TECHNICAL REPORT

ISO/TR 21186-3

First edition
2021-02

Cooperative intelligent transport systems (C-ITS) — Guidelines on the usage of standards —

Part 3: Security

Systèmes de transport intelligents coopératifs (C-ITS) - Lignes directrices pour l'utilisation des normes —

Partie 3: Sécurité

Reference number
ISO/TR 21186-3:2021(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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 documents 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

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).

A list of all parts in the ISO 21186 series can be found on the ISO website.

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 provides informative material of interest to implementers deploying secure systems to carry out ITS applications. ITS stations are rapidly maturing with regards to specification, use and security conformance standards. In support of the ITS station ecosystem new standards have been developed, such as ISO/TS 21177, which provide a framework for device-to-device secure sessions and resource access authorization. Common criteria protection profiles have been developed and adopted for use in distinctive European ITS service domains, such as automotive V2X safety services, as well as a narrow set of infrastructure messaging based services.

NOTE ITS services are provided by means of ITS applications.

Given the diversity of anticipated ITS services and potential data sensitivities, this document was constructed to provide ITS stakeholders with a holistic analysis and indication of possible extensions to the ITS station security ecosystem.

This document includes the following sections:

1) An overview of security considerations for application specification and deployment in ITS. This overview also provides a detailed rationale for the following sections.

2) A use-case driven threat model based roughly on common criteria processes in establishment of threats, security objectives and SFR relative to three genericized ITS station data sensitivity and access control scenarios. Each scenario can be used by security practitioners as a starting point to baseline ITS station platform protection profiles of varying application types and data sensitivities. The genericized protection profile security requirements are then compared to several existing (or under development) protection profiles established for automotive use cases to determine possible gaps in security controls that should be addressed when tailoring subsequent security targets or related protection profiles.

3) An implementation example of the development of an access control policy implementation for an ISO/TS 21177 conformant ITS station unit. The example access control policy is application-specific and depends on many factors, including the type of ITS station unit on which the access control policy is used. Consequently, this access control policy implementation example is not suitable for being copy-pasted to the context of other ITS applications. Rather, the process described in this example can be considered as a suitable template for a process aimed at creating an access control policy for any ITS application running in an ISO/TS 21177 conformant unit.

4) Inputs for the development of a CP governing the issuance of certificates for ITS station units. A CP is necessary for the deployment of a system to ensure consistent behaviour of different CAs (or, more generally, credential issuance actors) within the system. This consistent behaviour enables receiving devices to trust all received messages to the appropriate level, knowing that those devices have been through the same certificate-issuing process no matter where the certificates were obtained. In early 2019, the European Commission published a CP for use for "Day 1" ITS applications, to be enforced by a top-level root of trust implemented in an entity called the TLM. This document concludes with a set of high-level gaps and potential mitigations for ITS PKI participants and implementers.

5) A description of additional functionality that extends the functionality of ISO/TS 21177. This material is written in a manner which will enable it to be inserted into a future revision of ISO/TS 21177.

These five areas of content significantly ease the process of deploying new ITS applications securely.

This document is forms part of the ISO 21186 series on "Guidelines on the usage of standards," which is comprised of the following Parts:

1) Standardization landscape and releases;

2) Hybrid communications;

3) Security (this document).

Cooperative intelligent transport systems (C-ITS) — Guidelines on the usage of standards —

Part 3: **Security**

1 Scope

This document provides guidelines on security applicable in Intelligent Transport Systems (ITS) related to communications and data access.

In particular, this document provides analyses and best practice content for secure ITS connectivity using ISO/TS 21177.

This document analyses and identifies issues related to application security, access control, device security and PKI for a secure ITS ecosystem.

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.

ISO/IEC 27000, *Information technology — Security techniques — Information security management — Overview and vocabulary*

ISO/IEC 27032, *Information technology — Security techniques — Guidelines for cybersecurity*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 27000 and ISO/IEC 27032 and the following apply.

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

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
attack vector
extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behaviour (member functions or methods) in object-oriented programming

4 Symbols and abbreviated terms

AA	authorization authority
ACL	access control list
APDU	application protocol data unit
API	application programming interface
CA	certificate authority
CAM	cooperative awareness message
CP	certificate policy
CPS	certification practice statement
C-ITS	cooperative intelligent transportation systems
COER	canonical octet encoding rules
CPOC	certification point of contact
CRL	certificate revocation list
CTL	certificate trust list
DEK	data encryption key
DoS	denial-of-service
EA	enrolment authority
ECDSA	elliptic curve digital signature algorithm
ECIES	elliptic curve integrated encryption scheme
ECTL	European certificate trust list
ECU	electronic control unit
HSM	hardware security module
IDX	ITS data exchange
IVN	in-vehicle network
ITS	intelligent transport systems
ITS-AID	ITS application object identifier
ITS-S	ITS station
ITS-SU	ITS station unit
IVIM	infrastructure to vehicle information message
KEK	key encryption key
MAPEM	MAP extended massage

ND	nomadic device
NIST	National Institute for Standards and Technology
OCSP	online certificate status protocol
OEM	original equipment manufacturer
PAKE	password authenticated key exchanges
PDU	protocol data unit
PII	personally identifiable information
PKI	public key infrastructure
PP	protection profile
RSU	roadside unit
SCMS	security credentials management system
SCN	sensor and control network
SDEE	secure data exchange entity
SFR	security functional requirements
SPaT	signal phase and timing
SPaTEM	SPaT extended message
SPDU	secured protocol data unit
SPII	sensitive or personally identifiable information
SREM	signal request extended message
SSEM	signal request status extended message
SSP	service specific permission
TLM	trust list manager
TOE	target of evaluation
TSF	TOE security functions
TVRA	threat, vulnerability and risk analysis
UGP	unified gateway protocol
V-ITS	vehicle intelligent transport systems
VMS	variable message sign

5 Security in C-ITS

5.1 General

This subclause provides an overview of security in C-ITS and a rationale for the material in the rest of the document.

Systems have functional goals, and also have security goals which support these functional goals. The details of security goals depend on context, but high-level security goals are always the same:

- Provide assurance that parties within the system receive the right information necessary for acheiving their functional goals.
- Provide assurance that parties who are not authorized to receive information do not receive that information.

Systems use security controls to achieve their security goals. A security control is a specific mechanism implemented as part of a strategy to achieve the security goal. (For ease of discussion, this document also uses the concept of a security service. A security service is an identifier of the kind of action which needs to be performed in order to achieve a security goal, while a control is concrete and implementable). There are many different kinds of security controls, including the following:

- Communications security controls, which provide assurance that communications between two trusted parties meet the security goals of the system, i.e. that if two parties are legitimate, then there can be a data exchange between them in which each party is assured that the data came from the other party, is of known quality, and is not revealed in the course of the communications to unapproved parties.
- Platform security controls, which provide assurance that a device that is trustworthy at one point can remain trustworthy.
- Data processing security controls, which provide assurance that data is appropriately handled before or after it is communicated.
- Access control security controls, which provide assurance that activities within the system are carried out only by parties that have authorization to carry them out.
- Organizational and process security controls, which provide assurance that the other security controls in the system are implemented properly.

