ICS 03.220.20
R07

# T/ITS

# 中国智能交通产业联盟标准

T/ITS 0074—2017

# 智能交通 数字证书应用接口规范

Intelligent transport - Certificate application interface

2017-12-10 发布　　2018-03-01 实施

中国智能交通产业联盟 发布

# 目　　次

# 前　言

本标准按照GB/T 1.1—2009给出的规则起草。

本标准由中国智能交通产业联盟归口。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准起草单位：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、360企业安全集团、恒安嘉新(北京)科技股份公司、国家互联网应急中心、北京信息科技大学。

本标准主要起草人：王笑京、孟春雷、梅新明、周洲、孙婧、王立岩、武俊峰、宋向辉、陈晓光、郑新华、刘鸿伟、王永建、赵童、吴秋新。

# 引　　言

（为使智能交通数字证书应用接口规范能够按统一的标准进行说明和描述，特制定本标准。

为了保持标准的适用性与可操作性，各使用者在采标过程中，及时将对本标准规范的意见及建议函告交通运输部公路科学研究院，以便修订时研用。

地址：北京市海淀区西土城路8号，邮编：100088，邮箱：sunjing@itsc.cn。）

# 智能交通 数字证书应用接口

## 1 范围

本标准规定了智能运输系统中的数字证书应用接口和安全消息语法，规定了安全消息语法中的基本元素格式。

本标准适用于智能运输系统中数字证书应用相关的软硬件系统（包括合作式智能运输系统和车联网等应用）的设计、研发、测试及数字证书认证机构的运行、维护和管理。

## 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，仅所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 16262.1 信息技术 抽象语法记法-(ASN.1) 第1部分：基本记法规范

GB/T 20518-2006 信息安全技术 公钥基础设施 数字证书格式

GB/T 25069-2010 信息安全技术 术语

GM/T 0009 SM2密码算法使用规范

GM/T 0010 SM2密码算法加密签名消息语法规范

GM/T 0015-2012 基于SM2密码算法的数字证书格式规范

GM/T 0020-2012 证书应用综合服务接口规范

## 3 术语和定义

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 25069中的某些术语和定义。

3.1

**智能运输系统 intelligent transport systems(ITS)**

又称智能交通系统，是在较完善的交通基础设施之上，在先进的信息、通信、计算机、自动控制和系统集成等技术前提下，通过先进的交通信息采集与融合技术，交通对象交互以及智能化交通控制与管理等专有技术，加强载运工具载体和用户之间的联系，提高交通系统的运行效率，

减少交通事故，降低环境污染，从而建立一个高效、便捷、安全、环保、舒适的综合交通运输体系。

[GB/T 20839-2007，定义2.1]

3.2

**合作式智能运输系统 cooperative ITS**

通过人、车、路信息交互，实现车辆和基础设施之间（V2I）、车辆与车辆（V2V）、车辆与人（V2P）之间的智能协同与配合的一种智能运输系统体系。

3.3

**数字证书 digital certificate**

由认证权威数字签名的包含公开密钥拥有者信息、公开密钥、签发者信息、有效期以及一些扩展信息的数字文件。

3.4

**SM2算法 SM2 algorithm**

一种椭圆曲线密码算法，密钥长度为256比特。

3.5

**算法标识 algorithm identifier**

用于标明算法机制的数字化信息。

## 4 缩略语

以下缩略语适用于本文件。

ASN.1：抽象语法标记（Abstract Syntax Notation One）

CA： 认证机构（Certificate Authority）

OER：八位字节编码规则（Octet Encoding Rules）

CRL：证书撤销列表（Certificate Revocation List）

ITS：智能运输系统（Intelligent transport system）

OID：对象标识符（Object Identifier）

UTC：协调世界时（Coordinated Universal Time）

## 5 数字证书应用接口

## 5.1 概述

本规范所定义的证书应用接口包括：获取版本号、消息签名与验证、消息加密与解密、消息信封等。数字证书应用接口处理安全消息，定义的应用数字证书消息的数据结构在第6章说明。

## 5.2 消息签名

原型： BSTR SOF_SignData(INTEGER itsAid,BSTR InData)

描述： 对消息进行数字签名

参数： itsAid[in] 签名者证书

InData[in] 消息原文

返回值： 非空 安全消息

空 失败

消息原文为TBSData结构的编码结果。使用签名者证书找到匹配的签名密钥，对原文进行签名。接口返回的安全消息为本文定义的签名消息的编码结果。

## 5.3 验证签名

原型： BOOL SOF_VerifySignedData(BSTR EncodeCert, BSTR SignValue)

