



BSI Standards Publication

Intelligent transport systems — ITS station security services for secure session establishment and authentication between trusted devices

National foreword

This British Standard is the UK implementation of EN ISO 21177:2024. It is identical to ISO 21177:2024. It supersedes [BS EN ISO 21177:2023](#), which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee EPL/278, Intelligent transport systems.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2024
Published by BSI Standards Limited 2024

ISBN 978 0 539 27483 7

ICS 03.220.01; 35.030; 35.240.60

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 March 2024.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

English Version

**Intelligent transport systems - ITS station security services
for secure session establishment and authentication
between trusted devices (ISO 21177:2024)**

Systèmes de transport intelligents - Services de
sécurité des stations ITS pour l'établissement et
l'authentification des sessions sécurisées entre
dispositifs de confiance (ISO 21177:2024)

Intelligente Verkehrssysteme - Sicherheitsdienste für
eine ITS-Station zum sicheren Aufbau von Sitzungen
und zur Authentisierung zwischen
vertrauenswürdigen Geräten (ISO 21177:2024)

This European Standard was approved by CEN on 6 March 2024.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and United Kingdom.



EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

European foreword

This document (EN ISO 21177:2024) has been prepared by Technical Committee ISO/TC 204 "Intelligent transport systems" in collaboration with Technical Committee CEN/TC 278 "Intelligent transport systems" the secretariat of which is held by NEN.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by September 2024, and conflicting national standards shall be withdrawn at the latest by September 2024.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN shall not be held responsible for identifying any or all such patent rights.

This document supersedes EN ISO 21177:2023.

Any feedback and questions on this document should be directed to the users' national standards body/national committee. A complete listing of these bodies can be found on the CEN website.

According to the CEN-CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.

Endorsement notice

The text of ISO 21177:2024 has been approved by CEN as EN ISO 21177:2024 without any modification.

Contents

Page

Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Abbreviated terms	2
5 Overview	4
5.1 General description, relationship to transport layer security (TLS) and relationship to application specifications	4
5.2 Goals	5
5.3 Architecture and functional entities	5
5.4 Cryptomaterial handles	10
5.5 Session IDs and state	10
5.6 Access control and authorization state	11
5.7 Application level non-repudiation	11
5.8 Service primitive conventions	11
6 Process flows and sequence diagrams	12
6.1 General	12
6.2 Overview of process flows	12
6.3 Sequence diagram conventions	13
6.4 Configure	14
6.5 Start session	15
6.6 Send data	18
6.7 Send access control PDU	21
6.8 Receive PDU	22
6.9 Extend session	27
6.9.1 Goals	27
6.9.2 Processing	28
6.10 Secure connection brokering	28
6.10.1 Goals	28
6.10.2 Prerequisites	28
6.10.3 Overview	29
6.10.4 Detailed specification	30
6.11 Force end session	38
6.12 Session terminated at session layer	39
6.13 Deactivate	40
6.14 Secure session example	41
7 Security subsystem: interfaces and data types	43
7.1 General	43
7.2 Access control policy and state	43
7.3 Enhanced authentication	44
7.3.1 Definition and possible states	44
7.3.2 States for owner role enhanced authentication	45
7.3.3 State for accessor role enhanced authentication	46
7.3.4 Use by access control	46
7.3.5 Methods for providing enhanced authentication	47
7.3.6 Enhanced authentication using SPAKE2	47
7.4 Extended authentication	48
7.5 Security Management Information Request	48
7.5.1 Rationale	48
7.5.2 General	49
7.6 Data types	50

7.6.1	General	50
7.6.2	Imports	50
7.6.3	“Helper” data types	50
7.6.4	Iso21177AccessControlPdu	51
7.6.5	AccessControlResult	51
7.6.6	ExtendedAuthPdu	51
7.6.7	ExtendedAuthRequest	52
7.6.8	InnerExtendedAuthRequest	52
7.6.9	AtomicExtendedAuthRequest	53
7.6.10	ExtendedAuthResponse	53
7.6.11	ExtendedAuthResponsePayload	53
7.6.12	EnhancedAuthPdu	53
7.6.13	SpakeRequest	54
7.6.14	SpakeResponse	54
7.6.15	SpakeRequesterResponse	54
7.6.16	SecurityMgmtInfoPdu	54
7.6.17	SecurityMgmtInfoRequest	55
7.6.18	EtsiCrlRequest	55
7.6.19	CertChainRequest	55
7.6.20	SecurityMgmtInfoResponse	56
7.6.21	SecurityMgmtInfoErrorResponse	56
7.6.22	EtsiCrlResponse	56
7.6.23	EtsiCtlResponse	56
7.6.24	IeeeCrlResponse	57
7.6.25	CertChainResponse	57
7.6.26	SessionExtensionPdu	57
7.7	App-Sec Interface	59
7.7.1	App-Sec-Configure.request	59
7.7.2	App-Sec-Configure.confirm	60
7.7.3	App-Sec-StartSession.indication	60
7.7.4	App-Sec-Data.request	60
7.7.5	App-Sec-Data.confirm	61
7.7.6	App-Sec-Incoming.request	61
7.7.7	App-Sec-Incoming.confirm	62
7.7.8	App-Sec-EndSession.request	63
7.7.9	App-Sec-EndSession.indication	63
7.7.10	App-Sec-Deactivate.request	63
7.7.11	App-Sec-Deactivate.confirm	64
7.7.12	App-Sec-Deactivate.indication	64
7.8	Security subsystem internal interface	64
7.8.1	General	64
7.8.2	Sec-AuthState.request	65
7.8.3	Sec-AuthState.confirm	65
8	Adaptor layer: interfaces and data types	66
8.1	General	66
8.2	Data types	67
8.2.1	General	67
8.2.2	Iso21177AdaptorLayerPDU	67
8.2.3	Apdu	68
8.2.4	AccessControl	68
8.2.5	TlsClientMsg1	68
8.2.6	TlsServerMsg1	68
8.3	App-AL Interface	68
8.3.1	App-AL-Data.request	68
8.3.2	App-AL-Data.confirm	69
8.3.3	App-AL-Data.indication	69
8.3.4	App-AL-EnableProxy.request	70
8.4	Sec-AL Interface	71

8.4.1	Sec-AL-AccessControl.request	71
8.4.2	Sec-AL-AccessControl.confirm	72
8.4.3	Sec-AL-AccessControl.indication	72
8.4.4	Sec-AL-EndSession.request	73
8.4.5	Sec-AL-EndSession.confirm	73
9	Secure session Services	73
9.1	General	73
9.2	App-Sess interfaces	73
9.2.1	App-Sess-EnableProxy.request	73
9.3	Sec-Sess interface	74
9.3.1	Sec-Sess-Configure.request	74
9.3.2	Sec-Sess-Configure.confirm	76
9.3.3	Sec-Sess-Start.indication	76
9.3.4	Sec-Sess-EndSession.indication	77
9.3.5	Sec-Sess-Deactivate.request	77
9.3.6	Sec-Sess-Deactivate.confirm	78
9.4	AL-Sess interface	78
9.4.1	AL-Sess-Data.request	78
9.4.2	AL-Sess-Data.confirm	78
9.4.3	AL-Sess-Data.indication	78
9.4.4	AL-Sess-EndSession.request	79
9.4.5	AL-Sess-EndSession.confirm	79
9.4.6	AL-Sess-ClientHelloProxy.request	79
9.4.7	AL-Sess-ClientHelloProxy.indication	80
9.4.8	AL-Sess-ServerHelloProxy.request	81
9.4.9	AL-Sess-ServerHelloProxy.indication	81
9.5	Permitted mechanisms	82
9.5.1	TLS 1.3	82
9.5.2	DTLS 1.3	83
	Annex A (informative) Usage scenarios	84
	Annex B (normative) ASN.1 module	92
	Annex C (normative) Session extension PDU functional type	93
	Annex D (normative) Owner authorization	94
	Bibliography	98

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*, in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/TC 278, *Intelligent transport systems*, in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

This second edition cancels and replaces the first edition (ISO 21177:2023), of which it constitutes a minor revision. The changes are as follows:

- cross-references to RFC 8942 have been updated to RFC 8902 throughout the document;
- information concerning patent(s) required for the implementation of this document has been moved from the Introduction to the Foreword.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

This document specifies ITS station security services that provide authenticity of the source and confidentiality and integrity of application activities taking place between trusted devices. The two devices taking part in a data exchange establish a cryptographically secure session. As part of establishing this session, each device [or, more precisely, each end entity (EE) which is an application on the device] is sent one or more digital certificates that are cryptographically bound to the other EE and contain statements, made by a trusted third party, about the EE's capabilities, properties and permissions. This allows each EE to have assurance about the properties of the other EE in the session, and this in turn allows each EE to make trust and access control decisions about data that the other EE can access, commands that the other EE can execute, states that the other EE can change, and other types of access that the other EE can request. In other words, the two EEs establish a trust relationship where each EE is trusted by the other EE to carry out specific actions, without requiring one EE to allow the other EE to have arbitrary access.

The mechanisms specified in this document allow each EE to establish trusted facts about the other EE. For these mechanisms to be used, the EE specification needs to include an access control policy, indicating which properties are required to be known to be true about the other EE for that other EE to be allowed to carry out particular actions. In other words, this document provides a means to obtain security-relevant information, but the use of that security-relevant information is to be specified in the specification of the EE.

The trust relation between two devices is illustrated in [Figure 1](#). Two devices cooperate in a trusted way, i.e. exchange information with optional explicit bi-directional protection.