5.2 Security design process for C-ITS applications

A number of security design process approaches have been proposed for ITS applications. ETSI has specified a TVRA process[23] and applied it to the ETSI Day 1 ITS services[24]. The output of this TVRA process is a recommendation for specific security mechanisms. An alternative approach is outlined in ISO/IEC 15408-1, ISO/IEC 15408-2 and ISO/IEC 27001, which form the basis for the common criteria approach to security certification. A third approach is given in Federal Information Processing Standards (FIPS) 199[31], published by the NIST in the USA. Finally, SAE J2945/5[26] specifies an approach to deriving SSPs, a mechanism used to enable fine-grained access control statements to be made with IEEE 1609.2 certificates. As part of this process, it outlines an overall approach to deriving security requirements for a connected vehicle application.

All of these approaches use a systems engineering approach with three stages of the design: use case and concept of operations, requirements, and detailed design. Each stage can be considered more detailed than the previous one.

All of these approaches have a similar overall structure:

- Firstly, the ITS application is detailed to a level where information flows are specified allowing the ITS application to achieve its functional goals.

- Then, a security analysis is performed to identify the security requirements on the information flows and on the parties and to derive from the requirements on the flows the corresponding requirements on the parties that interact with each other in the ITS application.

- The security analysis can reveal that the application design needs to be changed, either to directly address identified security issues, or because the security analysis has uncovered additional use cases or features of the application which need to be incorporated into the main design.

- The analysis/design update process iterates until the design is stable at the current level of detail. At that point, the design can be moved forwards to the next, more detailed, level of detail and the security analysis is performed and iterated on that next level of detail until the third and final level of detail is reached.

- The output is a full specification of the application, including the security controls.

Security controls to be specified include communications security controls, implementation security controls, organizational security controls, policy security controls, and others. Details of how controls are to be derived are given in the referenced methodologies ([23],[10],[31],[26]).

Clause 5 focuses on the communications security controls and supporting security controls necessary for enabling communications security:

- An overview of communications security mechanisms in the C-ITS context is provided in 5.3.

- An overview of the role of CAs and certification processes is provided in 5.5.

- A rationale for the additional detailed technical material included in this document is provided in C.1.

Although interface standards typically focus on communications security controls, all types of controls are important and a full specification on how to securely deploy a system includes a full specification of all of the relevant security controls.

5.3 Communications security mechanisms in C-ITS

The communications security services and controls that are appropriate for a distributed ITS application depend on the communications topology. At a high level, there are two types of communication strategies: broadcast and non-broadcast. From a security perspective, "non-broadcast" includes both unicast and groupcast: the important thing from a security perspective is that in both the unicast and groupcast cases, some potential receivers are being excluded from receiving information (and so confidentiality mechanisms, and key management to enable those confidentiality mechanisms, are necessary).

Figure 1 illustrates typical communications security mechanisms in a non-broadcast setting. In this setting, one actor (the host or responder) has certain resources which the other actor (the accessor or initiator) wishes to access in order to carry out an operation. Typical operations include reading the resource value (potentially with associated metadata), writing to the resource location, or causing the execution of some operation on the resource. In this setting:

- The host uses an access control policy to determine which operations can be carried on each resource by different types of accessor.

- The accessor uses the security service authorization to access to demonstrate that it has rights to the particular access that it is requesting. See 5.4 for a discussion of access control types.

- The following security services are applied to each individual APDU sent as part of the exchange:
  - source authentication, to provide assurance that the message is sent by a valid participant in the exchange;
  - confidentiality to ensure that the contents cannot be read by an unauthorized actor;

- protection against modification to ensure that the contents are not modified in transit (or, more specifically, to ensure that if the contents are modified in transit, that this can be detected).

Protection against modification can include the application of multiple services, such as anti-replay (protecting a receiver against acting on the same APDU twice, as if it was two different APDUs) and freshness checking (protecting a receiver against acting on an APDU which is too old to be relevant). In the security service categorization developed by the NIST (USA), all of the services designed to ensure that the receiver of the APDU has the correct understanding of the PDU's properties (time of generation, generating party, data integrity, etc.) are considered part of one high-level service called integrity. This is the convention followed in this document.

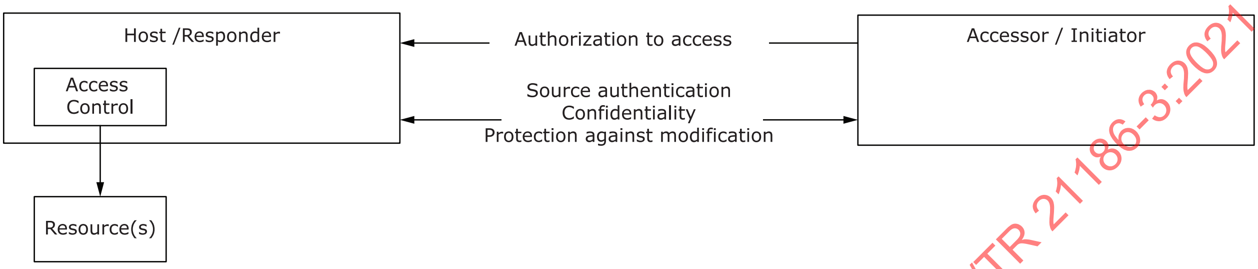

Figure 1 — Security services for non-broadcast communications topologies

Figure 2 illustrates typical communications security mechanisms in a broadcast setting. In this setting, the actor (the broadcaster) has locally available resources on the basis of which it broadcasts a PDU which receivers can opportunistically make use of. In this setting, the following security services are applied to each individual APDU sent as part of the exchange:

- source authentication to provide assurance that the message is sent by a valid participant in the exchange; and
- protection against modification to ensure that the contents are not modified in transit (or, more specifically, to ensure that if the contents are modified in transit, this can be detected).

  Protection against modification can also include services such as anti-replay and freshness checking, as in the non-broadcast case.
- The sent PDU can also make use of pseudonymity, which is a security service that enables receivers of the message to understand the instantaneous state of the broadcaster, but provides protection against the receivers being able to track the state of the broadcaster over time. (For example, if the broadcaster is a vehicle sending CAMs, pseudonymity inhibits receivers from being able to use the CAMs to determine the entire route that the vehicle took). Pseudonymity is not achieved by a single mechanism, but is an outcome of multiple mechanisms acting in concert.

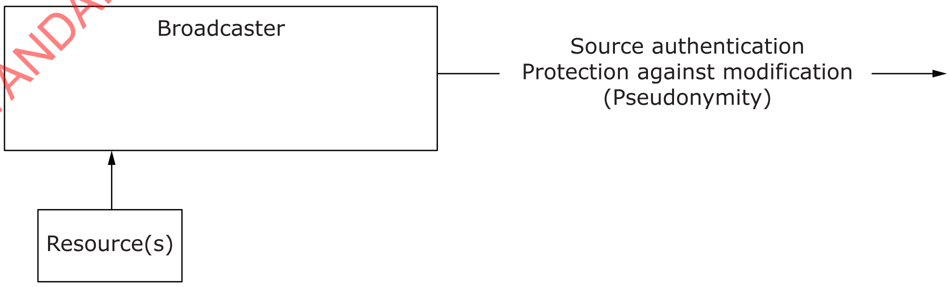

Figure 2 — Security services for broadcast communications topologies

5.4 Source authentication and access control mechanisms

This section discusses appropriate mechanisms for source authentication within ITS applications, starting from a discussion of access control types.

In this document, communications security is primarily discussed in terms of access control. Access control mechanisms are mechanisms that manage access to resources. Access control policies are a configuration of access control mechanisms that allow the integrity and confidentiality goals of the system to be achieved.

