point

指网络上可访问的节点。

3.2 缩略语

下列缩略语适用于本标准。

DICOM：医学数字影像和通讯（Digital Imaging and Communications in Medicine）

HTTP：超文本传输协议（HyperText Transfer Protocol）

HL7：卫生信息第7层协议（Health Level Seven）

NA：节点验证（Node Authentication）

NAS：节点验证服务器(Node Authentication Server)

SN：安全节点（Security Node）

4 角色

4.1 角色定义

节点验证包括以下角色：

——节点验证服务(NAS)：提供节点验证服务；

——安全节点(SN)：在网络上的两个节点之间建立信任关系，建立一个用户身份，授权对节点处数据和应用的访问。

4.2 角色的交易关系

与节点验证服务直接相关的角色与角色间的交易关系见图1。

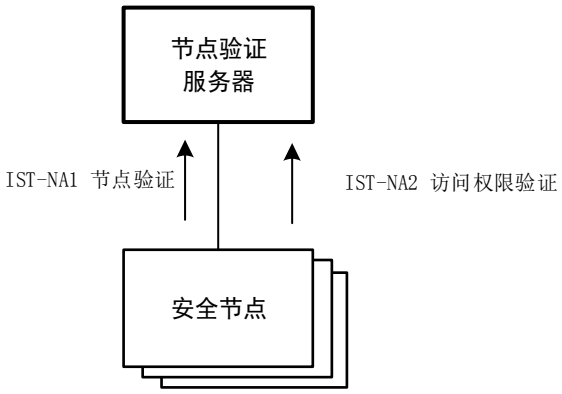

图1 节点验证服务角色图

4.3 角色的交易可选性

与角色相关的交易见表1。如果声明支持该交互标准，则应实现标准中指定为“R”的交易。

表1 节点验证服务交互标准-角色和交易

角色	交易	编号	可选项
节点验证服务器(NAS)	节点验证	IST-NA1	R
	访问权限验证	IST-NA2	O
安全节点(SN)	节点验证	IST-NA1	R
	访问权限验证	IST-NA2	O