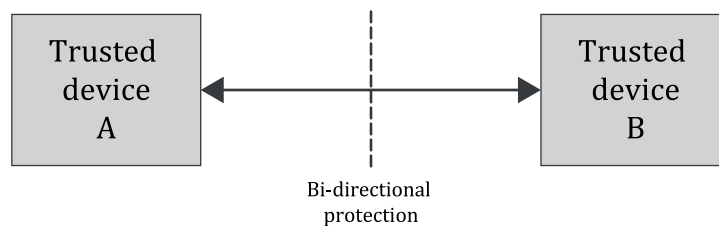


Figure 1 — Interconnection of trusted devices

According to ISO 21217, an ITS station unit (ITS-SU), i.e. the physical implementation of the ITS station (ITS-S) functionality, is a trusted device, and an ITS-SU may be composed of ITS station communication units (ITS-SCUs) that are interconnected via an ITS station-internal network. Thus, an ITS-SCU is the smallest physical entity of an ITS-SU that is referred to as a trusted device.

NOTE 1 ISO 21217 fully covers the functionality of EN 302 665,^[16] which is a predecessor of ISO 21217.

NOTE 2 An ITS-SU can be composed of ITS-SCUs from different vendors where each ITS-SCU is linked to a different ITS-SCU configuration and management centre specified in ISO 24102-2 and ISO 17419. Station-internal management communications between ITS-SCUs of the same ITS-SU are specified in ISO 24102-4. The European C-ITS regulation refers to the "ITS-SCU configuration and management centre" as "C-ITS station operator" meaning the entity responsible for the operation of a C-ITS station. The C-ITS station operator can be responsible for the operation of one single C-ITS station (fixed or mobile), or a C-ITS infrastructure composed of a number of fixed C-ITS stations, or a number of mobile ITS stations.

Four implementation contexts of communication nodes in ITS communications networks are identified in the ITS station and communication architecture of ISO 21217, each comprised of ITS-SUs taking on a particular role: personal, vehicular, roadside or central. These ITS-SUs are ITS-secured communication nodes as required in ISO 21217 that participate in a wide variety of ITS services related to, for example, sustainability, road safety and transportation efficiency. See also [Figure 2](#), [Figure 3](#), [Figure 4](#) and [Figure 5](#).

Over the last decade, ITS services have arisen that require secure access to data from sensor and control networks (SCN), for example, from in-vehicle networks (IVN) and from infrastructure/roadside networks (IRN), some of which require secure local access to time-critical information; see [Figure 2](#) and [Figure 3](#).

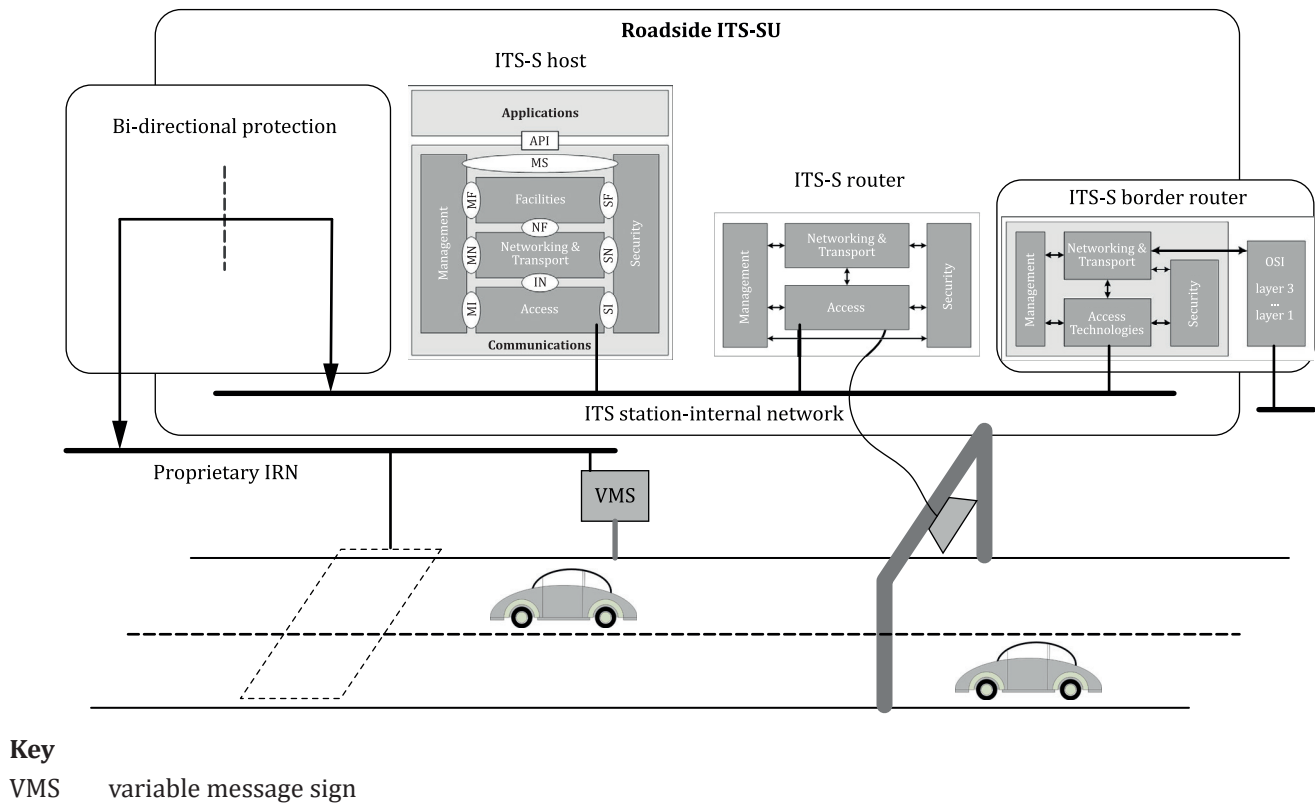


Figure 2 — Example of a roadside ITS-SU connected with proprietary IRN

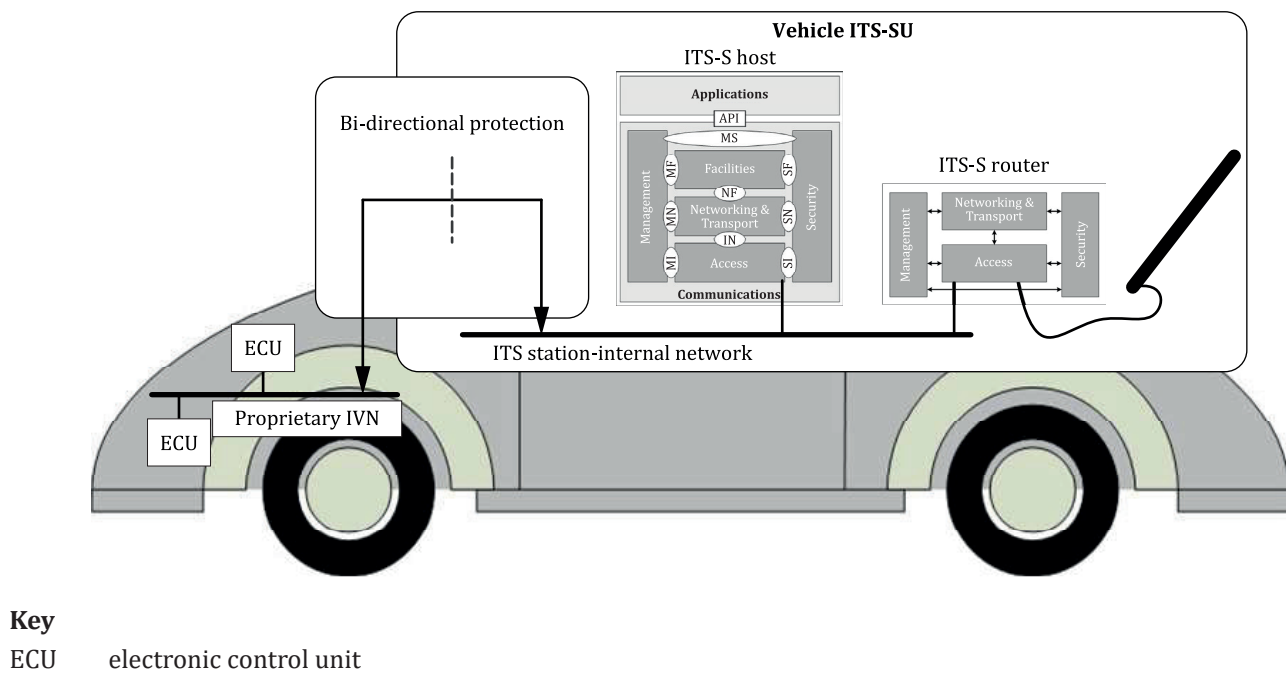
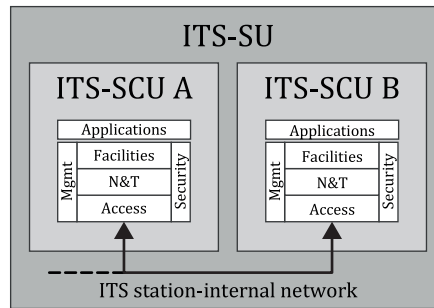


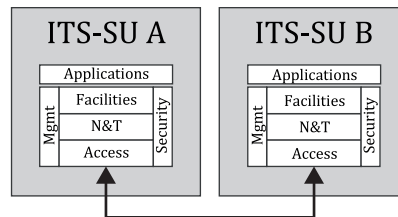
Figure 3 — Example of a vehicle ITS-SU connected with proprietary IVN



Key

N&T Networking & Transport

Figure 4 — Interconnection of ITS-SCUs in an ITS-SU



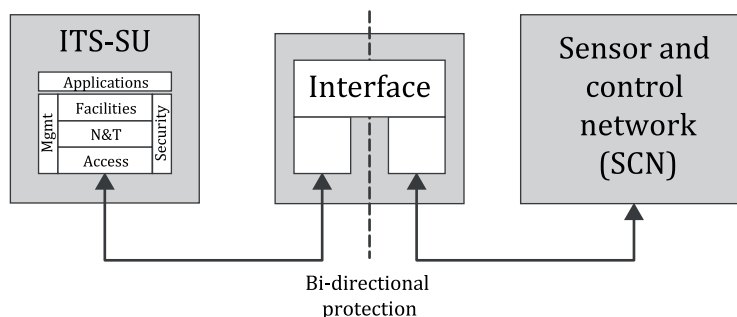
Key

N&T Networking & Transport

Figure 5 — Interconnection of ITS-SUs

By applying basic security means specified in this document, the ITS-SUs can establish secure application sessions. Establishment of sessions either requires prior knowledge about a session partner or can be achieved by means of a service announcement as specified in ISO 22418. Further on, the broadcasting of messages is secured by means of authenticating the sender of such a message, applicable for the service advertisement message (SAM) specified in ISO 16460 and used in ISO 22418. Additionally, other security means may be applied, e.g. encryption of messages.