A key mechanism for enabling access control is source authentication. This mechanism provides assurance to the receiver of an incoming message (which can be data, a command, or both) that the sender has a particular property. There are two main types of source authentication.

— **Identity authentication**: In identity authentication, the property transmitted is a unique identifier of the sender. Identity authentication enables identity-based access control. In this case, the receiver maintains an ACL which maps from the sender's identity to the actions that the sender is permitted to take. Identity-based access control is common for managing access to centralized systems, as these systems can manage individual permissions for each accessing party. Identity-based access control allows the certification of accessing parties to be tied to a single, long-lived, property of the device, i.e. the identity. Permissions can then be mapped to the identity and managed by a single centralized process using this identity as a look-up key. This enables permissions to be changed dynamically (e.g. if an employee changes job role, they can be granted access to different corporate information without needing to change how they authenticate to the system). However, identity-based authentication creates a requirement for parties granting access to maintain some information file (a username/password file, an ACL, or some equivalent) with one entry for each accessing party and to update this if new parties are granted access.

— **Role-based authentication**: In role-based authentication, the property transmitted contains explicit semantics and is not just an identifier. For example: the sender is entitled to act as a police vehicle; the sender is entitled to send CAMs; the sender is entitled to request tolling information; the sender is entitled to create a software image that is appropriate for installation on a particular device. In role-based authentication, the property transmitted can be about a physical property of the sender, or about a role in which the device is entitled to act, or about any other property that will enable the receiver to make an access control decision. In some contexts, a distinction is made between role-based and attribute-based access control, where attribute-based access control uses more fine-grained properties of the sender than role-based access control. This document does not make use of that distinction. Additionally, attribute-based access control can include properties of the environment or the resource itself in making access control decisions. In role-based access control, the receiver maintains an access control policy mapping properties or combinations of properties to the activities that a party with those properties is entitled to carry out, similarly to the use of the ACL in identity-based access control. The distinction is that in role-based access control, a receiver does not need to maintain separate access control permissions for each individual sender, just for each individual role. This means that the receiver does not need to manage information for each party in the system, allowing for more robust access control when edge devices interact without going through a centralized system. The trade-off is that if a device's role changes, the device needs to be issued with new credentials indicating that new role. Role-based authentication has an advantage over identity-based authentication in those cases where the cost of updating each sender is less than the cost of updating each of the likely receivers.

In either case, the receiving device will apply an access control policy to determine whether the sending device is entitled to take the particular action requested. Examples of access control policy include the following.

NOTE 1 The following examples are linked to use cases (i.e. applications) for illustrative purposes, not to suggest that the use case uses the exact access control policy specified below.

NOTE 2 The access control policy examples below are for action taken on the receiving side. The assumption is that the sender is aware of the access control policy likely to be implemented on the receiving side and will create outgoing messages and metadata that provide the information needed for the message to be accepted under the receiver's access control policy.

- **Cooperative awareness**: For a broadcast message, if the message is validly signed by an IEEE 1609.2 certificate with the Cooperative Awareness Basic Service ITS-AID and meets other conditions for freshness and relevance, accept. Otherwise, reject. See [19].
- **Attaching an RSU to a signal controller for SPaT message generation**: If the RSU is a valid RSU for generating SPaT messages, and is owned by the road authority that owns the signal controller, and is being installed by a licensed installer, accept. Otherwise, reject.

Regarding the choice between identity-based and role-based authentication based on communications topology, note that for many application data exchanges between two edge devices there are two natural topologies to consider: a centralized topology, where communications are routed through one or more cloud services, and a local topology, where the edge devices communicate directly. (See Figure 3).

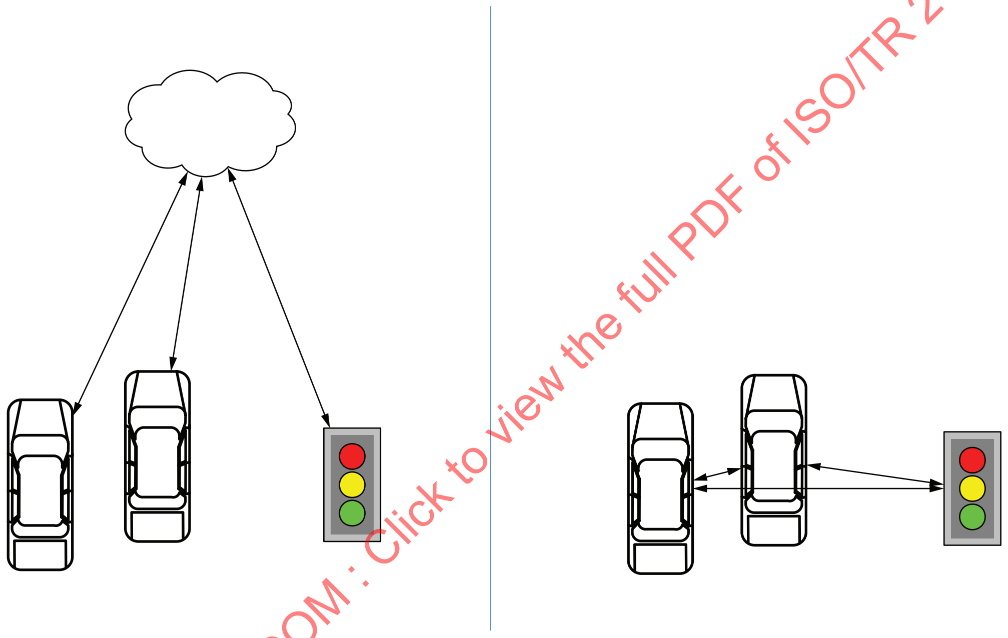

Figure 3 — Centralized topology (left); decentralized topology (right)

For the centralized topology it will often make sense to use identity-based access control on each of the communications links. A natural example of this topology is in Figure 4, where each device has its own "home server" and the two home servers communicate to carry out the technical goals of the ITS application. In this setting, identity-based access control is used as follows:

- The initiating device has a persistent connection i.1 to its home server. The connection is associated with the device's identity. The home server manages the initiating device's attributes and authorizations, including for example its balance in any financial accounts.
- The responding device also has a persistent connection i.3 with its own home server.
- The two home servers have a persistent connection with each other. Each home server manages a list of the actions that the other home server is entitled to authorize. When one device attempts to carry out an action through the cloud connection, its home server determines whether it is authorized and communicates the authorization to the other home server.

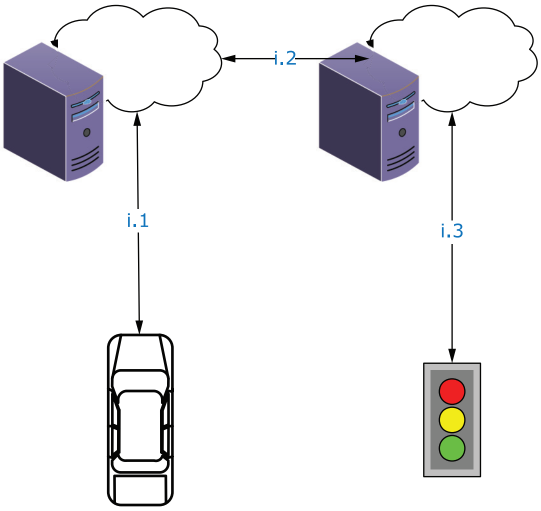

Figure 4 — Devices communicating via their home servers

Typically these connections will be authorized by standard web technologies such as cookie based authentication, token based authentication, OAuth over TLS[12], and others. The initial authentication of the field device to its home server can be via username and password, or X.509 certificates, or a PIN transmitted out of band, or some other means. The authentication mechanism also manages continuity of authentication through the lifetime of the connection.