描述： 验证消息数字签名

参数： EncodeCert[in] 签名者证书

SignValue[in] 安全消息

返回值： TRUE 成功

FALSE 失败

## 5.4 非对称消息加密

原型： BSTR SOF_EncryptData(PublicEncryptionKey[] pk,BSTR InData)

描述： 对消息进行数据加密

参数： pk [in] 加密公钥

InData[in] 消息原文

返回值： 非空 安全消息

空 失败

## 5.5 非对称消息解密

原型： BSTR SOF_DecryptData(BSTR InData)

描述：　对消息进行数据解密

参数：

| | | |
|---|---|---|
| | InData[in] | 安全消息 |
| 返回值： | 非空 | 消息原文 |
| | 空 | 失败 |

### 5.6　对称消息加密

原型：　BSTR SOF_EncryptData(INTEGER symmkeyindex,BSTR InData)

描述：　对消息进行数据加密

| | | |
|---|---|---|
| 参数： | symmkeyindex | 对称密钥索引 |
| | InData[in] | 消息原文 |
| 返回值： | 非空 | 安全消息 |
| | 空 | 失败 |

### 5.7　对称消息解密

原型：　BSTR SOF_DecryptData(BSTR InData)

描述：　对消息进行数据解密

| | | |
|---|---|---|
| 参数： | InData[in] | 安全消息 |
| 返回值： | 非空 | 消息原文 |
| | 空 | 失败 |

## 6　安全消息语法

### 6.1　综述

安全消息语法由多种类型的基本元素构成，基本元素格式在第7章描述，本规范对应的接口处理的数据分为两种类型：

a)　当应用证书类型为机构证书、公务人员证书、社会公众证书、设备证书时，消息数据结构类型遵循 GM/T 0010；

b)　当应用证书类型为 ITS 设备证书时，安全消息数据结构类型遵循本规范。

### 6.2　合作式 ITS 安全消息语法

#### 6.2.1　安全消息　SecuredMessage

此结构定义了如何编码通用的安全信息。

合作式 ITS 安全消息基本数据结构如下：

```
secureMessage ::= SEQUENCE{
    version     Uint8,
    payload     Payload
}
```

version：本文档中定义版本为 2；

payload：消息负载；

示例参见附录A。

### 6.2.2 消息负载

```
Payload ::= CHOICE{
    unSecuredData     Opaque,
    signedData        SignedData,
    encData           EncryptedData
}
```

消息负载分为3种：

unSecuredData：未签名、未加密的内容；

signed：签名负载；

encData：加密负载；

#### 6.2.2.1 签名消息内容

```
SignedData ::= SEQUENCE{
    signer     SignerInfo,
    tbs        TBSData,
    sign       Signature
}
```

signer：签名者标识；

tbs：待签名的数据；

```
TBSData ::= SEQUENCE{
    headerInfo    HeaderInfo,
    data          OCTET STRING (SIZE(0..MAX))     Optional,
    extHash       OCTET STRING (SIZE(32))           Optional
```

}

待签名数据形式分为 2 种：

data：随负载发送的待签名的数据；

extHash：如果待签名数据较大或对方已知，不随负载发送，则用外部待签数据的摘要表示；

sign：对 tbs 的签名结果；

#### 6.2.2.2 签名消息头

```
HeaderInfo ::= SEQUENCE{
    itsAid        INTEGER,
    hashAlg      HashAlgorithm   Optional,
    genTime      Time64    Optional,
    expiryTime    Time64     Optional,
    location      ThreeDLocation    Optional,
    digest        HashedId3  Optional,
    encKey       PublicEncryptionKey    Optional
}
```

itsAid：智能交通应用标识；

hashAlg：digest和extHash所使用的摘要算法；缺省为SM3；

genTime：消息产生时间；

expiryTime：消息的失效时间；

location：消息产生的地理坐标；

digest：用于SDS获取仅有标识、但没有完整内容的证书；

当encKey存在时，表示响应数据需要使用encKey指定的对称加密算法加密，且用指定的公钥对对称密钥加密。

#### 6.2.2.3 加密消息

```
EncryptedData ::= SEQUENCE{
    recipients     SequenceOfRecipientInfo,
    cipherText    SymmetricCipherText
}
```

recipients：消息接收者；

cipherText：被加密的消息密文和算法参数；

```
SequenceOfRecipientInfo ::= SEQUENCE OF RecipientInfo
```