A further trust relation in the ITS domain is between an ITS-SU consisting of one or several ITS-SCUs and a sensor and control network (SCN). Trust is achieved by applying security means in an interface as illustrated in [Figure 6](#) with details specified in this document.



Key

N&T Networking & Transport

Figure 6 — Interface between ITS-SU and sensor and control network

The interface presented in [Figure 6](#) may be a stand-alone device, or may be integrated in the ITS-SU, or may be part of the SCN. Examples of SCNs are "in-vehicle networks" (IVN) and "infrastructure/roadside networks" (IRN).

Related use cases of these ITS services have largely been derived from regulatory requirements and ITS operational needs, and they include:

- secure real-time access to time-critical vehicle-related data for safety of life and property applications, e.g. collision avoidance, emergency electronic brake light and event determination;
- secure local access to detailed real-time data for efficiency applications (traffic management), e.g. intersection interaction, congestion avoidance and dynamic priorities;
- protection of private data, e.g. in compliance with the European "General Data Protection Regulation" (GDPR);^[18]
- local access to certified real-time data for sustainability applications, e.g. dynamic emission zones (controlled zones as currently standardized by CEN/TC 278), intersection priorities based on emissions, and interactive optimum vehicle settings to minimize fuel consumption.

There are many use cases of ITS services currently identified where real-time exchange of time-critical information between ITS-SUs in close proximity is essential, and this number will grow (see the US National ITS Reference Architecture,^[19] for example). It is critical that ultimately all ITS-SUs in a given area be able to be engaged in these distributed services. This, in turn, requires vehicle ITS-SUs to have real-time access to vehicle data, and roadside ITS-SUs to have real-time access to infrastructure data. All ITS-SUs need to be capable of secure software updates.

According to ISO 21217, an ITS-SCU of an ITS-SU can communicate with devices that, in a strict sense, are not compliant with the architecture specified in ISO 21217. However, in order to have trusted communications, a certain minimum level of security measures need to be shared between an ITS-SCU and such an external device. Examples of such external devices are a node in the Internet, or a node in a sensor and control network. In this document, the assumption is made that ITS-S application processes operating on ITS-SUs are issued with certificates by a Certificate Authority (CA), and that the CA is a trusted third party in the sense that before issuing the certificate to the ITS-S application process, it ensures that the ITS-SU on which the ITS-S application process is resident meets the minimum security requirements for that application. This allows peer ITS-S application processes which observe that an ITS-S application process possesses a valid certificate to have a level of assurance that the ITS-S application process is in fact secure and trustworthy.

The subject of this document thus is three-fold.

- 1) Specification of ITS station security services for enabling trust between ITS-S application processes running on different ITS-SCUs of the same ITS-SU, i.e. establishing a trusted processing platform, considering also trust inside an ITS-SCU:
 - protection of applications from the actions of other applications;
 - protection of shared information;
 - protection of shared processing resources such as communications software and hardware, which includes methods of prioritization and restricted access.
- 2) Specification of ITS station security services for enabling trust between ITS-S application processes running on the same ITS-SU.
- 3) Extension of these ITS security services for enabling trust between an ITS-SCU and devices being part of a sensor and control network.

Such security services include, for example, the basic security features of:

- a) authentication and authorization;
- b) confidentiality and privacy;

- c) data integrity;
- d) non-repudiation.

Tasks related to communications are:

- e) establishing secure sessions for bi-directional communications, e.g. based on service advertisement as specified in ISO 22418;
- f) authenticating a sender of broadcast messages, e.g. CAM, DENM, BSM, SPaT, MAP, FSAM, WSA, etc.;
- g) encrypting messages.

NOTE 3 Tasks f) and g) above related to communications are already specified in other standards: see IEEE 1609.2 and several related standards from ETSI, for example.

Intelligent transport systems — ITS station security services for secure session establishment and authentication between trusted devices

1 Scope

This document contains specifications for a set of ITS station security services required to ensure the authenticity of the source and integrity of information exchanged between trusted entities, i.e.:

- between devices operated as bounded secured managed entities, i.e. "ITS Station Communication Units" (ITS-SCU) and "ITS station units" (ITS-SU) as specified in ISO 21217; and
- between ITS-SUs (composed of one or several ITS-SCUs) and external trusted entities such as sensor and control networks.

These services include the authentication and secure session establishment which are required to exchange information in a trusted and secure manner.

These services are essential for many intelligent transport system (ITS) applications and services, including time-critical safety applications, automated driving, remote management of ITS stations (ISO 24102-2), and roadside/infrastructure-related services.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ETSI TS 102 941, *Intelligent Transport Systems (ITS); Security; Trust and privacy management*

ETSI TS 103 097, *Intelligent Transport Systems (ITS); Security; Security header and certificate formats*

IEEE 1609.2, *including Amendment 1, IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages*

IEEE 1003.1:2017, *IEEE Standard for Information Technology--Portable Operating System Interface (POSIX(R)) Base Specifications, Issue 7*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1

access control PDU

protocol data unit (PDU) generated by the security subsystem for purposes of establishing the authorization status of a peer ITS-S application process

3.2

access control policy

data source governing what access to resources is permissible by peer applications

3.3

application

functional entity, i.e. an ITS-S application process

Note 1 to entry: The natural language term "application" is used in this document as a synonym to the term "ITS-S application process" specified in ISO 21217.

3.4

cryptomaterial

cryptographic keys and associated material

Note 1 to entry: Cryptomaterial refers to either a secret key for a symmetric algorithm, or a private key for an asymmetric algorithm and the associated public key or certificate.

3.5

cryptomaterial handle

reference to cryptomaterial allowing that cryptomaterial to be used in cryptographic operations (sign, verify, encrypt, decrypt)

3.6

handshake ITS-AID

ITS application identifier (ITS-AID) that appeared in the SignedData HeaderInfo in the signed transport layer security (TLS) messages within the TLS handshake that started a secure session

3.7

resources

functional entity constituting endpoints of ITS-S application process activity

3.8

security subsystem

functional entity providing security functionality for use by an ITS-S application process

3.9

security adaptor layer

functional entity providing multiplexing and demultiplexing functionality for data and session control commands

3.10

secure session service

functional entity providing confidentiality, integrity, authentication, guaranteed in-order delivery and replay protection on the datagrams that are passed over it

4 Abbreviated terms

ACK	acknowledge
ALPDU	adaptor layer protocol data unit
APDU	application protocol data unit
ASD	aftermarket safety device
ASN.1	abstract syntax notation 1
BSM	basic safety message

CA	certification authority
CAM	cooperative awareness message
C-ITS	cooperative intelligent transport systems
C-OER	canonical octet encoding rules
CRL	certificate revocation list
DENM	decentralized environmental notification message
DTLS	datagram transport layer security
ECU	electronic control unit
EE	end entity
FSAM	fast service advertisement message
ID	identifier
IRN	infrastructure/roadside network
ITS	intelligent transport systems
ITS-AID	ITS application identifier (PSID and ITS-AID share a common number space)
ITS-S	ITS station
ITS-SCU	ITS station communication unit
ITS-SU	ITS station unit
IVN	in-vehicle network
MAP	(digital) map (message)
NYC	New York city
OCSP	online certificate status protocol
OSI	open system interconnection
OTP	one-time password
PB-AKE	password based authenticated key exchange
PDU	protocol data unit
PIN	personal identification number
PSID	provider service identifier (PSID and ITS-AID share a common number space)
RSU	roadside unit
SAM	service advertisement message
SAP	service access point
SCN	sensor and control network

SDEE	secure data exchange entity
SPAKE2	secure password authenticated key exchange 2
SPaT	signal phase and timing (message)
SPDU	signed protocol data unit
SRM	service response message
SSP	service specific permission
SSTD	secure session between trusted devices
TCP	transmission control protocol
TIM	traveller information message
TLM	trust list manager
TLS	transport layer security
TMC	traffic management centre
VMS	variable message sign
WAVE	wireless access in vehicular environment
WSA	WAVE service announcement

5 Overview

5.1 General description, relationship to transport layer security (TLS) and relationship to application specifications

This document specifies a secure communications mechanism for secure sessions between ITS station units (ITS-SUs). The sessions are secure in that data within the session is protected against eavesdroppers (encryption) and protected against modification (integrity) and that the participants in the session know cryptographically-assured information about each other that can be used by one participant to determine which access privileges to grant to the other participant.

The secure communications sessions are based on transport layer security (TLS) 1.3, where digital certificates and secured protocol data units as specified in IEEE 1609.2 are incorporated into the TLS handshake as specified in RFC 8902.

TLS is a secure session protocol above the transport layer in the OSI model, and the endpoints of the ISO 21177 session are applications on the endpoint ITS-SUs. From the point of view of the applications, the use of ISO 21177 for secure communications is identical to the use of TLS over TCP and PDU handling for transport over the ISO 21177 session is the same as PDU handling for transport over a TLS session. The main differences between ISO 21177 and TLS are that ISO 21177:

- integrates the use of IEEE 1609.2 certificates, and
- provides mechanisms for one party to request additional certificates from the other party.