For the direct or decentralized topology, a natural approach is to use IEEE 1609.2 certificates for authorization, either to sign the messages directly in the case of broadcast or to authorize a persistent secure session using ISO/TS 21177 in the case of multicast/unicast. The ISO/TS 21177 approach supports the case where one device needs to provide evidence of multiple properties to the other device, by allowing the device to provide multiple certificates within a given flow. This allows properties with different lifecycles to be managed separately: for example, the sensors that are built in to a device are a long-lived property, but the fact that it is operated by a registered operator might be short-lived. In the ISO/TS 21177 model, these two properties can be attested separately but associated with the same counterparty, allowing the application of sophisticated access control policies. This is illustrated in Figure 5, which shows an example of the access control policy that can be in place for a police car requesting signal pre-emption.

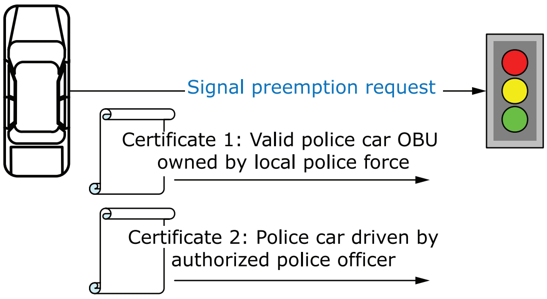
Signal preemption request
Certificate 1: Valid police car OBU owned by local police force
Certificate 2: Police car driven by authorized police officer

Figure 5 — Example of a police car requesting signal preemption under an access control policy that requires multiple certificates

A natural way to architect applications is so that they can be executed in two ways: via a cloud service and via a local service. The cloud service approach allows for processing associated with the application activities to be carried out in the cloud, where scalability and new functionality is easier to implement. The local approach allows activities to be carried out with less latency and also to be robust against loss of connectivity to the cloud service for any reason. If an application is architected in this way, it can be appropriate for the application to be provided with two different sets of security credentials:

one set of identity-based credentials for use with the cloud service approach, and one set of role-based credentials for use with the local service approach.

For some applications a natural architecture is to have no direct communications between distinct edge devices. An example of this would be an application where a central service directly carries out the service of interest to the edge device with no direct involvement of other edge devices. Another example would be an application where the central service aggregates and analyses data from edge devices before providing post-processed, statistical data to its client edge devices. In these cases, an authorization approach based solely on identity-based authorization can be appropriate.

For other applications a natural architecture can be to only have direct communications between edge devices. An example of this would be applications with extremely time-critical communications latency requirements, or applications where the volume of data is so large that it would cause a significant network burden to send it "up" to a server and then back "down" to a neighbouring edge device, or applications where it is critical that the application can operate in the absence of connectivity and so it does not make sense to deploy a version that depends on network connectivity. In this case, an authorization approach based solely on role-based authorization (using IEEE 1609.2 certificates and, if appropriate, ISO/TS 21177 secure sessions) can be appropriate.

5.5 Certificate authorities and certification processes

Almost all secured digital communications are secured using cryptography. The general model is as follows:

- A device obtains a credential, which is a binding between a secret value and a property of the device (or of an actor using the device, or of an application processes running on the device). More specifically, it is a binding between a means of demonstrating that the device knows the secret value and the property of the device.

  The means of demonstrating that the device knows the secret value can take different forms:

  - If the secret value is a shared secret, used in a centralized system, the means of demonstrating that the device knows the secret can simply be the secret itself or some value derived from it.
  - If the secret value is a private key for an asymmetric cryptosystem, the means of demonstrating that that the device knows the secret is to perform an operation involving the private key and a related operation with the public key – either using the private key to sign a message and the public key to verify it, or using the public key to encrypt random data and using the private key to decrypt it.

  The binding between the knowledge of the secret value and the property can also take different forms:

  - The data used to prove knowledge can be stored in a database with the property. This is common in centralized/identity-based systems as discussed above.
  - The property can be distributed with the data used to prove knowledge, for example in an X.509 or IEEE 1609.2 certificate containing the public key (the data used to prove knowledge) a name or a set of permissions. In this case, a third party is trusted to ascertain that the actor using the device does in fact have the property being claimed.

- When an actor using the device engages in secure communications, the device demonstrates that it knows the secret value. The receiver trusts the binding between the data used to prove knowledge of the secret and the property. In other words, it trusts that if the device knows the secret, then it has the claimed property. Cryptographically secure operations can then be carried out, leveraged by the secret value, as is done in (for example) Transport Layer Security (TLS)[13], ISO/TS 21177, or IEEE 1609.2.

The purpose of this discussion is to illustrate that the communications security in any system relies fundamentally on two things:

1) The secret known to the device is protected from being extracted from the device, as any actor who knows that secret can claim to be the device or have the device's permissions.

2) The binding process establishes without dispute that the device or the actor using the device does in fact have the properties claimed.

Examples of "the properties claimed" in point 2) are: that the device's name is what it says it is, or that the device is operated by licensed operators (for example, the police), or that the device has the sensors and data quality checks necessary to enable it to usefully send CAMs.

To provide assurance that 1) is true, an ITS application specification includes (inline or by reference) a specification of the physical, operating system, and software security requirements on the device. This specification can take many forms, but a standard type of this specification is a PP as defined in ISO/IEC 15408-1 and ISO/IEC 15408-2. A PP allows certification laboratories to determine that a particular type of device meets the security properties necessary for it to operate. An application deployment plan is not complete unless there is a PP or similar document. PPs exist for different types of devices in the context of C-ITS[10],[11],[12]. Clause 6 of this document contains input material for a PP for IDX devices, i.e. devices that use ISO/TS 21177 to access and make use of data on other ITS devices.

To provide assurance that 2) is true, in the specific case of certificates, a certificate issuer (certificate authority, or CA) can be required to follow a CP and CPS. A CP and CPS already exist for the ETSI Day ITS applications. Clause 8 of this document provides input material for a CP for CAs that issue certificates to IDX devices.

5.6 Introduction to the rest of this document

As discussed in 5.5, a full specification of an aplpication includes a CP and a PP (either directly in the specification or by reference to external documents). As of December 2019, there is only a limited number of CPs and PPs suitable for use with ITS-SUs. Specifically, there is a European Certificate and Security Policy[27],[28] which addresses the "ETSI Day 1 applications", but there is no policy for other types of application, and there is no published policy for any type of application in other regions.

The goal of this document is to facilitate deployment of an additional class of applications in addition to the ETSI Day 1 applications. This class is IDX applications. An IDX application is any ITS application that uses unicast connectivity to exchange data (including commands/requests) directly with a peer. IDX applications are defined in more detail in Clause 6.

To facilitate deployment of these IDX applications, this document provides the following:

— Security analysis and controls for three types of IDX device in Clause 6. Additional analysis of the security functional requirements for each type of device is provided in Annex B.

— Worked example of the development of an access control implementation for an IDX device in Clause 7.