5 交易

5.1 节点验证

5.1.1 用例

验证节点的用例见图2。

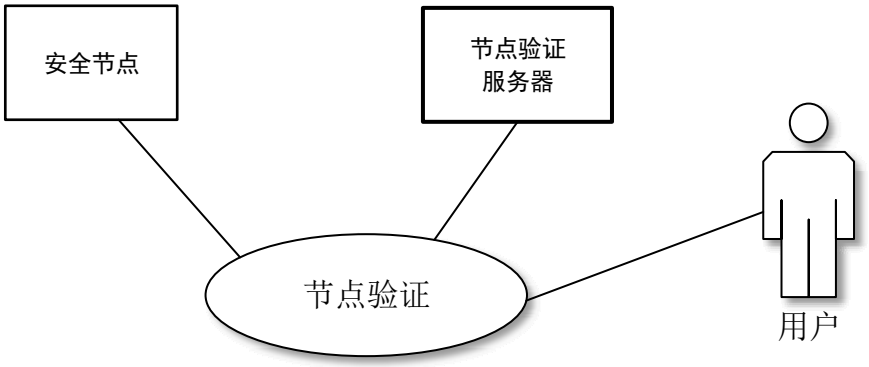

图2　节点验证服务用例图

当触发事件发生时，由安全节点向节点验证服务请求验证节点。

5.1.2　交易流程

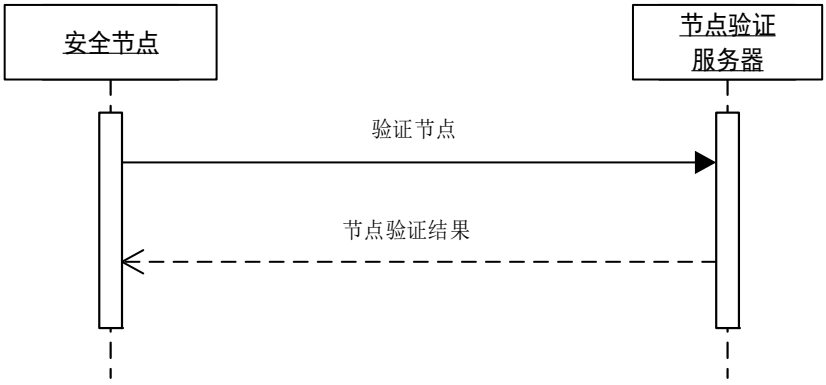

图3　节点验证交易流程图

5.1.3　消息请求

5.1.3.1 触发事件

当本地安全节点与远程安全节点之间想要进行信息交换时，触发此交易。

基本安全节点对每个DICOM、HTTP或HL7连接总是应用验证节点过程。

5.1.3.2 消息结构及约束

消息结构应符合附录B节点验证服务消息格式Authenticate元素构件要求，消息补充约束见表2。

表2　节点验证请求消息的消息结构

节点	基数	可选项	节点说明	对应数据元标识符
/NodeAuthenticate	1..1	R	发送消息	
/NodeAuthenticate/id	1..1	R	OID类型，节点唯一标识。	
/NodeAuthenticate/digestValue	1..1	R	消息摘要	
/NodeAuthenticate/signatureValue	1..1	R	消息摘要的签名	

5.1.4 消息应答

5.1.4.1 触发事件

当节点验证服务器接收到验证请求时，触发该消息应答。

5.1.4.2 消息结构及约束

消息结构应符合附录B节点验证服务消息格式AuthenticateResponse元素构件要求，消息补充约束见表3。

表3 节点验证请求应答的消息补充约束

节点	基数	可选项	节点说明	对应数据元标识符
/NodeAuthenticateResponse	1..1	R	节点验证应答消息	

5.1.5 消息语义

验证节点交易应为代表节点身份的证书交换。这些证书用于验证节点，通知授权和审计日志。证书要求如下：

——节点验证采用 ITU-T X.509(03/00)证书，证书结构见附录 C；

——证书签名算法可选用 C.1 所列。

5.2 访问权限验证

5.2.1 用例

访问权限管理的用例见图4。

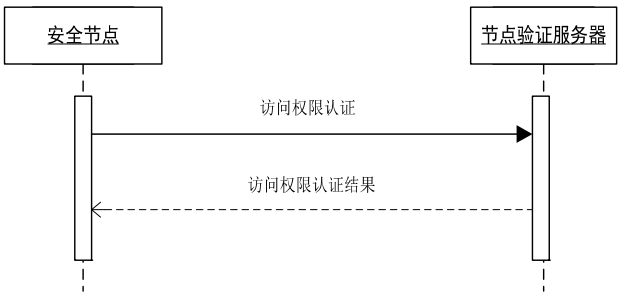

图4 访问权限验证用例图

当触发事件发生时，由安全节点向节点验证服务请求验证节点。

5.2.2 交易流程

图5　　访问权限验证用例图

5.2.3　消息请求

5.2.3.1 触发事件

当本地安全节点与远程安全节点之间想要进行信息交换时，触发此交易。

基本安全节点对每个DICOM、HTTP或HL7连接总是应用节点权限管理过程。

5.2.3.2 消息结构及约束

消息结构应符合节点权限管理服务消息格式AccessPermissionRequest元素构建要求，消息补充约束见表6。

表 4 节点访问权限管理请求应答消息补充约束

节点	基数	可选项	节点说明	对应数据元标识符
/ServiceAccessAuthenticate	1..1	R		
/ServiceAccessAuthenticate/id	1..1	R	节点OID标识符	
/ServiceAccessAuthenticate/ServiceName	1..1	R	访问的服务名称	