The mechanisms specified in this document allow each end entity (EE) to establish trusted facts about the other EE. These trusted facts are known as the authorization state of the other EE (in the context of that application session). However, the use of the authorization state is out of scope for this document, as this use will be specific to the specific application. The application specification itself is the appropriate home for a specification of the access control policy that maps known facts about the other party in the session to permissions of that other party. The application specification is also the home for specifications of customizations of the IEEE 1609.2

certificates used by instances of that application, via the use of the service specific permissions (SSP) field associated with the ITS-AID in the certificate. The approach anticipated is that:

- the SSP specification for the application will specify semantics of the SSP (for example, “if bit 1 is set the sender can take action A”, or “if bit 1 is set the sender is a police car”, or “if bit 1 is set the sender is blue”);
- the access control policy will map from the semantics of the SSP to permitted activities (for example, “if the SSP says the holder can take action A, then the holder can take action A”, or “if the SSP says the holder is a police car, then the holder can take action B”, or “if the SSP says the holder is blue, then the holder can take action C”).

The access control policy may also key off other properties in the certificate, for example in the certificateId field. It may also key off other information, such as whether the party granting access is in motion or at rest. All of this is to be included in the application specification for that application.

5.2 Goals

[Clause 5](#) presents the logical architecture followed in this document. The logical architecture is designed to accomplish the following goals.

- Two peer ITS-S application processes can communicate securely, i.e. in an authorized, integrity-protected and confidential manner.
- The ITS-S application processes can authenticate to each other using role- or attribute-based access control.
- Each individual incoming application protocol data unit (APDU) can be subject to individual access control processes.
- The security state of the connection (i.e. the authentication status of one ITS-S application process with respect to access to the other connection) can be updated within the secure session as follows:
 - an ITS-S application process can prove to the other that it knows a shared secret (enhanced authentication, see [7.3](#)) – the intended use of this is to allow the owner or other legitimate operator of one ITS-S application process to permit access by a specific peer ITS-S application process (see [Clause 6](#) for further discussion);
 - an ITS-S application process can provide additional authentication within the secure session (extended authentication, see [7.4](#)) – for example to provide an identity as well as application permissions, or to provide additional application permissions.
- Secure session communication uses the same credentials as ITS-S application processes use.
- An ITS-S application process can configure a secure session so that it terminates under specific conditions (e.g. timeouts of different kinds) and can also terminate the secure session directly.
- To allow secure and private service discovery, the initialization stage (the “handshake”) of one secure session between two peer ITS-S application processes can be proxied over an existing secure session between two different peers.

5.3 Architecture and functional entities

The protocols specified in this document are for secure communications between two peer applications on different ITS-Ss. The applied model is that each application on an ITS-S has access to resources, some of which are specific to that application, and some of which are shared with (i.e. can be accessed and potentially modified by) other processes on the ITS-S. The two application instances on different ITS-Ss that communicate using the protocol specified in this document are referred to as the home and the peer application instances. The secure communications between the peer and the home application instances can be considered a series of requests by the peer application to access the local resources: for example, to read a value, or to update the local state, or to carry out a financial transaction by accessing and modifying financial information and metadata stored by the home application. The goal of the protocols is to give the

home application instance the information it needs about the peer application instance to make a decision as to whether or not to allow the requested access to the local resources.

This document uses the following definitions; see also [Clause 3](#):

- **Resources:** These are endpoints of ITS-S application process activity and may be inside or outside the ITS-SU or ITS-SCU on which the ITS-S application process resides.
- **Application:** This is an ITS-S application process that uses input from resources, from a peer ITS-S application process (peer application), and from its own state to carry out application activities.

NOTE To simplify reading of this document, the term "application" is used as a replacement for "ITS-S application process", where appropriate.

[Figure 7](#) shows the functional entities and data sources involved in secure communications between pairs of ITS-S application processes; i.e. between EEs, using mechanisms specified in this document. A pair of ITS-S application processes can reside:

- on the same ITS-SCU, or
- in different ITS-SCUs of the same ITS-SU, or
- in different ITS-SUs, or
- in an ITS-SU and another trusted entity not being a dedicated ITS device.

The description in this document refers in general to activities on the "home trusted entity" and the "peer trusted entity". Details of functional blocks in an ITS-S are shown in [Figure 8](#).

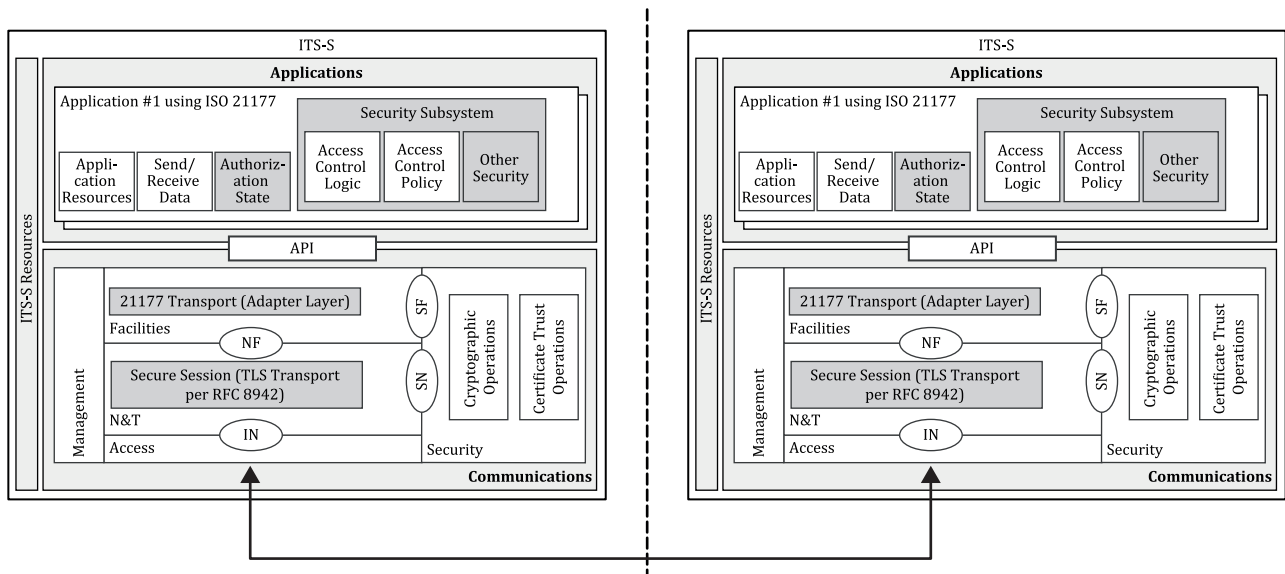


Figure 7 — Logical Architecture with two ITS-SUs

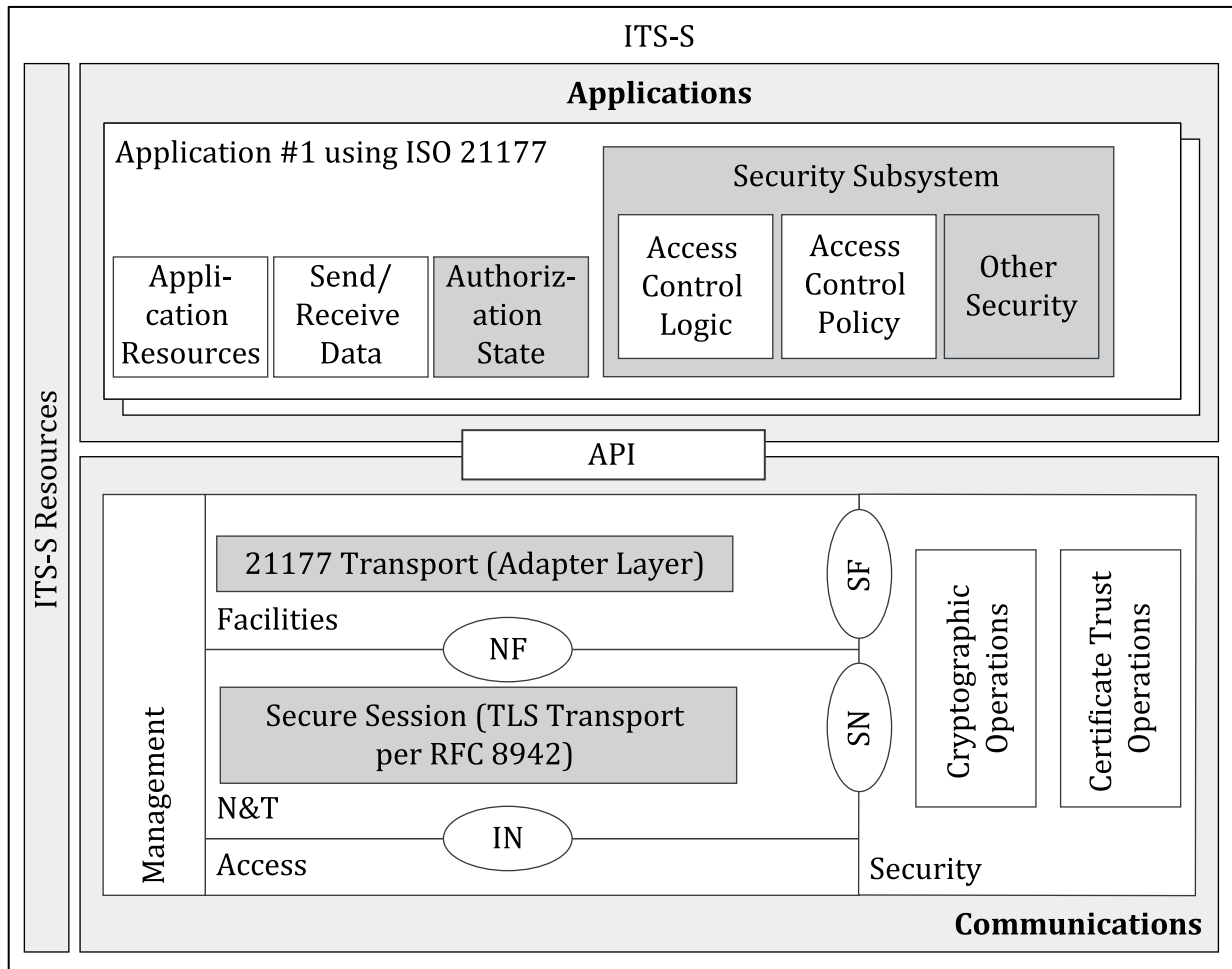


Figure 8 — Logical Architecture within the ITS-S architecture

Figure 8 shows the allocation of functional entities involved in communicating using the mechanisms of this document in the ITS station architecture specified in ISO 21217. The functional entities shaded in grey are the subject of this document. It is to be noted that several applications may use simultaneously the functionalities specified in this document; to simplify the graphical presentation, only one such application is fully presented.