— Identification of threats to the PKI from a stakeholder perspective, along with proposed mitigations. Additionally, an analysis of changes necessary to the EU CP to enable the deployment and certification of IDX devices is provided in Clause 8.

This document also provides the following new material related to ISO/TS 21177. This material is motivated by feedback from industry stakeholders on additional functionalities that would enhance the abilities of the ISO/TS 21177 approach:

— Ability to provide up-to-date information about the revocation status of CAs, specified in Annex C.

— Support for the concept of an "owner" of an IDX device. The owner has the ability to explicitly permit, prohibit, or end rights to access that devices. This is specified in Annex D.

— Additional proposed modifications and enhancements to ISO/TS 21177, specified in Annex E.

6 Security analysis and controls for an IDX device

6.1 Background

An IDX application is any application that uses unicast connectivity to exchange data (including commands/requests) directly with a peer. Examples include:

- RSU to signal controller connections to enable SPaT messages to be generated;
- vehicle to road weather control station connections to enable weather condition reporting per SAE J2735/3[25];
- vehicle diagnostic interactions when carried out over a local interface rather than through a cloud service. Implicit in the definition of the IDX application is the concept that ISO/TS 21177 is an appropriate communications security mechanism.

NOTE Individual application specifications can use a different communications security mechanism if this is appropriate for that application.

As discussed in 5.2, a full deployment-oriented application specification includes platform security requirements for devices running instances of that application in specific roles. A standard approach for specifying these requirements is to develop a common criteria PP for that application. Common criteria PPs have been developed and adopted for use in individual European ITS service domains, including automotive V2X safety and a narrow set of infrastructure messaging applications. They are widely accepted as an appropriate way to specify testable security requirements for different types of devices or modules.

Clause 6 provides material that can be used as input to the specification of PPs for the three types of IDX device. The material includes SFRs as well as the analysis leading to the derivation of those SFRs.

Clause 6 also presents a gap analysis, comparing the derived security requirements for the three device types are compared to several, existing (or under development) PPs for automotive use cases. This assists device suppliers who wish to supply devices that can run both IDX and other C-ITS applications, by indicating a set of security features to support IDX applications and enabling a gap analysis of these features against the features of C-ITS devices.

6.2 IDX device concept

6.2.1 General

This subclause defines the basic concept of operations for three types of IDX device analysed using an abbreviated common criteria-like requirements analysis process.

The aim of this section is to provide clear architectural definitions and functional activities for an IDX device that interfaces with an ITS-SU fronting a SCN in order to access or upload data resources of varying sensitivity. The functional activities provide the context for defining assets and more complex use cases making use of those assets. The three generic IDX device/application types are listed below. The assumption of this description is that every IDX application is fundamentally one of these types. Each application can have unique features of its own that cause it to have unique security requirements and, as such, each application's security features should be independently analysed and derived. However, by providing this analysis of the three fundamental types of application, this document aims to kickstart the process of providing a security analysis, security requirements, PP and CP for each and every new application. The three types are as follows:

- IDX devices running public data retrieval applications, where the accessing device is requesting data that would not be subsequently linked to the device providing the data (for example, a road weather management system requesting road weather data from a vehicle on the road, as per SAE J2945/3).

- IDX devices running private data exchange applications, where the accessing device is requesting data that might be subsequently linked to the device providing the data, for example malfunction reports from a traffic signal controller, or path information from a pedestrian ITS-SU.
- IDX devices running active access applications, where the accessing device is requesting to write to the host device or execute operations on the home device. An example of this is a management device wirelessly accessing a VMS to change the message.

Additional analysis of the security functional requirements for each type of device is provided in Annex B.

For each device type, this document provides a security analysis in the form of a) threats, b) security objectives and c) SFR. These form the basis of IDX device PP which can be useful for:

1) analysing existing European ITS-S-related PPs to determine if they are suitable for applying to IDX devices; and

2) establishing new PPs and security targets intended for varying ITS data sensitivity levels. The goal is that this material can be used to develop a consistent set of PPs for a wide range of device and data sensitivities in the ITS domain.

Security analysis uses the concept of a TOE, which is the system that is to undergo security analysis and which is contained within a security boundary, such that all interactions across the boundary are known and observable. A high-level flow of the PP development process is shown in Figure 6.

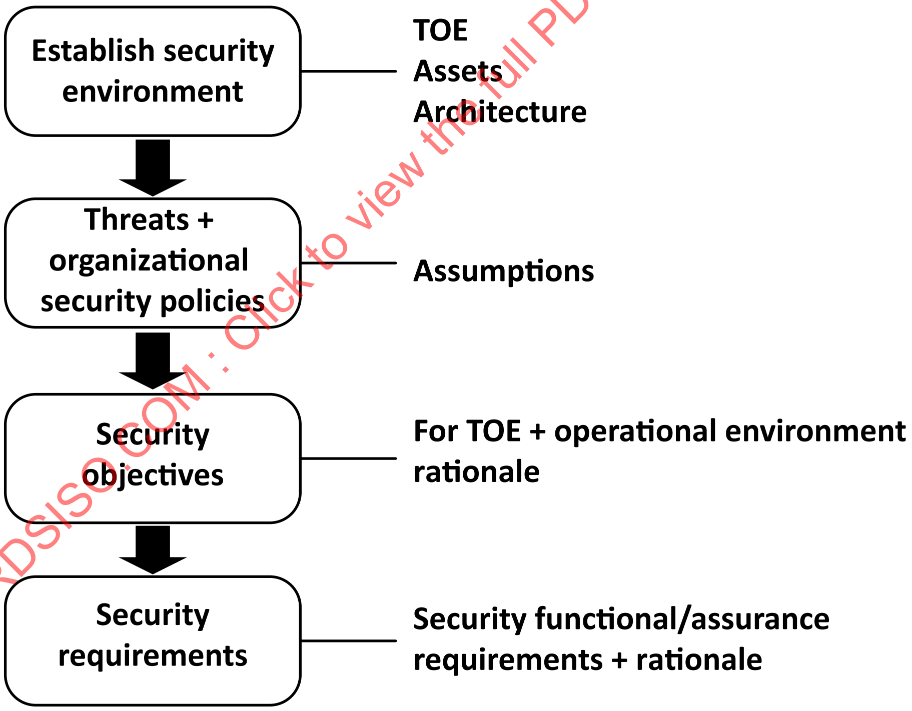
Establish security environment
TOE
Assets
Architecture
Threats + organizational security policies
Assumptions
Security objectives
For TOE + operational environment rationale
Security requirements
Security functional/assurance requirements + rationale

Figure 6 — High-level depiction of ISO 15408 series PP development process

The TOE in this case is the IDX device that is assumed to interact with another ITS-S in order to obtain access to that ITS-S's resources. The terms TOE and IDX device are used interchangeably throughout this material.

Readers of this clause are assumed to be proficient in the language and processes of the common criteria.

Contrast with full common criteria approach: The security requirements in this clause are not identified as being derived from a full common criteria evaluation, as a full common criteria evaluation is dependent on the specifics of the application definition. A full common criteria evaluation requires detailed knowledge of the application interfaces, physical connections and intended usage environment in order to arrive at a detailed threat model, set of inherited organizational policy requirements, SFR and customized operations for those requirements. This document does not provide specific customizations of SFR for specific applications. Developers of specific applications are expected to carry out their own analysis of security assumptions and requirements, using this section as an example.

