NORME INTERNATIONALE

ISO 24534-4

Première édition
2010-07-15

AMENDEMENT 1
2019-02

Identification automatique des véhicules et des équipements — Identification d'enregistrement électronique (ERI) pour les véhicules —

Partie 4: Communications sécurisées à l'aide de techniques asymétriques

AMENDEMENT 1

Automatic vehicle and equipment identification — Electronic registration identification (ERI) for vehicles —

Part 4: Secure communications using asymmetrical techniques

AMENDMENT 1

Numéro de référence
ISO 24534-4:2010/Amd.1:2019(F)

ISO copyright office
Case postale 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Genève
Tél.: +41 22 749 01 11
Fax: +41 22 749 09 47
E-mail: copyright@iso.org
Web: www.iso.org

Publié en Suisse

Avant-propos

L'ISO (Organisation internationale de normalisation) est une fédération mondiale d'organismes nationaux de normalisation (comités membres de l'ISO). L'élaboration des Normes internationales est en général confiée aux comités techniques de l'ISO. Chaque comité membre intéressé par une étude a le droit de faire partie du comité technique créé à cet effet. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'ISO participent également aux travaux. L'ISO collabore étroitement avec la Commission électrotechnique internationale (IEC) en ce qui concerne la normalisation électrotechnique.

Les procédures utilisées pour élaborer le présent document et celles destinées à sa mise à jour sont décrites dans les Directives ISO/IEC, Partie 1. Il convient, en particulier de prendre note des différents critères d'approbation requis pour les différents types de documents ISO. Le présent document a été rédigé conformément aux règles de rédaction données dans les Directives ISO/IEC, (voir www.iso.org/directives).

L'attention est attirée sur le fait que certains des éléments du présent document peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. L'ISO ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et averti de leur existence. Les détails concernant les références aux droits de propriété intellectuelle ou autres droits analogues identifiés lors de l'élaboration du document sont indiqués dans l'Introduction et/ou dans la liste des déclarations de brevets reçues par l'ISO (voir www.iso.org/brevets).

Les appellations commerciales éventuellement mentionnées dans le présent document sont données pour information, par souci de commodité, à l'intention des utilisateurs et ne sauraient constituer un engagement.

Pour une explication de la nature volontaire des normes, la signification des termes et expressions spécifiques de l'ISO liés à l'évaluation de la conformité, ou pour toute information au sujet de l'adhésion de l'ISO aux principes de l'Organisation mondiale du commerce (OMC) concernant les obstacles techniques au commerce (OTC), voir www.iso.org/avant-propos.

Le présent document a été élaboré par le comité technique ISO/TC 204, *Systèmes de transport intelligents*.

Il convient que l'utilisateur adresse tout retour d'information ou toute question concernant le présent document à l'organisme national de normalisation de son pays. Une liste exhaustive desdits organismes se trouve à l'adresse www.iso.org/fr/members.html.

Identification automatique des véhicules et des équipements — Identification d'enregistrement électronique (ERI) pour les véhicules —

Partie 4:
Communications sécurisées à l'aide de techniques asymétriques

AMENDEMENT 1

Page 42 Clause 6.2.10.6

Remplacez la définition de CommissionErtErrors par:

```
CommissionErtErrors ErrorCode ::= {illegalArgument | illegalVehicleId |
illegalCertificate | illegalSignature | illegalEntity | illegalDate
| notCustomized | resourceLimitExceeded | noEnciphermentCapability |
secretKeyEncryptionAlgorithmNotSupported |
publicKeyEncryptionAlgorithmNotSupported | noSigningCapability |
hashingAlgorithmNotSupported | signingAlgorithmNotSupported | otherError, .. }
```


Page 42 Clause 6.2.11.1

Remplacez

```
withdrawCommissioningCode INTEGER ::=  9
```

par

```
withdrawCommissioningCode INTEGER ::=  14
```

Page 43 Clause 6.2.11.3

Remplacez

```
WithdrawCommissioningResultData ::= [APPLICATION withdrawCommissioningCode ]
SEQUENCE {
withdrawn WithdrawCommissioningArgument,
historicComData HistoricComData
}
```

par

```
WithdrawCommissioningResultData ::= SEQUENCE {
withdrawn WithdrawCommissioningArgument,
historicComData HistoricComData
}
```