The ITS-S resources are those resources of an ITS-SU that the home application can access and that are also accessed by other applications or processes on the same ITS-SU. Note that the term ITS-SU indicates a physical implementation of the ITS-S functionality; see ISO 21217.

There may also be application-specific resources that are not necessarily shared with other applications.

The functionalities "21177 Transport" and "Secure Session" specified in this document allow secure communication of the authorization properties of the peer application instances.

- The functionality of "Secure Sessions (TLS Transport as per RFC 8902)" is located in the ITS-S networking & transport layer.
- The functionality "21177 Transport (Adapter Layer)" is located in the ITS-S facilities layer.

The functionality "Authorization State" stores those authorization properties.

- The functionality "Authorization State" is located inside the application using the functionalities of this document. An equally suited alternative location would be in the IST-S facilities layer; there serving all applications that use the functionalities of this document.

Note that for an implementation of the functionality of this document, the location of functionalities in the architecture is not at all important, and thus, the presented location is only for illustration purposes.

The functional entities have the following responsibilities:

- **Authorization State:** Stores authorization information about the peer entity as well as metadata such as when the authorization information was received.
- **ISO 21177 Transport (Adaptor Layer):** This is a multiplexer/demultiplexer that allows both data (i.e. communications between the applications themselves) and session control commands (i.e. communications between peer instances of security subsystem or the security adaptor layer) to be sent over the same secure session. See [Clause 8](#).
- **Secure Sessions (TLS Transport as per RFC 8902):** This provides confidentiality, integrity, authentication, guaranteed in-order delivery, and replay protection on the datagrams that are passed over it. In this version of this document there are two types of secure session:
 - **Cryptographic secure session:** This uses cryptography to achieve the listed security properties. Any secure session which passes outside the secure boundary of the ITS-SU shall be a cryptographic secure session. In this document the only supported cryptographic secure session mechanism is TLS 1.3 with certificates specified in IEEE 1609.2. See [Clause 9](#).
 - **Physical secure session:** This is a session between two applications running in the same ITS-SU, i.e. the information flow does not pass outside the ITS-SU secure boundary. In this case, because the ITS-SU is a trusted domain, all the security properties listed above are assumed to hold. This document does not provide a specification of a physical secure session but permits the use of a physical secure session.

The components of the application are the application resources, the “Send/Receive Data” functional entity, and the functional entities within the “Security subsystem” component of the application. These functional entities play the following roles:

- **Application Resources:** resources accessed only by that application instance.
- **Send/Receive Data:** Functionality that sends and receives data.
- **Security Subsystem:** Security functionality that includes the following functional entities:
 - **Access Control Logic:** Functionality that determines whether a particular resource access request by the peer application instance will be granted.
 - **Access Control Policy:** The specification of the criteria used by access control Logic to make an access decision.
 - **Other Security:** Other security operations, for example signing or verifying individual PDUs.
- **ITS-S Resources:** general resources of an ITS station unit accessible by different applications.

The relationships between instances of these functional elements and data sources on a single trusted entity are shown in [Figure 9](#). In [Figure 9](#), heavy arrows between functional elements indicate flows containing information that is exchanged with processes on the peer trusted entity, and light arrows with dots between functional elements indicate control flows within the home ITS-SU. Grey dashed arrows within functional elements indicate relationships between different data flows handled by those functional elements.

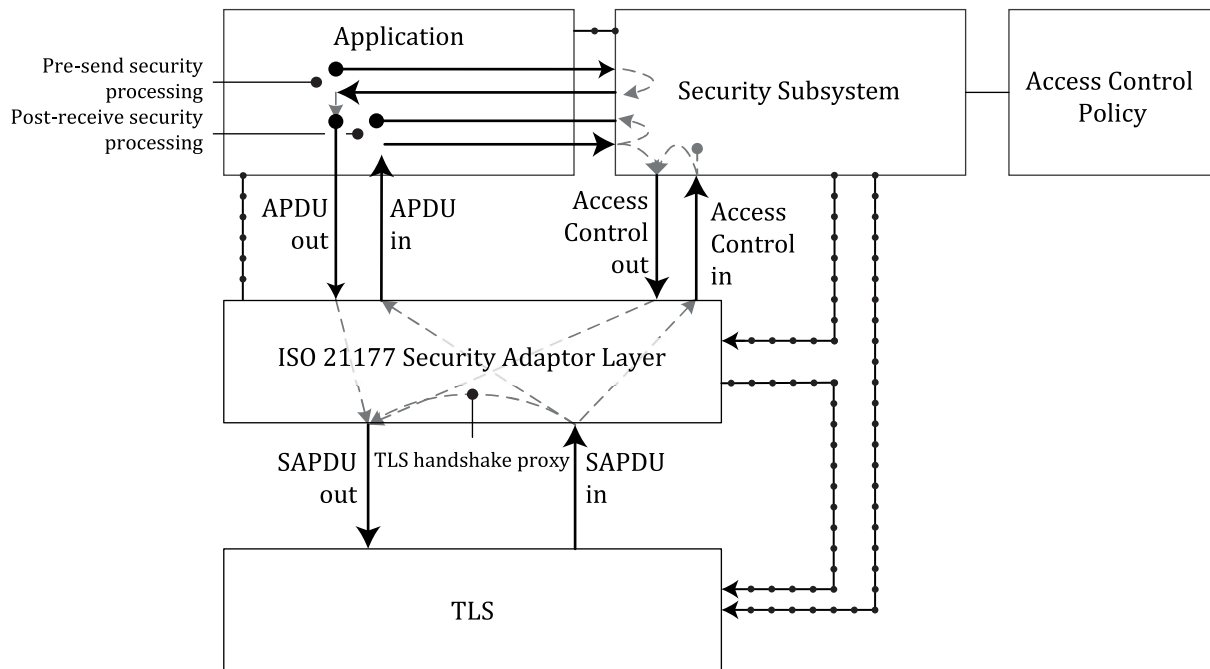


Figure 9 — Interactions between local functional elements

- The **Application** acts as a source and a sink for APDUs.
- The **application-security subsystem interface** is specified in [7.7](#). It is used:
 - optionally, by the application, to request IEEE 1609.2 application-level processing for outgoing APDUs specified in IEEE 1609.2 as described in [6.6](#);
 - by the application, to request that access control services are applied to activities that can potentially be carried out as a result of incoming APDUs. Specifically, in this document, the interface provided is an internal interface between the authorization state subsystem and the access control subsystem that allows the access control subsystem to request the authorization state. A full application specification also contains a specification of the interface between the application and the access control subsystem that allows the application to request access control decisions and provide full context. This second interface is application-specific and is not provided in this document;
 - by the application, to configure, end and deactivate secure session services;
 - by the security subsystem, to notify the application of a decision to end or deactivate secure session services.
- The **"Security Subsystem"** is specified in [Clause 7](#). It acts as:
 - a source for outgoing access control commands, which may be generated without a trigger or in response to incoming access control commands or APDUs;
 - a sink for incoming access control commands, which can result in the security subsystem updating its state or in the security subsystem generating outgoing access control commands (or both);
 - a source and a sink for security configuration commands.
- The **security subsystem-adaptor layer interface** is specified in [8.4](#). It is used to exchange APDUs and access control PDUs between the adaptor layer and the other functional entities, and for the application or security subsystem to control and configure the adaptor layer.

- The **ISO 21177 Security Adaptor Layer** is specified in [Clause 8](#). It:
 - wraps outgoing APDUs and outgoing access control PDUs as security adaptor layer protocol data units (ALPDUs) and passes them to the secure session service;
 - acts as a proxy for TLS handshake PDUs as specified in [6.10](#);
 - inspects incoming ALPDUs and
 - unwraps incoming APDUs and incoming access control PDUs and passes them to the security subsystem,
 - proxies incoming TLS handshake proxy PDUs for TLS brokering as specified in [6.10.4](#).
- The **security subsystem-secure session service interface** is specified in [9.2](#). It is used to control and configure persistent properties of the secure session service.
- The **adaptor layer-secure session service interface** is specified in [9.3](#). It is used to control and configure session-specific properties of the secure session service.
- The **secure session service** is specified in [Clause 9](#). In this version of this document, it is required to be based on TLS, DTLS, or physical secure sessions. It:
 - receives ALPDUs from the security adaptor layer and treats them as application input to record layer operations, performing outgoing TLS processing on them;
 - receives secure session datagrams from the network stack and performs incoming TLS processing on them, resulting in ALPDUs which the TLS service passes to the security adaptor layer.

5.4 Cryptomaterial handles

This document uses the concept of cryptomaterial handles specified in IEEE 1609.2 to represent logical access to cryptographic keys and, if appropriate, their associated certificates. See IEEE 1609.2 for more details.