Additionally, the focus of common criteria is on developing security requirements for the whole system, including organizational security requirements. The goal of this document is to enable development of a certification programme for devices, and so very few organizational security requirements are assumed or provided.

6.2.2 System architecture and device

6.2.2.1 Target of evaluation

The TOE can be any type of IDX device, for example a unit that is used in automotive maintenance facilities to connect to and electronically interact with a SCN for diagnostics, maintenance, emissions testing or other purposes.

The IDX device high-level view is indicated in Figure 7. The IDX device is assumed to conform to the architectures for an ITS-S standardized in ISO 21217. A certificate management functional block is explicitly shown in the block diagram due to the high significance and dependence that other management blocks and applications have on it.

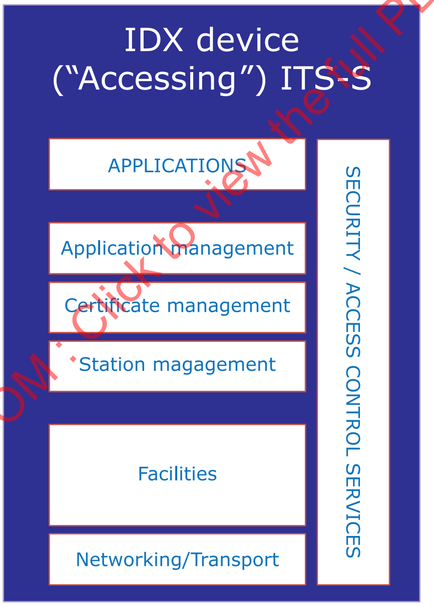

Figure 7 — Target of Evaluation (IDX device that accesses an SCN ITS-S application)

6.2.2.2 General system view

The IDX device is assumed to be used within an untrusted security environment. It can be manually or remotely commanded by a user and will connect to an ITS application (e.g. in a vehicle) using either a wired or wireless protocol stack.

NOTE ISO/TS 21185 specifies a methodology to identify globally uniquely communication protocol stacks and communication profiles and provides a list of respective globally unique identifiers.

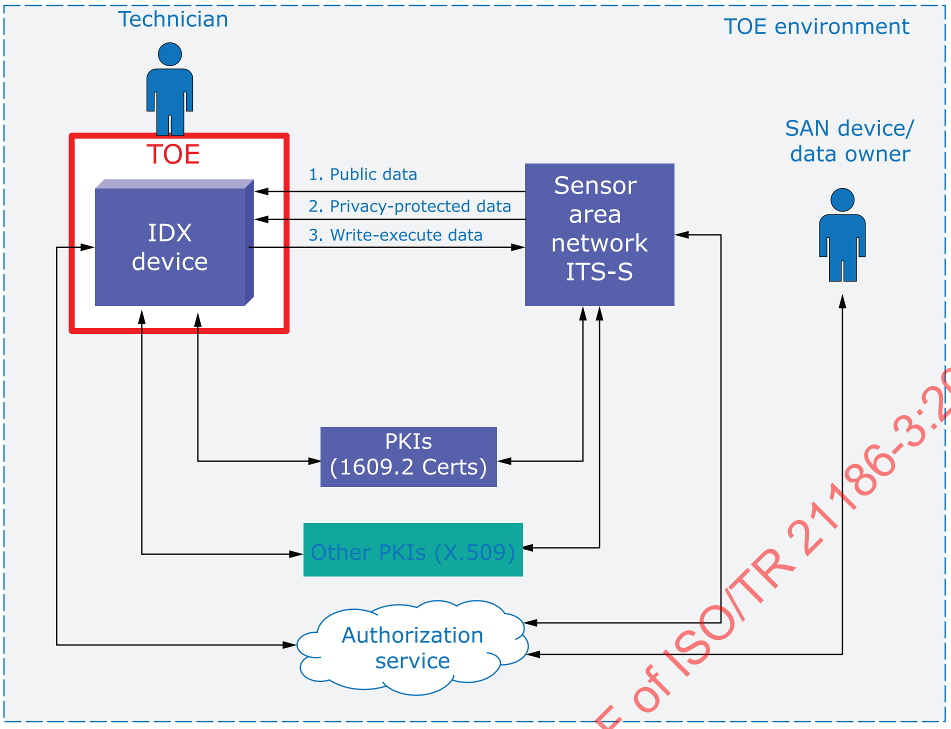

Figure 8 — System View: TOE and TOE Environment

Three resource sensitivity types:

1) public,

2) privacy-relevant, and

3) write-execute data,

each corresponding to the IDX device classes described in 6.2.2, are shown to be accessed in Figure 7.

These resource types vary in security/privacy sensitivity and are used to differentiate the three access scenarios defined later in this clause. These scenario-driven accesses are used to differentiate the level of threats to an IDX device ITS-S based on the resource sensitivity of the data it accesses via the SCN ITS-SU.

6.2.2.3 Connections and trusted relationships

This subclause defines connectivity between the TOE and other system actors. Connections to and from the TOE are characterized as the following:

— **User to IDX device:** This connection can be manual (local) or remote. In most cases, the user is assumed to be local to the IDX device (used to access the resources of the SCN) and able to leverage its User Interface to request and perform various resource accesses via the SCN ITS-S to which it is connected. If the user is remote to the IDX device, this CONOP assumes that a secure connection, for example TLS, exists between the user and the IDX device.

— **IDX Device to SCN**: This connection uses a communications protocol stack (see ISO/TS 21185) conformant with ISO/TS 21177. ISO/TS 21177 will provide secure connectivity using TLS 1.3 over the underlying stack. ISO/TS 21177 is a secure connection and authorization framework; its capabilities can be leveraged for secure access between the IDX device and SCN ITS-SU. This document makes no assumptions related to the duration of the connection between an IDX device and a SCN. The connection can be persistent or ad-hoc.

— **IDX device to PKI:** An IDX device is assumed to possess IEEE 1609.2 certificates for securely accessing SCN resources. Therefore, IDX devices require certificate provisioning services of a 1609.2

PKI. At minimum, this requires bootstrapping an IDX device with PKI trust anchors and providing it with an enrolment certificate it can use to request application-specific certificates.

- **IDX device to evice Vevndor:** The device is expected to communicate directly (i.e. secure sessions) or indirectly (i.e. end-to-end file security) with its vendor in order to receive updated software, firmware or configurations.
- **IDX device to other PKIs**: All other secure connections are assumed to leverage conventional, internet-based PKIs based on X.509. The provisioning steps of these certificates can vary between manufacturers and PKI providers.
- **Connections to third-party authorization service**: Facilitating secure access between an IDX device and a SCN ITS-SU are connections each has with an external authorization service. In cases when sensitive accesses require consent of the data owner (e.g. entity owning the sensor network data), the SCN ITS-SU can either directly/locally request owner consent, or the SCN ITS-SU and IDX device can interact via a 3rd party authorization service to obtain owner consent (e.g. via a concurrence message or PIN to enter via mobile app).

6.2.3 Threat modelling data scenarios and examples

6.2.3.1 Scenario 1— non-sensitive public data

Scenario 1 consists of an IDX device that is only used to access non-confidential 'public' data such as diagnostics data not intended for any regulatory use. The information is assumed to be publicly viewable and does not require integrity or source authentication protections. While this scenario is unlikely, except for the most consumer-oriented devices, it is useful for differentiating the level of protections needed with rising, scenario-based sensitivity levels.