5.2.4　消息应答

5.2.4.1 触发事件

当本地安全节点与远程安全节点之间想要进行信息交换时，触发该消息应答。

5.2.4.2 消息结构及约束

消息结构应符合节点访问权限管理服务消息格式AccessPermissionResponse元素构件要求，消息补充约束见表5。

表 5　节点访问权限管理请求应答的消息补充约束

节点	属性	基数	节点说明	对应数据元标识符
/ServiceAccessAuthenticateResponse	id	1..1	节点OID	

附 录 A
（规范性）
服务定义

节点验证服务WSDL定义如下:

文件名: rhin_Authentication.wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Ver.0.12.1/2018-03-22 Dejun Hsu-->
<wsdl:definitions xmlns:wsoap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.chiss.org.cn/rhin/2015" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
targetNamespace="http://www.chiss.org.cn/rhin/2015">
    <wsdl:import namespace="http://www.chiss.org.cn/rhin/2015" location="rhin_Authentication.xsd"/>
    <wsdl:message name="NodeAuthenticate">
        <wsdl:part name="parameters" element="NodeAuthenticate"/>
    </wsdl:message>
    <wsdl:message name="NodeAuthenticateResponse">
        <wsdl:part name="parameters" element="NodeAuthenticateResponse"/>
    </wsdl:message>
    <wsdl:message name="ServiceAccessAuthenticate">
        <wsdl:part name="parameters" element="ServiceAccessAuthenticate"/>
    </wsdl:message>
    <wsdl:message name="ServiceAccessAuthenticateResponse">
        <wsdl:part name="parameters" element="ServiceAccessAuthenticateResponse"/>
    </wsdl:message>
    <wsdl:portType name="NodeAuthProvider">
        <wsdl:operation name="NodeAuthenticate">
            <wsdl:input name="NodeAuthenticate" message="NodeAuthenticate"/>
            <wsdl:output name="NodeAuthenticateResponse" message="NodeAuthenticateResponse"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:portType name="ServiceAccessAuthProvider">
        <wsdl:operation name="ServiceAccessAuthenticate">
            <wsdl:input name="ServiceAccessAuthenticate" message="ServiceAccessAuthenticate"/>
            <wsdl:output name="ServiceAccessAuthenticateResponse" message="ServiceAccessAuthenticateResponse"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="NodeAuthProviderBinding" type="NodeAuthProvider">
        <wsoap12:binding style="document" transport="authenticateWebserviceSoap"/>
        <wsdl:operation name="NodeAuthenticate">
```

```
            <wsoap12:operation style="document"/>
            <wsdl:input name="NodeAuthenticate">
                <wsoap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="NodeAuthenticateResponse">
                <wsoap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="ServiceAccessAuthProviderBinding" type="ServiceAccessAuthProvider">
        <wsoap12:binding style="document" transport="accessPermissionWebserviceSoap"/>
        <wsdl:operation name="ServiceAccessAuthenticate">
            <wsoap12:operation style="document"/>
            <wsdl:input name="ServiceAccessAuthenticate">
                <wsoap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="ServiceAccessAuthenticateResponse">
                <wsoap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="naWebservice">
        <wsdl:port name="NodeAuthProviderImplPort" binding="NodeAuthProviderBinding">
            <wsoap12:address location="{rhinHost}/NodeAuthProvider"/>
        </wsdl:port>
        <wsdl:port name="ServiceAccessAuthProviderImplPort" binding="ServiceAccessAuthProviderBinding">
            <wsoap12:address location="{rhinHost}/NodeAuthProvider"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