5.5 Session IDs and state

The security subsystem service, adaptor layer service, and secure sessions service maintain state information of each (application, session)-tuple.

NOTE The interfaces presented in this document are passing (application ID, session ID)-tuples to the relevant primitives.

Unique identification of an instance of an ITS-S application process of an ITS application is by means of the identifier ITS-SAPIID of ASN.1 type ITSapiid specified in ISO 17419.

An implementation shall ensure that state information corresponding to a specific (application, session)-tuple is not made available for use by activities corresponding to a different (application, session)-tuple except through the interfaces specified in this document.

In the case where an application has the role of the client in a secure session, the session ID is generated by this application. The application ensures that the session ID is unique among active sessions for that application. The application supplies the session ID to the client via the `App-Sec-Configure.request` and `Sec-Sess-Configure.request` primitives.

In the case where an application has the role of the server in a secure session, the session ID is generated by the server in response to an incoming connection. The server ensures that the session ID is unique among active sessions for that application. The session ID is provided to the application using the `Sec-Sess-StartSession.indication` and `App-Sec-StartSession.indication` primitives.

5.6 Access control and authorization state

The security subsystem maintains the authorization state of the session. The authorization state is the collection of all “authorization statements” made by the peer party which were valid at the time made and which have not timed out (where the timeout conditions are set by the access control policy). An “authorization statement” is any PDU sent by the peer party that contains cryptographically protected information about the authorization of that peer party. Specifically, in this document, the authorization statements presented in [Table 1](#) are specified.

Table 1 — Authorization statements

Type	Description	Related sub-clauses	Service primitives for provision to security subsystem
Initial authorization	The credentials presented during the handshake of the secure session, if one occurs	6.5	Sec-Sess-StartSession.indication
Enhanced authorization	A demonstration that the peer party shares a secret value with the home party, presented during the secure session via an access control PDU of type EnhancedAuthPdu as specified in 7.6.12	6.7 , 6.8 , 7.3	Sec-AL-AccessControl.indication
Extended authorization	Additional credentials presented during the secure session via an access control PDU of type ExtendedAuthPdu as specified in 7.6.6 .	6.7 , 6.8 , 7.4	Sec-AL-AccessControl.indication

This document specifies access control request PDUs and access control response PDUs in [7.6](#).

An access control PDU containing an extended authorization statement may be generated in response to an access control Request PDU or may be generated without a request. An access control PDU containing an enhanced authorization statement is always made in response to an access control PDU containing an enhanced authorisation request. Sending access control PDUs is specified in [6.7](#). Receiving access control PDUs is specified in [6.8](#).

Exactly how the authorization statements are used by the security subsystem and governed by the access control policy is to be specified in the specification of an application that uses the mechanisms provided by this document.

5.7 Application level non-repudiation

An application may apply non-repudiation to individual application PDUs by signing them with the mechanisms of IEEE 1609.2. An application that applies non-repudiation to individual APDUs shall format any non-signed APDUs as `Ieee1609Dot2Data` structures of type unsecured, to ensure consistent interpretation of the incoming APDUs. A detailed specification of this process is provided in [6.6](#) for outgoing APDUs and [6.8](#) for incoming APDUs.

5.8 Service primitive conventions

This document uses service primitives to specify information flows between functional entities, i.e. information flows through service access points (SAPs). This document is organized such that each functional entity is specified in a separate clause:

- Security subsystem in [Clause 7](#),
- Adaptor layer in [Clause 8](#),
- Secure session services in [Clause 9](#).

NOTE ISO 24102-3 specifies service primitives and service primitive functions for various SAPs introduced in ISO 21217 at the level of ASN.1 type definitions. The service primitive functions identified in this document can be used to complement the specifications in ISO 24102-3.

Service primitives are not testable but indicate the information that one functional entity is obliged to make available to another in order for the second functional entity to carry out the indicated activities.

Following the conventions of the layered OSI model specified in ISO/IEC 7498-1, in this document, a message from a higher layer to a lower layer is specified with an:

— XXX.request service primitive of the service XXX,

and the response is specified with an:

— XXX.confirm service primitive.

Similarly, a message from a lower layer to a higher layer is specified with an:

— XXX.indication service primitive,

and the response from the layer entity to the lower layer is specified with an:

— XXX.response service primitive.

The information contained in a service primitive is referred to as "service primitive function"; see ISO 24102-3. The list of parameters of a service primitive function does not contain any information allowing to map .confirm service primitives on respective .request service primitive, and .indication service primitives on respective .response service primitive; such mapping parameters are specified in ISO 24102-3 for the service primitives. Thus, this document parameterizes only service primitive functions.

6 Process flows and sequence diagrams

6.1 General

[Clause 6](#) specifies process flows supported by the security services specified in this document and the sequence of data flows associated with each of these process flows.

[Subclause 6.2](#) provides an overview on process flows. [Subclause 6.3](#) presents state diagram conventions.

Owner authorization shall be as specified in [Annex D](#).

6.2 Overview of process flows

This document supports the following process flows; see [Annex A](#) for use cases that make use of these process flows.

- Configure, see [6.4](#): Prepare the functional elements to participate in secure communications.
- Start Session, see [6.5](#): Perform processing once a secure session has been established but before data is exchanged.
- Send data, see [6.6](#): Send data from the home application to the peer application.
- Send access control PDU, see [6.7](#): Exchange messages between the home security subsystem and the peer security subsystem with a goal of updating the state of one or both.
- Receive PDU, see [6.8](#): Receive an APDU, an access control PDU, or a proxied TLS handshake packet.
- Extend Session, see [6.9](#): Mechanism for one party to indicate to the other that it has a new certificate with which it wishes the session to be associated.

- Secure connection brokering, see [6.10](#): Proxy a TLS handshake between two other applications, one of which already has a secure connection to the home application and one of which already has a secure connection to the peer application.
- Force end session, see [6.11](#): Terminate a secure session between the home application and one peer application, with the termination initiated by the application or the security subsystem.
- Session terminated at session layer, see [6.12](#).
- Deactivate, see [6.13](#): Prevent the functional elements from starting any more sessions with the current set of parameters.

An example of how these process flows are combined to create a full secure session, start to end, is given in [6.14](#).

6.3 Sequence diagram conventions

The sequence diagrams follow the following graphical conventions. In this description the word “message” is used in the UML sequence diagram sense of a communication between two participants and is not intended to imply any over the air communications.

- In each sequence diagram, each message between participants is associated with a descriptive name and with the identifier of the service primitive that is used to implement that message. For example, in the clipping of [Figure 10](#), the message contains an APDU and is sent using the primitive `App-Sec-Data.request`, see [7.7.4](#).

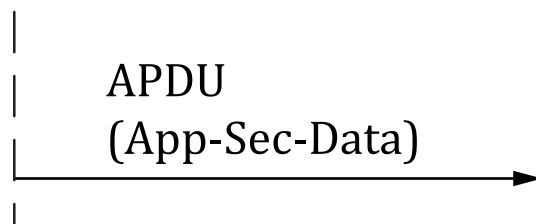


Figure 10 — Convention for sequence diagram - 1

- If the descriptive name is in italics, it is associated with a local message, i.e. a message that does not directly contain data that has passed or is intended to pass to the peer trusted entity. For example, in the clipping of [Figure 11](#), the message *configure (role, session type)* contains only local information and is sent using the `App-Sec-Configure.request` service primitive.

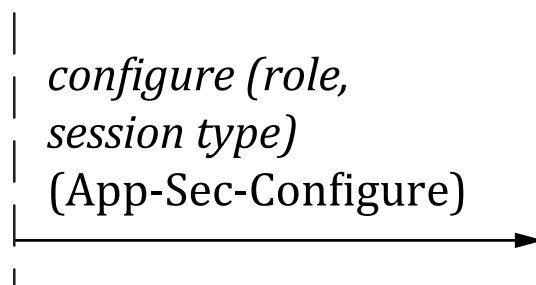


Figure 11 — Convention for sequence diagram - 2

- Mapping to primitives: See [5.8](#) on service primitive conventions. In the sequence diagrams presented in [Clause 6](#), the suffixes `.request`, `.confirm`, `.indication` and `.response` are omitted for compactness. In addition, also for compactness, the `.confirm` and `.response` service primitives are not shown in the diagrams of [Clause 6](#) unless they are significant information flows (in other words, amount to anything other than an ACK). In all cases, the description of the diagrams includes the service primitive suffixes and describes whether the `.confirm` and `.response` service primitives are used as anything other than an ACK.

6.4 Configure

In this process flow the application configures the security subsystem with configuration information necessary to run the secure communications. At the end of this process flow, if the security subsystem has been configured to take the client role in the secure session, the security subsystem initiates the handshake for the secure session. If the security subsystem has been configured to take the server role, it awaits incoming connections.

A prerequisite for this step is that the application has a socket identifier, i.e. an identifier for the communications session to which security is to be applied. This document does not specify socket functionality other than to note that all functional entities shall interpret a socket identifier consistently.

The process steps for this process flow are as follows.