Page 43 Clause 6.2.11.4

Remplacez la définition de WithdrawCommissioningError par:

```
WithdrawCommissioningErrors ErrorCode ::= {illegalArgument | illegalVehicleId
| illegalCertificate | illegalSignature | illegalEntity | illegalDate |
notCustomized | notCommissioned | otherError, ... }
```

Page 57 Clause 6.2.18.3.5

Remplacez la définition de EntityRole par:

```
EntityRole::= INTEGER {
   topLevelCertificationAuthority      (0),
   intermediateCertificationAuthority  (1),
   manufacturer                        (2),
   registrationAuthority               (3),
   authority                           (4),
   serviceProvider                     (5),
   eriHolder                           (6)
   } (0..7)
```


Page 66 Annexe A

Remplacez l'intégralité de l'annexe par l'annexe suivante:

Annexe A
(normative)

Module ASN.1

A.1 Vue d'ensemble

Les modules ASN.1 suivants sont spécifiés dans la présente annexe normative:

— Le module normatif **ElectronicRegistrationIdentificationTransactionsModule** {iso(1) standard(0) iso24534 (24534) transactions (2) version1 (1)} dans A.2.

— Le module informatif **AviEriDSRCData** {iso(1) standard(0) iso24534 (24534) transactions (2) example1(1) version1 (1)} dans A.3.

Si les spécifications ASN.1 fournies dans cette Annexe ne sont pas conformes aux illustrations ou aux spécifications fournies dans toute autre partie de la présente Norme internationale, les spécifications fournies dans cette Annexe prévalent.

Les modules ASN.1 figurant dans cette Annexe seront publiés sur le site Internet http://standards.iso.org/iso/24534/-4.

