



Technical Specification

## Intelligent Transport Systems (ITS); Security; Security header and certificate formats

**STANDARDS PREVIEW**  
(standard.it-eu-api)  
Full standard available at <https://standards.iteh.ai/catalog/standards/sist/0080ce51-8a04-44b6-ae23-b692aeb41296/etsi-ts-103-097-v1.1.1-2013-04>

---

**Reference**

DTS/ITS-0050023

---

**Keywords**

ITS, privacy, protocol, security

**ETSI**

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

**Important notice**

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**Copyright Notification**

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2013.  
All rights reserved.

DECT™, PLUGTESTS™, UMTS™ and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.  
3GPP™ and LTE™ are Trade Marks of ETSI registered for the benefit of its Members and  
of the 3GPP Organizational Partners.  
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

# Contents

Intellectual Property Rights .....	5
Foreword.....	5
Introduction .....	5
1 Scope .....	6
2 References .....	6
2.1 Normative references .....	6
2.2 Informative references.....	6
3 Definitions and abbreviations.....	7
3.1 Definitions .....	7
3.2 Abbreviations .....	7
4 Basic format elements .....	7
4.1 Presentation Language .....	7
4.2 Specification of basic format elements.....	9
4.2.1 IntX.....	9
4.2.2 PublicKeyAlgorithm.....	9
4.2.3 SymmetricAlgorithm .....	9
4.2.4 PublicKey .....	9
4.2.5 EccPoint.....	10
4.2.6 EccPointType.....	11
4.2.7 EncryptionParameters.....	11
4.2.8 CrlSeries .....	11
4.2.9 Signature .....	11
4.2.10 EcdsaSignature .....	12
4.2.11 SignerInfo .....	12
4.2.12 SignerInfoType .....	13
4.2.13 HashedId8 .....	13
4.2.14 HashedId3.....	13
4.2.15 Time32.....	13
4.2.16 Time64.....	14
4.2.17 Time64WithStandardDeviation.....	14
4.2.18 Duration .....	14
4.2.19 TwoDLocation .....	14
4.2.20 ThreeDLocation .....	15
4.2.21 GeographicRegion .....	15
4.2.22 RegionType.....	16
4.2.23 CircularRegion.....	16
4.2.24 RectangularRegion.....	16
4.2.25 PolygonalRegion.....	16
4.2.26 IdentifiedRegion .....	17
4.2.27 RegionDictionary.....	17
5 Specification of security header .....	17
5.1 SecuredMessage .....	17
5.2 Payload .....	18
5.3 PayloadType.....	18
5.4 HeaderField .....	18
5.5 HeaderFieldType .....	19
5.6 TrailerField.....	20
5.7 TrailerFieldType.....	20
5.8 RecipientInfo .....	20
5.9 EciesNistP256EncryptedKey .....	21
6 Specification of certificate format .....	21
6.1 Certificate .....	21

6.2	SubjectInfo .....	22
6.3	SubjectType .....	22
6.4	SubjectAttribute .....	23
6.5	SubjectAttributeType .....	23
6.6	SubjectAssurance .....	24
6.7	ValidityRestriction .....	24
6.8	ValidityRestrictionType .....	25
6.9	ItsAidSsp .....	25
6.10	ItsAidPriority .....	25
6.11	ItsAidPrioritySsp .....	25
7	Security profiles .....	26
7.1	Security profile for CAMs .....	26
7.2	Security profile for DENMs .....	27
7.3	Generic security profile for other signed messages .....	28
7.4	Profiles for certificates .....	28
7.4.1	Authorization tickets (pseudonymous certificates) .....	29
7.4.2	Enrollment credential (long-term certificates) .....	29
7.4.3	Certificate authority certificates .....	29
<b>Annex A (informative): Data structure examples.....</b>		<b>31</b>
A.1	Example security envelope structure for CAM .....	31
A.2	Example structure of a certificate .....	31
History .....		33

**iTeh STANDARD PREVIEW**  
 (standards.iteh.ai)

Full standard:  
<https://standards.iteh.ai/catalog/standards/sist/0080ce51-8a04-44b6-ae23-b692aeb41296/etsi-ts-103-097-v1.1.1-2013-04>

---

## Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://ipr.etsi.org>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

---

## Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Intelligent Transport Systems (ITS).