附 录 B
（规范性）
节点验证服务消息格式

节点验证服务消息格式采用XML Schema定义如下：

文件名：rhin_Authentication.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Ver.0.11.0/2018-04-26 Dejun Hsu-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.chiss.org.cn/rhin/2015"
    targetNamespace="http://www.chiss.org.cn/rhin/2015" elementFormDefault="qualified">
    <xs:include schemaLocation="Base/RHIN-Base.xsd"/>
    <xs:element name="NodeAuthenticate" type="NodeAuthenticateType">
        <xs:annotation>
            <xs:documentation>节点认证请求消息</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="NodeAuthenticateResponse" type="NullResponse">
        <xs:annotation>
            <xs:documentation>节点认证请求应答消息</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="ServiceAccessAuthenticate" type="ServiceAccessAuthenticateType">
        <xs:annotation>
            <xs:documentation>访问权限认证请求消息</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:element name="ServiceAccessAuthenticateResponse" type="NullResponse">
        <xs:annotation>
            <xs:documentation>访问权限认证响应消息</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:complexType name="NodeAuthenticateType">
        <xs:sequence>
            <xs:element name="id" type="OID" minOccurs="1" maxOccurs="1">
                <xs:annotation>
                    <xs:documentation>节点OID</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="digestValue" type="Base64Binary" minOccurs="1" maxOccurs="1">
                <xs:annotation>
```

```
                    <xs:documentation>消息摘要</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="signatureValue" type="Base64Binary" minOccurs="1" maxOccurs="1">
                <xs:annotation>
                    <xs:documentation>消息摘要签名</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ServiceAccessAuthenticateType">
        <xs:sequence>
            <xs:element name="id" type="OID" minOccurs="1" maxOccurs="1">
                <xs:annotation>
                    <xs:documentation>节点OID</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="serviceName" type="String" minOccurs="1" maxOccurs="1">
                <xs:annotation>
                    <xs:documentation>访问的服务名称</xs:documentation>
                </xs:annotation>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

附 录 C
（规范性）
证书结构

C.1 证书及结构表达

证书采用X.509结构,采用ISO/ITU-T ASN.1语法进行表达。

C.2 证书整体结构

整体结构描述如下：

```
Certificate::=SEQUENCE{
    tbsCertificate          TBSCertificate,
    signatureAlgorithm    AlgorithmIdentifier,
    signatureValue          BIT STRING
}
```

表达式说明如下：

——Certificate：证书；

——SEQUENCE：表示序列结构；

——tbsCertificate TBSCertificate：表示证书基本域，TBSCertificate 类型；

——signatureAlgorithm AlgorithmIdentifier：表示签名算法，AlgorithmIdentifier 类型；

——signatureValue BIT STRING：表示签名值，BIT STRING 类型。

签名算法可采用表C.1所列的算法OID代码：

表 C.1 签名算法 OID

算法OID代码	算法名称	算法说明
1.2.840.113549.1.1.4	MD5wihRSAEncryption	基于MD5的RSA的签名算法
1.2.840.113549.1.1.5	SHA1withRSAEncryption	基于SHA1的RSA签名算法
1.2.840.10045.2.1	ECC	椭圆加密算法
1.2.156.10197.1.301	SM2	国密SM2签名算法
1.2.156.10197.1.501	SM3withSM2	基于国密SM3的SM2签名算法
1.2.156.10197.1.503	SHA256withSM2	基于SHA256的SM2签名算法
1.2.156.10197.1.504	SM3withRSAEncryption	基于SM3的RSA签名算法

C.3 签名算法类型（AlgorithmIdentifier）结构

签名算法类型（AlgorithmIdentifier）结构描述如下：

```
AlgorithmIdentifier::=SEQUENCE{
    algorithm           OBJECT IDENTIFIER,
    parameters          ANY DEFINED BY algorithm OPTIONAL
```

}

表达式说明如下：

——AlgorithmIdentifier：表示签名算法类型；

——SEQUENCE：表示序列结构；

——algorithm OBJECT IDENTIFIER：表示采用的签名算法及所代表的 OID；

——parameters ANY DEFINED BY algorithm OPTIONAL：表示签名算法传递的参数。

C.4 证书基本域类型（TBSCertificate）结构

证书基本域类型（TBSCertificate）结构描述如下：

```
TBSCertificate::=SEQUENCE{
    version                 [0]     EXPLICIT Version DEFAULT v1,
    serialNumber                     CertificateSerialNumber,
    signature                        AlgorithmIdentifier,
    issuer                           Name,
    validity                         Validity,
    subject                          Name,
    subjectPublicKeyInfo             SubjectPublicKeyInfo,
    issuerUniqueID          [1]      IMPLICIT UniqueIdentifier OPTIONAL,
    subjectUniqueID         [2]      IMPLICIT UniqueIdentifier OPTIONAL,
    extensions              [3]      EXPLICIT Extensions OPTIONAL
}
```