It is assumed that a Scenario 1 IDX device enrols with a PKI in order to obtain certificates for interface connectivity to a peer device (e.g. SCN ITS-S).

The Scenario 1 data access and flow area indicated in Figure 9.

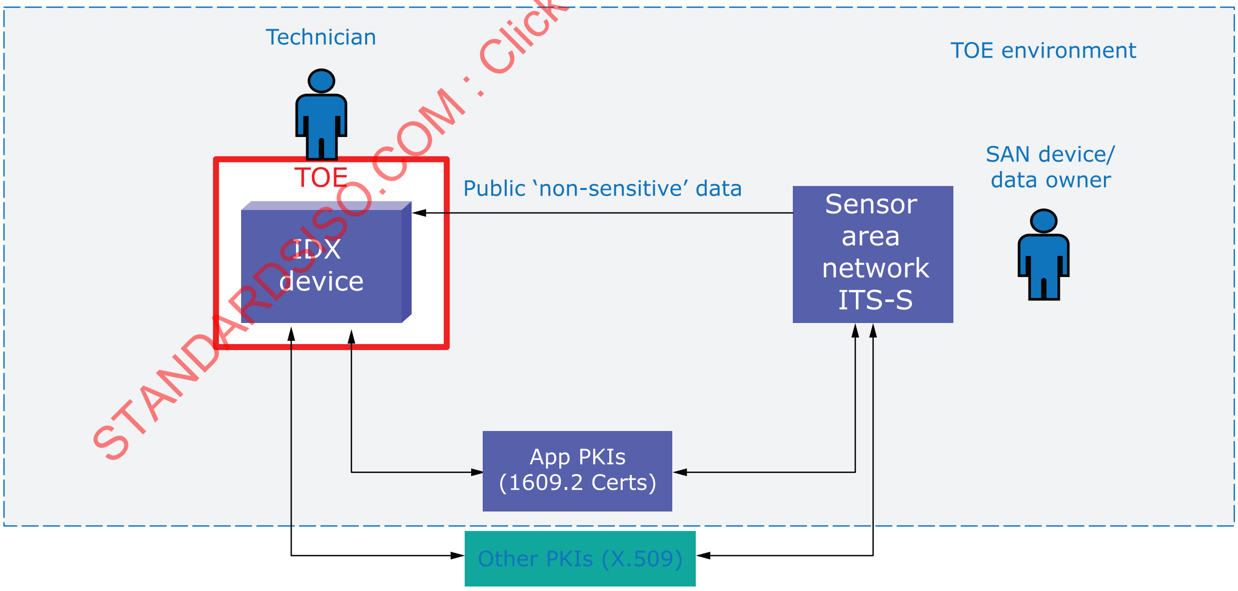

Figure 9 — Scenario 1: Access of public, diagnostics data

6.2.3.2 Scenario 2 — privacy-protected data assets

Scenario 2 is distinguished from Scenario 1 in that the IDX device accesses:

- privacy-sensitive data, i.e. data that reveals identifying or trackable data (e.g., location history) associated with the SCN or its owner, or
- public data that requires integrity protection, as can be the case with data used for regulatory adherence.

The principle data threats in this scenario include unauthorized information disclosure and a violation of data source or integrity. For privacy-sensitive data, it is assumed that the data is not only access-controlled, but that the data owner has a right to authorize the disclosure through a local or external authorization service. Scenario 2 is depicted in Figure 10.

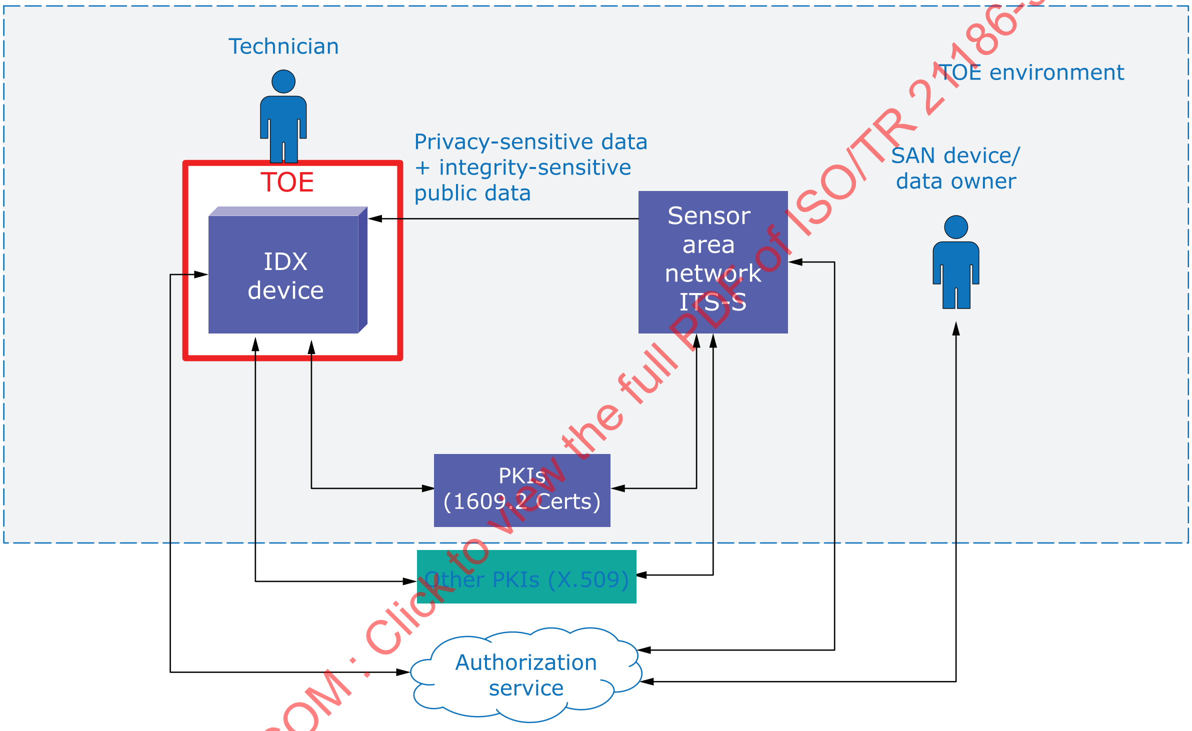

Figure 10 — Scenario 2: Access of privacy-sensitive data

6.2.3.3 Scenario 3 — executable or writeable data assets

Scenario 3 introduces the ability of an IDX device to perform actions that modify the state of the SCN, i.e. perform write actions such as execution of sensitive commands/files or loading of new firmware or configurations to the SCN or its fronting ITS-SU. This scenario is illustrated by two primary examples of IDX device functionality:

1) The IDX device supports the trusted load of new software images for the SCN.

2) The IDX device supports the trusted modification of existing SCN access control policies.

Scenario 3 is depicted in Figure 11.