---

## Introduction

Security mechanisms for ITS consist of a number of parts. An important part for interoperability is a common format for data elements being transferred between ITS stations for security purposes.

The present document intends to provide such a format definition. A special focus is to include as much as possible from existing standards. At the same time, the major goal is simplicity and extensibility of data structures.

iTeh STANDARD PREVIEW  
(standards.iteh.ai)  
Full standard:  
https://standards.iteh.ai/catalog/standards/sist/0080ce51-8a04-44b6-ae23-b692aeb41296/etsi-103-097-v1.1.1-2013-04

---

# 1 Scope

The present document specifies security header and certificate formats for Intelligent Transport Systems. These formats are defined specifically for securing G5 communication.

---

## 2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

### 2.1 Normative references

The following referenced documents are necessary for the application of the present document.

- [1] IEEE Std. 1363-2000: "Standard Specifications For Public Key Cryptography".
- [2] NIMA Technical Report TR8350.2: "Department of Defense World Geodetic System 1984. Its Definition and Relationships with Local Geodetic Systems".
- [3] ISO 3166-1: "Codes for the representation of names of countries and their subdivisions - Part 1: Country codes".
- [4] NIST SP 800-38C: "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality".
- [5] IETF RFC 2246: "The TLS Protocol Version 1.0".
- [6] ETSI TS 102 637-2: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service".

### 2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] IEEE Std 1363a-2004: "Standard Specifications For Public Key Cryptography- Amendment 1: Additional Techniques".
- [i.2] IEEE Std. 1609.2-2012 (draft D12): "Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages".
- [i.3] IEEE Std. 1609.2-2012 (draft D17): "Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages".
- [i.4] IEEE Std. 1609.3-2010: "Wireless Access in Vehicular Environments (WAVE) - Networking Services".

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

**enumeration:** set of values with distinct meaning

### 3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CA	Certificate Authority
CAM	Cooperative Awareness Message
CRL	Certificate Revocation List
DENM	Decentralized Environmental Notification Message
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
G5	5,9 GHz radio communications
ITS	Intelligent Transport Systems
ITS-AID	ITS Application ID
ITS-S	Intelligent Transport Systems Station
NIMA	National Imagery and Mapping Agency
NIST SP	National Institute of Standards and Technology, Special Publication
PSID	Provider Service Identifier

NOTE: It is a synonym for ITS-AID.

SSP	Service Specific Permissions
TAI	Temps Atomique International (International Atomic Time)
UTC	Universal Time Coordinated
WGS	World Geodetic System

---

## 4 Basic format elements

### 4.1 Presentation Language

The presentation language is derived from the Internet Engineering Task Force (IETF) RFC 2246 (TLS) [5] and from IEEE Std. 1609.2-2012 [i.2] (draft D12) and is described in table 1.

NOTE: The presentation language is not formally defined. Parsing tools based on this notation cannot be guaranteed to be consistent or complete.