- a) The application uses the App-Sec-Configure.request primitive (see [7.7.1](#)) to inform the security subsystem of:
 - 1) the role it has in the secure session (i.e. client or server);
 - 2) the socket to be used for communications with the peer trusted entity (this is pass-through information for the secure session service). This includes an indication of whether the other endpoint is inside or outside the secure boundary. This socket is not used by the security subsystem but is passed on to the secure session services via Sec-Sess-Configure.request;
 - 3) if the other endpoint is outside the secure boundary, i.e. if cryptographic security mechanisms are to be used to protect the session:
 - i) whether the socket shall be used for reliable or unreliable communications (i.e. TLS or DTLS),
 - ii) which cryptomaterial handle should be used for authentication of the secure session.
- b) The security subsystem:
 - 1) uses the access policy to determine what secure session types are permitted. In this edition of this document, three secure session types are permitted:
 - i) TLS for reliable transport mechanisms;
 - ii) DTLS for unreliable transport mechanisms;
 - iii) no specific security mechanism for communications sessions within a secure boundary of an ITS-SCU (which are secure by assumption);
 - 2) acknowledges the request from the application using App-Sec-Configure.confirm, see [7.7.2](#), (not shown in the diagram). This may include an indication that the secure session type requested (physical versus cryptographic) is not permitted for this application by the access control policy. If the secure session type is not permitted, the sequence terminates here. Otherwise, go to the next step;
 - 3) if a cryptographic secure session is requested, uses the Sec-Sess-Configure.request, see [9.3.1](#), to pass the following to the secure session service:
 - i) from the application:
 - a) role,
 - b) cryptomaterial handle,
 - c) socket type;
 - ii) from the access control policy:
 - a) the permissions constraints to apply to incoming IEEE 1609.2 certificates, i.e. the acceptable ITS-AID and SSP values,

- b) the SDEE Identifier used to identify the application to the IEEE 1609.2 secure data service, as specified in IEEE 1609.2,
- c) the timeout parameters for the session:
 - how long the session may be inactive before a fresh handshake is required,
 - how long the session may be active before a fresh handshake is required,
 - if the secure session services are playing a server role, how long the service shall be open to incoming requests;

If the parameter *role* is “Client”, the instance attempts to carry out the handshake.

- c) The secure session service uses Sec-Sess-Configure.confirm, see [9.3.2](#), to ACK the exchange.

A sequence diagram for this process flow is given in [Figure 12](#).

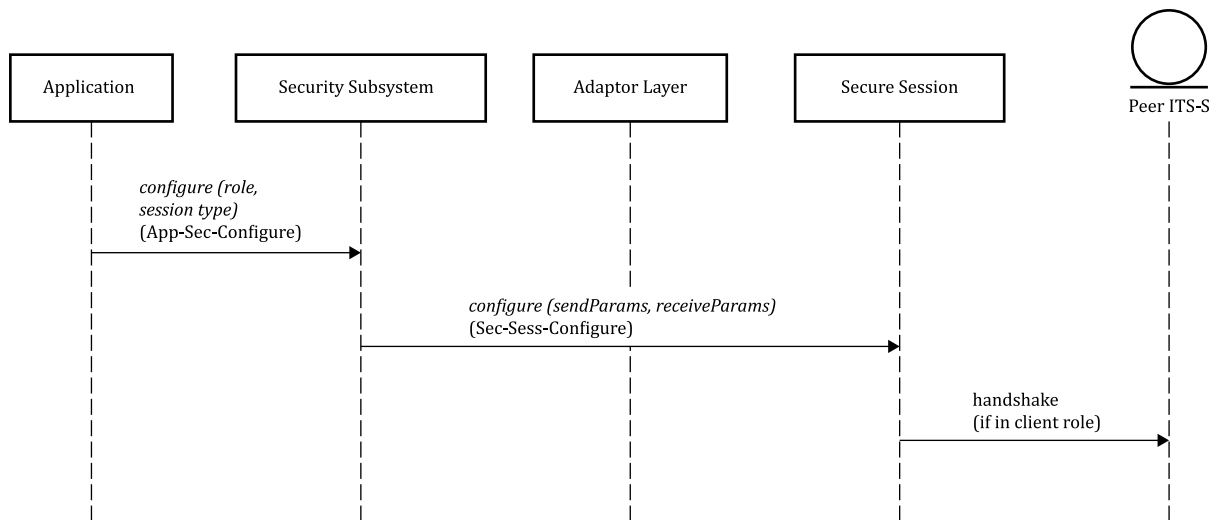


Figure 12 — Sequence diagram for Configure process flow

NOTE To change the key (the cryptomaterial handle) to be used for sessions, the application or security subsystem can Deactivate the current secure session instance as specified in [6.13](#) and Configure a new secure session instance using the mechanisms of this subclause.

6.5 Start session

In this process flow, following a handshake between the secure session service on the home ITS-SU and the secure session service on the peer trusted entity, the secure session service provides the security subsystem with the credentials received from the peer trusted entity and the security subsystem initiates any additional access control activities that the access policy specifies is required to complete before data can be exchanged.

In the case of a physical secure session, the session may start without credentials being exchanged. In this case the extended authentication mechanisms of this document may be used to send credentials during the session.

Prerequisite:

- The secure session is a secure session as specified in [5.3](#).
- The secure session service has been configured using the methods of [6.4](#).

- If the secure session is a cryptographic secure session, a handshake has taken place (initiated by the client in the secure session as the final step in "Configure").
- The certificate received in the handshake of the secure session (if applicable) matches the constraints provided to the secure session service during the Configure process flow.

The process steps for this process flow are as follows.

- a) The local secure session service uses the Sec-Sess-Start.indication service primitive to communicate with the security subsystem, see [9.3.3](#), providing:
 - 1) an indication that the secure session has started and passed the validity conditions that were specified to be checked by the secure session service;
 - 2) the certificate from the secure session service on the peer trusted entity (in the case of a cryptographic secure session);
 - 3) if the secure session services are acting in the server role, the Session ID to be used to identify the session; otherwise, the Session ID is provided in Sec-Sess-Configure.request, see [9.3.1](#).
- b) The security subsystem uses the access control policy to determine the status of the incoming connection. (Note that the security subsystem will only be notified of incoming connection if the certificate in the handshake has met the conditions provided in the Configure step, and so the security subsystem has assurance that the certificate meets those conditions; as such, it is permissible for there to be no additional access control processing at this step). The status may be:
 - 1) Success;
 - 2) Unauthorized, communication rejected. In this case the security subsystem ends the secure session as specified in [6.12](#);
 - 3) Additional access control actions are necessary. The additional access control actions specified in this document are:
 - i) Request extended authentication, as specified in [7.4](#).
 - ii) Request enhanced authentication, as specified in [7.3](#).
- c) If additional access control actions are to be carried out or if the policy indicates that success or failure are to be communicated to the peer:
 - 1) the security subsystem generates an access control PDU and passes it to the adaptor layer via the Sec-AL-AccessControl.request service primitive (see [8.4.1](#)):
 - i) in the case where success is being communicated to the peer, this is an access control PDU of type AccessControlResult.success, as specified in [7.6.5](#),
 - ii) in the case where failure is being communicated to the peer, this is an access control PDU of type AccessControlResult.failure, as specified in [7.6.5](#),
 - iii) in the case of enhanced or extended authentication, the contents are as specified in [7.3](#) and [7.4](#), respectively;
 - 2) the adaptor layer acknowledges the access control PDU via the Sec-AL-AccessControl.confirm primitive, see [8.4.2](#);
 - 3) the adaptor layer wraps the access control PDU in an adaptor layer PDU of the appropriate subtype as specified in [8.2](#);
 - 4) the adaptor layer passes the ALPDU to the secure session service via the AL-Sess-Data.request primitive, see [9.4.1](#);

- 5) the secure session service acknowledges the ALPDU via the AL-Sess-Data.confirm service primitive, see [9.4.2](#), secures the ALPDU, and passes this as a datagram to the network stack for transmission.
- d) If the status is success and the secure session services are acting in the server role, the security subsystem uses the App-Sec-StartSession.indication service primitive (see [7.7.3](#)) to provide the session ID obtained in step a.3) above to the application. (If the secure session services are acting in the client role, the successful initiation is indicated by the return code from Configure).

Additional access control requests may be sent within the secure session, see [6.7](#) (sending) and [6.8](#) (receiving).

A sequence diagram for this process flow is given in [Figure 13](#).