#### 6.2.2.4 消息密文

定义了由不同的对称加密算法得到的密文。

```
SymmetricCipherText ::= CHOICE{
    sm4Ecb    CipherText,
    sm4Cbc    SM4CipherTextCbc,
    sm4Cfb    SM4CipherTextCbc,
    sm4Ofb    SM4CipherTextCbc,
    aesCcm    AesCcmCipherText
}
```

对称加密的密文：

```
CipherText ::= Opaque
```

国密算法cbc/ofb/ofb分组模式的密文和参数：

```
SM4CipherTextCbc ::= SEQUENCE{
    iv    OCTET STRING (SIZE(16)),
    cipher    CipherText
}
```

iv：初始化向量；

cipher：密文；

AesCcm的密文和参数：

```
AesCcmCipherText ::= SEQUENCE{
    nonce    OCTET STRING (SIZE(12)),
    cipher    CipherText
}
```





## 7 基本元素格式

### 7.1 综述

本标准使用ASN.1对数据结构和消息语法进行描述，采用八位字节编码规则(OER)对签名加密消息的各项信息进行编码。本部分定义在合作式智能交通系统数据结构中的基本元素类型，这些元素类型在合作式智能交通系统的安全消息中使用。

## 7.2 整型

下面的基本类型在数据结构定义中使用：

```
Uint3 ::= INTEGER (0..7)

Uint8 ::= INTEGER (0..255)

Uint16 ::= INTEGER (0..65535)

Uint32 ::= INTEGER (0..4294967295)

Uint64 ::= INTEGER (0..18446744073709551615)

IValue ::= Uint16
```

下列结构用来阐明定义：

```
OCTET STRING (SIZE(0..MAX))
```

## 7.3 32 位时间 Time32

```
Time32 ::= Uint32
```

Time32 是一个 32 位无符号整数,高位优先编码格式,自 2004 年 1 月 1 日 UTC 00:00:00 开始,给出国际原子时间的秒数。

注 1:232 秒的周期持续约 136 年，直到 2140 年。

## 7.4 64 位时间 Time64

```
Time64 ::= Uint64
```

Time64是一个64位无符号整数,高位优先编码格式,自2004年1月1日UTC 00:00:00开始，给出国际原子时间的微秒数。

## 7.5 纬度 Latitude

```
Latitude ::= NinetyDegreeInt
NinetyDegreeInt ::= INTEGER {
    min (-900000000),
    max (900000000),
    unknown (900000001)
} (-900000000..900000001)
KnownLatitude ::= NinetyDegreeInt (min..max)
```

-负90度到正90度，间隔为1微度。

```
UnknownLatitude ::= NinetyDegreeInt (unknown)
```

纬度字段包含一个整数编码，精度为1微度。

允许的纬度值范围从-900 000 000~ + 900 000 000。值900 000 001表明纬度是不可用的。

### 7.6 经度 Longitude

```
Longitude ::= OneEightyDegreeInt
OneEightyDegreeInt ::= INTEGER {
    min (-1799999999),
    max (1800000000),
    unknown (1800000001)
} (-1799999999..1800000001)
KnownLongitude ::= OneEightyDegreeInt (min..max)
UnknownLongitude ::= OneEightyDegreeInt (unknown)
```

经度字段包含一个整数编码，精度为1微度。

允许的经度值范围从-1 800 000 000~+1 800 000 000。值1 800 000 001表示经度是不可用的。

### 7.7 3D 位置 ThreeDLocation

```
ThreeDLocation ::= SEQUENCE {
    latitude        Latitude,
    longitude       Longitude,
    elevation       Elevation
}
Elevation ::= Uint16
```