Table 1: Presentation language

Element	Description	Example(s)
Variable names	Variable names are given in lower case	variable_name
Basic data types	Basic data types are given in lower case	uint8, uint16, uint32, uint64
Composed data types	Composed data types are given with at least the first letter in upper case	MyDataType
Comments	Comments start with the "//" indicator	// This is a comment
Numbers	Numbers are given as signed or unsigned big-endian octets, i.e. network byte order	uint8, uint16, uint32, uint64, sint32
Fixed-length vectors	Fixed-length vectors have a data type and a fixed octet size given in square brackets	uint8 Coordinates[2]; // two uint8 values uint32 Coordinates[8]; // two uint32 values
Variable-length vectors with fixed length encoding	The number in angle brackets gives the maximum number of octets. Depending on the maximum size, the first 1, 2, 4 or 8 bytes encode the actual field length	uint8 AsciiChar; AsciiChar Name<2^8-1>; // "abc" encoded as // 0x03, 0x61, 0x62, 0x63 AsciiChar LongName<2^16-1>; // "abc" encoded as // 0x00, 0x03, 0x61, 0x62, 0x63
Variable-length vectors with variable-length encoding	<var> indicates variable-length encoding. The length itself is encoded with a number of "1" bits according to the additional number of octets used to encode the length, followed by a "0" bit and the actual length value.	uint8 AsciiChar; AsciiChar Name<var>;  // encoding examples: (the bits with // grey background represent the // length encoding of the vector's // length, X the first of the // // vector's following payload bits)  // Vector length 5: // Bits: 00000101 XXXXXXXX XXXXXXXX  // Vector length 123: // Bits: 01111011 XXXXXXXX XXXXXXXX  // Vector length 388: // Bits: 10000001 10000100 XXXXXXXX
Opaque fields	Opaque fields are blocks of data whose content interpretation is not further specified	opaque fieldname[n]; opaque fieldname<n>; opaque fieldname<var>;
Enumerations	Enumerations are list of labels with a unique value for each label, and optionally a maximum value (which then determines length of encoding)	enum {de(0), fr(1), it(2)} Country; enum {de(0), fr(1), it(2), (2^8-1)} Country; // both variants encoding in one // octet enum {de(0), fr(1), it(2), (2^16-1)} Country; // Encoding in two octets
Constructed types	Constructed types contain other types	struct { Name name; Country country; } Person;
Case statements	Case statements are used inside constructed types to change the contents of the constructed type depending on the value of the variable given in brackets	struct { Name name; Country country; select(country) { case de: uint8 age; case fr: AsciiChar given_name<2^8-1>; } } Person;
External data	This is external data that has impact on a struct, e.g. in a select statement. It shall be described from where the external data is obtained.	struct { Name name; extern Country country; select(country) { case de: uint8 age; case fr: AsciiChar given_name<2^8-1>; } } Person;



## 4.2 Specification of basic format elements

### 4.2.1 IntX

IntX int\_x;

This data type encodes an integer of variable length. The length of this integer is encoded by a number of 1 bit followed by a 0 bit, where the number of 1 bit is equal to the number of additional octets used to encode the integer besides those used (partially) to encode the length.

EXAMPLE: `00001010` encodes the integer 10, while `10001000 10001000` encodes the integer 2 184. The bits encoding the length of the element are colored with a grey background.

NOTE: This definition is similar to the definition of PSID in IEEE 1609.3-2010 [i.4], clause 8.1.3, but allows bigger values of the encoded integer.

### 4.2.2 PublicKeyAlgorithm

```
enum {
    ecdsa_nistp256_with_sha256(0),
    ecies_nistp256(1),
    reserved(240..255),
    (2^8-1)
} PublicKeyAlgorithm;
```

This enumeration lists supported algorithms based on public key cryptography. Values in the range of 240 to 255 shall not be used as they are reserved for internal testing purposes.

NOTE: This definition is similar to the one in IEEE 1609.2 Draft D12 [i.2], clause 6.2.16, but `ecdsa_nistp224_with_sha224` is not supported by the present document. As a consequence, the numbering of identical elements (e.g. `ecdsa_nistp256`) differs.

### 4.2.3 SymmetricAlgorithm

```
enum {
    aes_128_ccm(0),
    reserved(240..255),
    (2^8-1)
} SymmetricAlgorithm;
```

This enumeration lists supported algorithms based on symmetric key cryptography. Values in the range of 240 to 255 shall not be used as they are reserved for internal testing purposes. The algorithm `aes_128_ccm` denotes the symmetric key cryptography algorithm AES-CCM as specified in NIST SP 800-38C [4].

NOTE: Except naming, this definition is identical to the one in IEEE 1609.2 Draft D12 [i.2], clause 6.2.23.

### 4.2.4 PublicKey

```
struct {
    PublicKeyAlgorithm    algorithm;
    select(algorithm) {
    case ecdsa_nistp256_with_sha256:
        EccPoint          public_key;
    case ecies_nistp256:
        SymmetricAlgorithm supported_symm_alg;
        EccPoint          public_key;
    unknown:
        opaque             other_key<var>;
    }
} PublicKey;
```

This structure defines a wrapper for public keys by specifying the used algorithm and - depending on the value of `algorithm` - the necessary data fields:

- `ecdsa_nistp256_with_sha256`: the specific details regarding ECC contained in an `EccPoint` structure shall be given.
- `ecies_nistp256`: the specific details regarding ECC contained in an `EccPoint` structure and the symmetric key algorithm contained in a `SymmetricAlgorithm` structure shall be given.
- `unknown`: in all other cases, a variable-length vector containing opaque data shall be given.