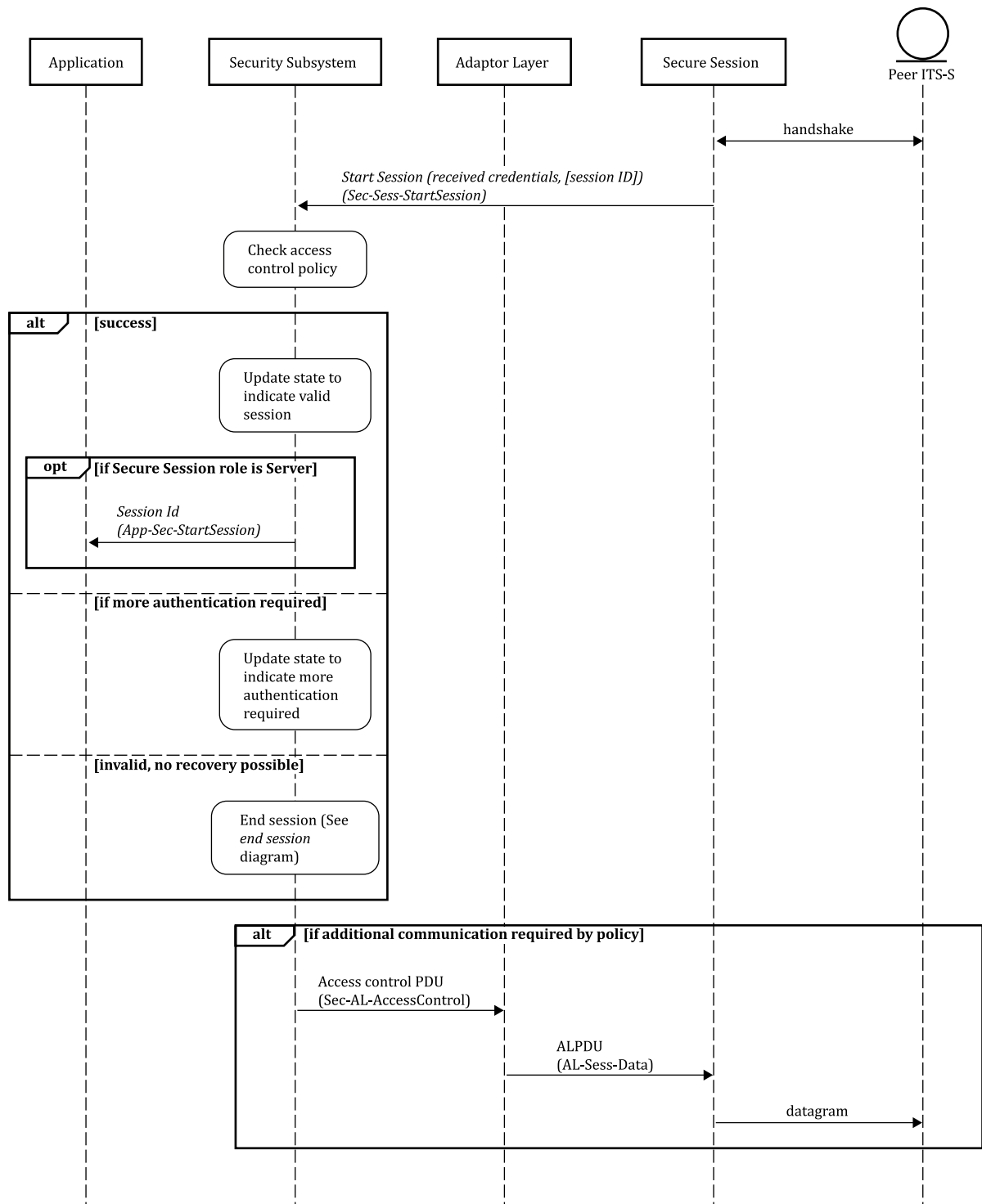


Figure 13 — Sequence diagram for Start Session

6.6 Send data

In this process flow the application on the home ITS-SU sends a datagram over an established ISO 21177 secure session. This subclause describes sending activities. The receive-side sequence is specified in [6.8](#).

Prerequisites:

- The secure session service has been configured using the methods of [6.4](#).

- If the secure session is a cryptographic secure session as specified in [5.3](#), there has been an initialize step using the methods of [6.5](#).

There are two alternate paths for this process flow. In one path, the application has the option to apply application-level non-repudiation to individual APDUs. In the other, the application never applies application-level non-repudiation to individual APDUs. This document does not provide a mechanism to negotiate which of these paths is followed; this choice is assumed to be a part of the application specification. If data is a stream rather than individual PDUs, the application determines how to split the stream into PDUs for submission to the secure session in the same way as is done for TLS.

The process steps for this process flow are as follows.

- a) The APDU is encapsulated as an `Ieee1609Dot2Data` in a way that depends on whether the application has the option to apply application-level non-repudiation to individual APDUs:
 - 1) if the current APDU shall apply non-repudiation, the application uses `App-Sec-Data.request`, see [7.7.4](#), to send the APDU to the security subsystem for signing. The security subsystem returns the signed APDU, which is an `Ieee1609Dot2Data` of type signed as specified in IEEE 1609.2 (or, if appropriate, an error message) via `App-Sec-Data.confirm`, see [7.7.5](#);
 - 2) otherwise, the application creates an `Ieee1609Dot2Data` as specified in IEEE 1609.2, of type unsecured, containing the APDU.
- b) The application uses the `App-AL-Data.request` primitive, see [8.3.1](#), to provide the original APDU or the APDU output from step a), as appropriate, to the adaptor layer.
- c) The adaptor layer:
 - 1) ACKs the data using `App-Sec-Data.confirm`, see [7.7.5](#);
 - 2) adds the session non-repudiation service if so configured. This service is not specified in this edition of this document but is planned for future revisions;
 - 3) adds the Data header as specified in [8.2](#), to indicate that the ALPDU is an application datagram;
 - 4) the adaptor layer provides the data to the secure session service via the `AL-Sess-Data.request` primitive, see [9.4.1](#).
- d) The secure session service:
 - 1) ACKs the data using `AL-Sess-Data.confirm`, see [9.4.2](#);
 - 2) fragments (if necessary) and cryptographically protects the data and passes it to the network stack for transmission to the peer trusted entity.

A sequence diagram for this process flow is given in [Figure 14](#).

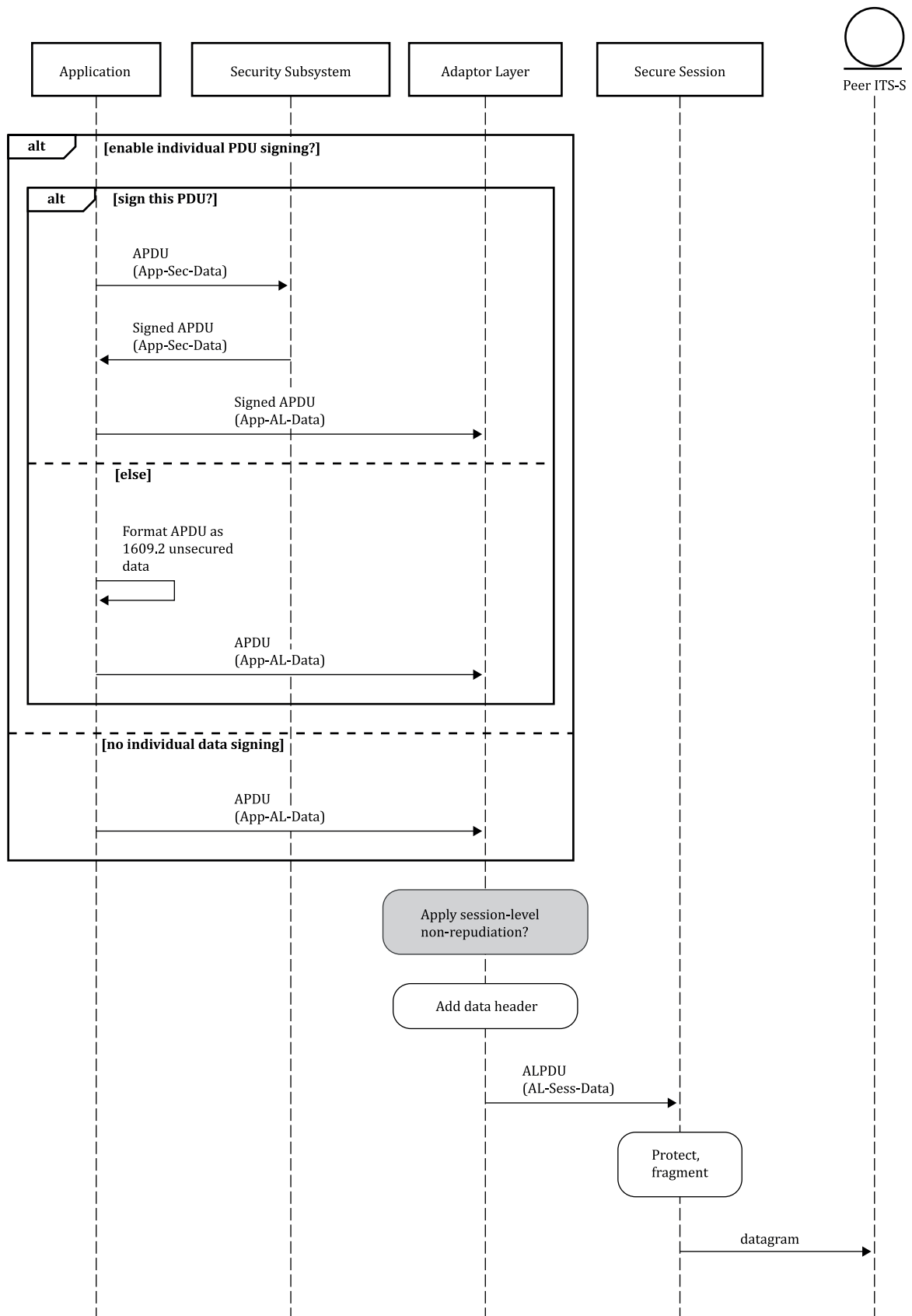


Figure 14 — Sequence diagram for sending data

6.7 Send access control PDU

In this process flow the security subsystem creates an access control PDU to update the authorization state on the peer security subsystem, i.e. to update the peer security subsystem's knowledge of the permissions and other authenticated properties of the home application. The access control PDU can be created as result of a local access policy decision, or as result of an application command (through an application/security subsystem interaction to be specified in the specification of an application that uses the mechanisms provided by this document), or as the result of an incoming data or access control PDU. This subclause describes sending activities. The corresponding receive-side sequence is specified in [6.8](#).

Prerequisites:

- The secure session service has been configured using the methods of [6.4](#).
- If the secure session is a cryptographic secure session as specified in [5.3](#), the session has been started using the methods of [6.5](#).

The process steps for this process flow are as follows.

- a) The security subsystem generates an access control PDU and passes it to the adaptor layer using the Sec-AL-AccessControl.request primitive, see [8.4.1](#).

NOTE In principle, the access control PDU can be individually signed or not. In this edition of this document all the access control PDUs are not signed and are authenticated to the peer trusted entity by virtue of being sent over the secure session.

- b) The adaptor layer:
 - 1) ACKs the access control PDU using Sec-AL-AccessControl.confirm,
 - 2) creates an ALPDU, which is an Iso21177AdaptorLayerPDUA with the component `messageId` equal to `accessControlId` and the component `value` equal to the received access control PDU, as specified in [8.2](#);
 - 3) provides the ALPDU to the secure session service via the AL-Sess-Data.request primitive, see [9.4.1](#).
- c) The secure session service:
 - 1) ACKs the ALPDU using AL-Sess-Data.confirm, see [9.4.2](#);
 - 2) fragments (if necessary) and cryptographically protects the data and passes it to the network stack for transmission to the peer trusted entity.

A sequence diagram for this process flow is given in [Figure 15](#).