表示海拔，单位：分米。取值范围 -4095（0xF001）dm ~ 61439（0xEFFF）dm。

### 7.8 8 字节哈希值 HashedId8

```
HashedId8 ::= OCTET STRING (SIZE(8))
```

本哈希值用来识别证书等数据。首先计算输入数据的哈希值,取32字节的哈希值的低8个字节。

### 7.9 3 字节哈希值 HashedId3

```
HashedId3 ::= OCTET STRING (SIZE(3))
```

本哈希值用来识别证书等数据。首先计算输入数据的哈希值,取32字节的哈希值的低3个字节。

### 7.10 Hash 算法

```
HashAlgorithm ::= ENUMERATED{
    sgdsm3,
    sha256,
    sha3-256,
    ...
}
```

### 7.11 椭圆曲线

```
EccCurve ::= ENUMERATED{
    sgdsm2,
    nistP256,
    brainpoolP256r1,
    ...
}
```

### 7.12 对称算法 SymmetricAlgorithm

```
SymmetricAlgorithm ::= ENUMERATED {
    sgdsm4ecb                    ,
    sgdsm4cbc                    ,
    sgdsm4cfb                    ,
    sgdsm4ofb                    ,
    aes128ccm,
    ...
}
```

本枚举表示基于对称密钥加密的算法。

### 7.13 加密公钥 PublicEncryptionKey

```
PublicEncryptionKey ::= SEQUENCE  {
    supportedSymmAlg      SymmetricAlgorithm,
    curve                 EccCurve,
    publicKey             ECCPoint
}
```

该结构表示一个用于非对称加密计算的公钥，和可支持的对称密码算法。

### 7.14 加密的对称密钥 EciesEncryptedKey

```
EciesEncryptedKey::= SEQUENCE {
    eccCcurve    EccCurve,
    v  ECCPoint,
    c OCTET STRING(SIZE(16)),
    t OCTET STRING(SIZE(32))
}
```

此结构定义了如何传送公钥加密的对称密钥。

eccCurve：非对称密码运算使用的椭圆曲线；

向量v：包含了用于加密的发送方的临时密钥。此临时密钥v应只使用一次，每次加密应生成一新的密钥。

向量c：包含被加密的对称密钥。

向量t：当公钥加密算法为国密算法时，对应国密算法中的M；当公钥加密算法为国际算法时，包含16字节的身份验证标记，且放在前16字节，后16字节为零。

### 7.15 签名者信息 SignerInfo

本项描述了证书签名者的信息。该项被定义为SignerInfo类型，其结构如下：

```
SignerInfo  ::= CHOICE {
    self                          NULL,
    certificateDigest        CertificateDigest,
    ...
}
CertificateDigest ::= SEQUENCE  {
    algorithm          HashAlgorithm,
```

```
    digest              HashedId8
}
```

self：自签名，无额外数据。

certificateDigest：用指定的摘要算法、对证书计算Hash，得到的标识。

## 7.16 接收者信息 RecipientInfo

```
RecipientInfo ::= CHOICE{
    pskRecipientInfo        PreSharedKeyRecipientInfo,
    symmRecipInfo           SymmRecipInfo,
    certRecipInfo           PKRecipientInfo,
    signedDataRecipInfo     PKRecipientInfo
}
```

PreSharedKeyRecipientInfo：预共享的对称密钥；

```
PreSharedKeyRecipientInfo ::= SEQUENCE{
    hashAlg     HashAlgorithm,
    symKeyHash     HashedId8
}
```

symKeyHash：共享的对称密钥的摘要；

hashAlg：计算 smyKeyHash 的摘要算法；

SymmRecipInfo：经过其他密钥对称加密过的对称密钥的密文；

```
SymmRecipInfo ::= SEQUENCE  {
    hashAlg     HashAlgorithm,
    recipientId         HashedId8,
    encKey          SymmetricCipherText
}
```

encKey：用于数据加密的对称密钥，被其他对称密钥加密过的密文；

recipientId：用于解密 encKey 的对称密钥的摘要；

hashAlg：用于计算 recipientId 的摘要算法；

PKRecipientInfo：公钥加密的对称密钥的密文；

```
PKRecipientInfo = SEQUENCE{
    hashAlg       HashAlgorithm,
    recipientId    HashedId8,
    kek           EciesEncryptedKey
}
```

recipientId：非对称加密的公钥所属内容的 hash。当公钥来自证书，则为证书的 hash；当公钥来自签名消息中的 encKey，则为 SignedData 的 hash。

hashAlg：计算 recipientId 的摘要算法；

kek：用接收者公钥加密的对称密钥；

### 7.17 签名 Signature

本结构定义了一个容器,该容器用于封装基于公钥密码算法的签名值。其结构如下:

```
Signature ::= SEQUENCE {
    curve        EccCurve,
    r       ECCPoint,
    s      OCTET STRING (SIZE (32))
}
```