A.2 Module ElectronicRegistrationIdentificationTransactionsModule


```
ElectronicRegistrationIdentificationTransactionsModule {iso(1) standard(0) iso24534 (24534)
transactions (2) version1 (1)}
DEFINITIONS AUTOMATIC TAGS ::= BEGIN

IMPORTS
RegistrationAuthority, VehicleId, EriData, AdditionalEriData, EntityId FROM
ElectronicRegistrationIdentificationVehicleDataModule {iso(1) standard(0) iso24534 (24534)
vehicleData (1) version1 (1)} -- ISO 24534-3

EmbeddedERIType FROM AVIAEIDSRCInterfaceModule {iso (1) standard(0) iso17264(17264)
version1 (1)}
;

EriPdu ::= SEQUENCE{
    fill                BIT STRING (SIZE(6)),
    pdu                 CHOICE {
       requestPdu       EriRequestPdu,
       reponsePdu       EriResponsePdu
       }
    }

EriRequestPdu ::= SEQUENCE {
   transactCode TRANSACTION.&transactionCode ( {EriTransactions}),
   argument TRANSACTION.&ArgumentType ( {EriTransactions} {@.transactCode}) OPTIONAL
   }

EriResponsePdu ::= CHOICE {
   resultPdu                EriResultPdu,
   errorPdu                 EriErrorPdu
   }

EriResultPdu ::= SEQUENCE {
   transactCode TRANSACTION.&transactionCode ( {EriTransactions}),
   result TRANSACTION.&ResultType ( {EriTransactions} {@.transactCode})
   }
```

```
-- warning: Constraint checking for field 'error' would not be done at runtime as
constraints applied to FixedTypeValueSetFields or VariableTypeValueSetFields are not
currently supported.

EriErrorPdu ::= SEQUENCE {
   transactCode TRANSACTION.&transactionCode ( {EriTransactions}),
   error TRANSACTION.&ErrorCodes ( {EriTransactions} {@.transactCode})
   }

-- TRANSACTIONS
TRANSACTION ::= CLASS {
    &ArgumentType ,
    &ResultType ,
    &ErrorCodes ErrorCode OPTIONAL,
    &transactionCode TransactionCode UNIQUE
    }
    WITH SYNTAX {
       ARGUMENT &ArgumentType
       RESULT &ResultType
       [ERRORS &ErrorCodes]
       CODE &transactionCode
    }

TransactionCode ::= INTEGER  {
   tc-getEriData                              (1),
   tc-getAuthenticatedEriData                 (2),
   tc-setEriData                              (3),
   tc-getCiphertextHistoricEriData            (4),
   tc-getCleartextHistoricEriData             (5),
   tc-getPublicCertificateVerificationKeyId   (6),
   tc-getPublicEnciphermentKeyErt             (7),
   tc-commissionErt                           (8),
   tc-getCiphertextHistoricComData            (9),
   tc-getCleartextHistoricComData             (10),
   tc-updateAccessControlList                 (11),
   tc-getCiphertextAccessControlListEntry     (12),
   tc-getCleartextAccessControlListEntry      (13),
   tc-withdrawCommissioning                   (14),
   tc-getErtCapabilities                      (15)
   } (0..255)

EriTransactions TRANSACTION ::= {getEriData | getAuthenticatedEriData | setEriData |
getCiphertextHistoricEriData | getCleartextHistoricEriData |
getPublicCertificateVerificationKeyId | getPublicEnciphermentKeyErt | commissionErt |
withdrawCommissioning | getCiphertextHistoricComData |
getCleartextHistoricComData | updateAccessControlList |
getCiphertextAccessControlListEntry | getCleartextAccessControlListEntry |
getErtCapabilities, ... }

-- Get ERI data
getEriData TRANSACTION ::= {
   ARGUMENT GetEriDataArgument
   RESULT GetEriDataResult
   ERRORS {notCustomized}
   CODE tc-getEriData
   }

GetEriDataArgument ::= SEQUENCE {
   onBehalfOf               EntityId OPTIONAL,
   challenge                Challenge,
   includeAdditionalData    BOOLEAN
   }

GetEriDataResult ::= SEQUENCE {
   registrationAuthority    RegistrationAuthority OPTIONAL,
   eriResultData            SECURED {CleartextEriData}
   }

-- Authenticate ERI data
getAuthenticatedEriData TRANSACTION ::= {
   ARGUMENT GetAuthenticatedEriDataArgument
```

```
   RESULT GetAuthenticatedEriDataResult
   ERRORS {notCustomized}
   CODE tc-getAuthenticatedEriData
   }

GetAuthenticatedEriDataArgument ::= SEQUENCE {
   ertHolderCredentials     ErtHolderCredentials,
   challenge                Challenge,
   includeAdditionalData    BOOLEAN
   }

GetAuthenticatedEriDataResult ::= SEQUENCE {
   registrationAuthority    EntityId OPTIONAL,
   authenticateResultData   CLEAR-SECURED {CleartextEriData}
   }

-- ERI data and ERT security flags
CleartextEriData ::= SEQUENCE {
   eriDataOrId              EriDataOrId,
   ertSecurityStatus        ErtSecurityFlags OPTIONAL
   }

EriDataOrId ::= CHOICE {
   vehicleId                VehicleId,
   unsignedDatedEriData     DATED { EmbeddedERIType{EriData} },
   datedAndSignedEriData    SIGNED {DATED { EmbeddedERIType{EriData} },
PrivateSignatureKey} -- BOE signed
   }

/*
EriData ::= SEQUENCE {
   id                       VehicleId,
   additionalEriData        OCTET STRING (CONTAINING AdditionalEriData) OPTIONAL
   }
*/

ErtSecurityFlags ::= BIT STRING {
   flagsHaveBeenResetted    (0),
   notCommissioned          (1),
   lowSupplyVoltageIndication       (2),
   highSupplyVoltageIndication      (3),
   lowClockFrequencyIndication      (4),
   highClockFrequencyIndication     (5),
   lowTemperatureIndication         (6),
   highTemperatureIndication        (7)
   } (SIZE (0..16)) -- bit 8 .. 15 reserved for future use

   -- Set ERI data
setEriData TRANSACTION ::= {
   ARGUMENT SetEriDataArgument
   RESULT NULL
   ERRORS {SetEriDataErrors}
   CODE tc-setEriData
   }

SetEriDataArgument ::= CHOICE {
   clearTextArgument        ClearTextSetEriDataArgument,
   encryptedArgument        ENCRYPTED {ClearTextSetEriDataArgument}
   }

ClearTextSetEriDataArgument ::= CHOICE {
   authenticatedEriData     BOE-AUTHENTICATED{DATED{EmbeddedERIType{EriData} }},
   notAuthenticatedEriData  DATED { EmbeddedERIType{EriData} }
   }

SetEriDataErrors ErrorCode ::= {illegalArgument | illegalVehicleId | illegalCertificate |
illegalSignature | illegalDate | notCommissioned | resourceLimitExceeded | otherError, ...
}

-- Retrieve historic ERI data
getCiphertextHistoricEriData TRANSACTION ::= {
```