NOTE: Except naming of included types, this definition is identical to the one in IEEE 1609.2 Draft D12 [i.2], clause 6.3.31.

## 4.2.5 EccPoint

```
struct {
    extern PublicKeyAlgorithm    algorithm;
    extern uint8                field_size;
    EccPointType                type;
    opaque                      x[field_size];
    select (type) {
        case x_coordinate_only:
        case compressed_lsb_y_0:
        case compressed_lsb_y_1:
            ;
        case uncompressed:
            opaque                y[field_size];
        unknown:
            opaque                data<var>;
    }
} EccPoint;
```

This structure defines a public key based on elliptic curve cryptography according to IEEE Std 1363-2000 [1] clause 5.5.6. An `EccPoint` encodes a coordinate on a two dimensional elliptic curve. The x coordinate of this point shall be encoded in `x` as an unsigned integer in network byte order. Depending on the key type, the y coordinate shall be encoded case-specific:

- `x_coordinate_only`: only the x coordinate is encoded, no additional data shall be given.
- `compressed_lsb_y_0`: the point is compressed and y's least significant bit is zero, no additional data shall be given.
- `compressed_lsb_y_1`: the point is compressed and y's least significant bit is one, no additional data shall be given.
- `uncompressed`: the y coordinate is encoded in the field `y`. The y coordinate contained in a vector of length `field_size` containing opaque data shall be given.
- `unknown`: in all other cases, a variable-length vector containing opaque data shall be given.

The `uint8 field_size` defining the lengths of the vectors containing the raw keys shall be derived from the given algorithm and the mapping as defined in table 2. The necessary algorithm shall be given as an external link to the parameter `pk_encryption` specified in the structure `RecipientInfo`.

**Table 2: Derivation of field sizes depending on the used algorithm**

PublicKeyAlgorithm value	Length in octets
<code>ecdsa_nistp256_with_sha256</code>	32

NOTE: Except inclusion of all remaining elements of the enumeration `EccPointType` that previously matched to case `uncompressed` and inclusion of case `unknown`, this definition is identical to the `EccPublicKey` in IEEE 1609.2 Draft D12 [i.2], clause 6.2.18.

## 4.2.6 EccPointType

```
enum {
    x_coordinate_only(0),
    compressed_lsb_y_0(2),
    compressed_lsb_y_1(3),
    uncompressed(4),
    (2^8-1)
} EccPointType;
```

This enumeration lists supported ECC key types.

NOTE: This definition is identical to the one in IEEE 1609.2 Draft D12 [i.2], clause 6.2.19.

## 4.2.7 EncryptionParameters

```
struct {
    SymmetricAlgorithm symm_algorithm;
    select(symm_algorithm) {
    case aes_128_ccm:
        opaque         nonce[12];
    unknown:
        opaque         params<var>;
    }
} EncryptionParameters;
```

This structure holds basic parameters and additional data required for encryption and decryption of data using different symmetric encryption algorithms. In case of `aes_128_ccm` a 12 octet nonce shall be given. In other cases the data shall be given as a variable-length vector containing opaque data. It is out of scope of this definition how resulting ciphertexts are transported. Typically, a ciphertext should be put into a `Payload` data structure marked as encrypted using the `PayloadType`.

NOTE: This structure is not available in IEEE 1609.2 Draft D12 [i.2].

## 4.2.8 CrlSeries

```
uint32 CrlSeries;
```

This number identifies a CRL series. The definition of a CRL series itself is outside the scope of the present document.

NOTE: This definition is identical to the one in IEEE 1609.2 Draft D12 [i.2], clause 6.3.21.

## 4.2.9 Signature

```
struct {
    PublicKeyAlgorithm algorithm;
    select(algorithm) {
    case ecdsa_nistp256_with_sha256:
        EcdsaSignature ecdsa_signature;
    unknown:
        opaque         signature<var>;
    }
} Signature;
```

This structure defines a container that encapsulates signatures based on public key cryptography. Depending on the value of `algorithm`, different data structures define the algorithm-specific details:

- `ecdsa_nistp256_with_sha256`: the signature contained in an `EcdsaSignature` structure shall be given.
- `unknown`: in all other cases, a variable-length vector containing the signature as opaque data shall be given.