*注： 国密算法时，ECCPoint均采用x-only形式。

签名计算的原文为：Hash( Hash(data) || Hash(signer) )；其中data为待签数据的字节，signer为签名者。当自签时，signer为零长字节数组【””.getBytes()】；当非自签时，为签名者证书的编码内容。

# 附 录 A

（资料性附录）
合作式 ITS 安全签名消息示例

## A.1 概述

本部分是一个合作式ITS安全签名消息的示例，示例数据定义了一个符合SecuredMessage结构的签名消息数据，有效负载为0x11223344，整个消息结构共87字节。

## A.2 签名消息数据

```
Total encoded length = 87
Encoded successfully in 87 bytes:
02818100 11223344 55667788 80000101 04112233 44008001 02030405 06070801
02030405 06070801 02030405 06070801 02030405 06070801 02030405 06070801
02030405 06070801 02030405 06070801 02030405 060708
```

## A.3 签名数据解析

```
SecureMessage SEQUENCE
  version Uint8 INTEGER [length = 1]
    2
  payload Payload CHOICE [index = 1]
    signedData SignedData SEQUENCE
      signer SignerInfo CHOICE [index = 1]
        certificateDigest CertificateDigest SEQUENCE
          algorithm HashAlgorithm ENUMERATED [length = 1]
            0
          digest HashedId8 OCTET STRING [length = 8]
            0x1122334455667788
      tbs TBSData SEQUENCE
```

```
  headerInfo HeaderInfo SEQUENCE
    itsAid INTEGER [length = 1]
      1
  data OCTET STRING [length = 4]
    0x11223344
sign Signature SEQUENCE
  curve EccCurve ENUMERATED [length = 1]
    0
  r ECCPoint CHOICE [index = 0]
    x-only OCTET STRING [length = 32]
      0x0102030405060708010203040506070801...
  s OCTET STRING [length = 32]
    0x0102030405060708010203040506070801...
```

# 附 录 B

（资料性附录）
合作式 ITS 安全加密消息示例

## B.1 概述

本部分是一个合作式ITS安全加密消息的示例，示例数据定义了一个符合SecuredMessage结构的加密消息数据，密文为0xfabbcc1ffabbcc1ffabbcc1ffabbcc1f，整个消息结构共146字节。

## B.2 加密消息数据

```
Total encoded length = 146
Encoded successfully in 146 bytes:
02820101 82001122 33445566 77880084 01020304 05060708 01020304 05060708
01020304 05060708 01020304 05060708 01020304 05060708 01020304 05060708
01020304 05060708 01020304 05060708 11223344 55667788 11223344 55667788
11223344 55667788 11223344 55667788 11223344 55667788 11223344 55667788
8010FABB CC1FFABB CC1FFABB CC1FFABB CC1F
```

## B.3 加密数据解析

```
SecureMessage SEQUENCE
  version Uint8 INTEGER [length = 1]
    2
  payload Payload CHOICE [index = 2]
    encData EncryptedData SEQUENCE
      recipients SequenceOfRecipientInfo SEQUENCE OF
        RecipientInfo CHOICE [index = 2]
          certRecipInfo PKRecipientInfo SEQUENCE
            hashAlg HashAlgorithm ENUMERATED [length = 1]
              0
```

```
        recipientId HashedId8 OCTET STRING [length = 8]
          0x1122334455667788
        kek EciesEncryptedKey SEQUENCE
          eccCcurve EccCurve ENUMERATED [length = 1]
            0
          v ECCPoint CHOICE [index = 4]
            uncompressed SEQUENCE
              x OCTET STRING [length = 32]
                0x0102030405060708010203040506070801...
              y OCTET STRING [length = 32]
                0x0102030405060708010203040506070801 ...
          c OCTET STRING [length = 16]
            0x11223344556677881122334455667788
          t OCTET STRING [length = 32]
            0x1122334455667788112233445566778811 ...
    cipherText SymmetricCipherText CHOICE [index = 0]
      sm4Ecb CipherText OCTET STRING [length = 16]
        0xfabbcc1ffabbcc1ffabbcc1ffabbcc1f
```

T/ITS 0074-2016

中国智能交通产业联盟
标准

**智能交通 数字证书应用接口规范**

T/ITS 0074-2016

北京市海淀区西土城路 8 号（100088）
中国智能交通产业联盟印刷
网址：http://www.c-its.org

2017 年 11 月第一版　2017 年 11 月第一次印刷