表达式说明如下：

——TBSCertificate：证书基本域；

——version [0] EXPLICIT Version DEFAULT v1:表示确切的版本信息，默认值为 0，表示版本 V1；

——serialNumber,CertificateSerialNumber:表示证书序列号，CertificateSerialNumber 类型；

——signature AlgorithmIdentifi：表示 CA 签发证书时的签名与签名算法， AlgorithmIdentifier 类型；

——issuer Name:表示证书发布者，Name 类型；

——validity Validity,表示有效期，Validity 类型；

——subject Name:表示证书主题，Name 类型；

——subjectPublicKeyInfo,SubjectPublicKeyInfo：表示被绑定的证书持有者的公钥信息，SubjectPublicKeyInfo 类型；

——issuerUniqueID,IMPLICIT UniqueIdentifier OPTIONAL:表示证书签发者的唯一标识，UniqueIdentifier 类型；

——subjectUniqueID [2] IMPLICIT UniqueIdentifier OPTIONAL：表示证书持有者的唯一标识，UniqueIdentifier 类型；

——extensions [3] EXPLICIT Extensions OPTIONAL：表示扩展信息，Extensions 类型。

C.5 证书序列号类型（CertificateSerialNumber）结构

证书序列号类型（CertificateSerialNumber）结构描述如下：

CertificateSerialNumber::=INTEGER

表达式说明如下：

——CertificateSerialNumber：表示证书序列号类型；

——INTEGER：表示正整数类型，长度不大于 20 字节。

C.6 名称类型（Name）结构

名称类型（Name）结构描述如下：

```
Name::=CHOICE{
    RDNSequence
}
RDNSequence::=SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName::=SET OF AttributeTypeAndValue
AttributeTypeAndValue::=SEQUENCE{
    type      AttributeType,
    value     AttributeValue
}
```

表达式说明如下：

——Name:表示名称类型；

——RDNSequence：表示 RDN 序列类型；

——RelativeDistinguishedName：表示相对专有名称（RDN）类型；

——AttributeTypeAndValue：属性值对类型；

——type AttributeType：表示类型，属性类型；

——value AttributeValue：表示值，属性值类型。

相对专有名称 (RelativeDistinguishedName)宜采用表C.2所表

表 C.2 相对专有名称(RDN)列表

RDN代码	RDN名称	英文名称
CN	通用名称	Common Name
OU	机构单元名称	Organizational Unit name
O	机构名	Organization name
L	地理位置	Locality
S	州/省名	State or province name
C	国名	Country

C.7 属性类型（AttributeType）及属性值（AttributeValue）类型结构

属性类型（AttributeType）及属性值（AttributeValue）类型结构描述如下：

```
AttributeType::=OBJECT IDENTIFIER
AttributeValue::=ANY DEFINED BY AttributeTypeAlgorithmIdentifier::=SEQUENCE{
    algorithm       OBJECT IDENTIFIER,
    parameters      ANY DEFINED BY algorithm OPTIONAL
```

}

表达式说明如下：

——AttributeType::=OBJECT IDENTIFIER：表示属性类型，采用 OID 结构；

——AttributeValue::=ANY DEFINED BY AttributeTypeAlgorithmIdentifier：属性值类型，由属性类型的签名算法 ID 决定。

参 考 文 献

1)IHE International, Inc.IHE IT Infrastructure Technical Framework, Volume 1 (ITI TF-1): Integration Profiles,2015.

2)IHE International, Inc.IHE IT Infrastructure Technical Framework, Volume 2a (ITI TF-2a): Transactions Part A,2015.

3)W3C.Web Services Security 1.0(WS-Security), April 5, 2002