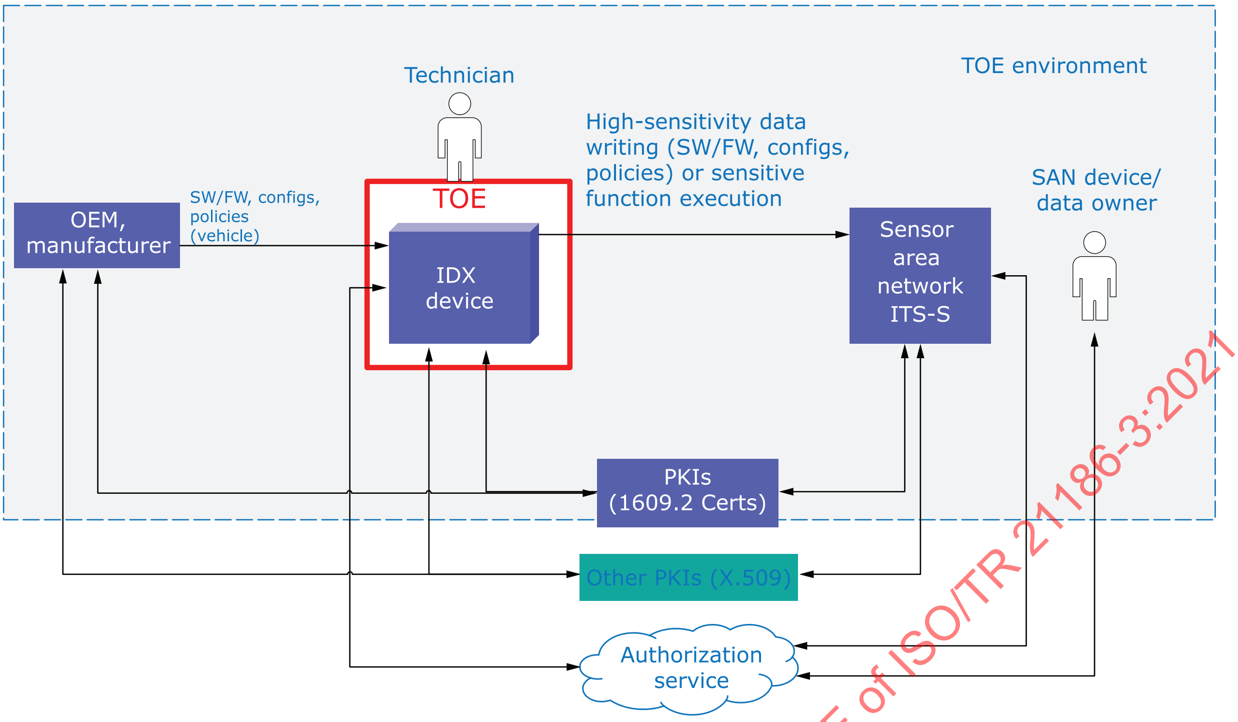
Technician
TOE environment
High-sensitivity data writing (SW/FW, configs, policies) or sensitive function execution
SAN device/ data owner
OEM, manufacturer
SW/FW, configs, policies (vehicle)
TOE
IDX device
Sensor area network ITS-S
PKIs (1609.2 Certs)
Other PKIs (X.509)
Authorization service

Figure 11 — Scenario 3: Access of sensitive write, execute or load functions

Examples of Scenario 3 include:

Example 1: Load software image: The desired result of this action is the successful update of one or more software images to a device within the SCN. The IDX device is loaded with a software image received from an OEM, Tier 1 supplier or other organization. Delegation of authority to sign software image updates can also occur. Metadata associated with the software image is assumed to be included in the load package. This metadata can include cryptographic signatures, timestamps, and other data. Supporting industry trends in firmware update frameworks, it is assumed that metadata can be signed by different actors than those who sign the software image.

Access control functions of the device are configured to evaluate the permissions of the signers to distribute or package both software image files and metadata files. This can be accomplished using the ITS-AID/SSP parameters provided in SCMS certificates.

Example 2: Modify (write) access control policy: The desired result of this action is a change to the SCN ITS-S ISO/TS 21177 access control policy. This example focuses on an access control modification that allows any software updates signed by a SCN device supplier to be installed without requiring future user-explicit authorization grants. In this example, ISO/TS 21177 enhanced authorization is used to receive authorization from the user to make the permanent change to the SCN access control policy.

Threat modelling assumptions pertinent to Scenario 3 include:

1) The trust relationship between the logged-in user and the IDX device is based on organizational policy and not in scope for any PP requirements derived from this example. It is assumed that the organization responsible for provisioning user accounts to the IDX device has procedures in place to properly vet users and map users to roles.

2) There is a secure access cloud service available to support extended authorization requests that solicit the SCN data owner for approval.

3) The software image is digitally signed by an authorized organization.

4) Metadata associated with the SCN software image is digitally signed by an authorized organization.

NOTE Software and firmware signing by manufacturers need not use the IEEE 1609.2 certificate on which ISO/TS 21177 resource access control decisions are made. Manufacturers can use X.509 certificates emanating from private or public PKIs.

6.2.4 Assumed device functions and activities

The activities pertinent to the three IDX device scenarios are provided in Table 1. Each activity represents an assumed capability of a generic IDX device. The table is broken into the following sections:

- General activities pertain to all Scenarios and can be used to construct more elaborate use cases. These activities are identified with an ID starting with "G" and are common to all IDX devices.
- Scenario-specific activities pertain to use cases involving access of each Scenario's SCN data and their unique sensitivities. IDs starting with S1, S2 or S3, respectively, indicate that the capability only applies to an IDX device in the indicated scenario.

For the purposes of scenario-centric threat modelling and traceability, it is assumed that each Scenario-capable device is exclusive to a given scenario. For example, a Scenario 3 access device is not able to perform Scenario 2 privacy data accesses. In practice, if a Scenario 3 device also performs Scenario 2 data accesses, security practitioners can baseline a PP by merging and reconciling the Scenario 2 and 3 threats and associated security controls.

The activities depicted in this section are intended provide an understanding of the capabilities, assumed limitations and other miscellaneous assumptions concerning the IDX device, such that security practitioners can understand and adjust the scenario threats and baseline security controls when building their own PPs.

Table 1 — General and scenario-specific operational activities

ID	Activity	Description
General activities (all Scenarios)		
G01	Bootstrap/enrol device into application PKI	The IDX device is bootstrapped and enroled into the PKI that generates such 1609.2 certificates. This activity includes type certification, provisioning with initial key material or random number generation seeds, and (if applicable) registration of the device platform with an appropriate registry.
G02	Download and install IDX device firmware	An IDX device manager downloads/retrieves and installs new IDX device platform firmware.
G03	Download and install IDX device applications	A device manager downloads/retrieves and installs new IDX device applications. These applications are installed on the IDX device for accessing SCN resources; they are not installed on the SCN.
G04	Obtain initial authorization certificates	The IDX device is provisioned with authorization certificate(s) in order to connect with SCN application(s). Scenario 1 authorization certificates do not require sensitive access. It is assumed that they only indicate that a valid IDX device is attached since they are only intended to read public data. Scenarios 2 and 3 authorization certificates are assumed to indicate authorized roles associated with special accesses.
G05	Refresh authorization certificates	The service updates its authorization certificate(s) in order to continue to be able to connect to peer ITS-SUs fronting an SCN.
G06	IDX device is configured with new user and mapped to role	The IDX device ITS-S should not allow all device functions to be available to all users. This can be restricted using roles mapped to permitted activities. Roles are indicated by certificate ITS-AIDs and SSPs associated with those ITS-AIDs. This activity results in an application access policy whereby some user identities are permitted to access certain applications and others are not.

Table 1 *(continued)*

ID	Activity	Description
G07	Securely connect to a peer ITS-SU	The user, using the IDX device, connects to SCN application via ISO/TS 21177 ITS-Station secure session. The protocol stack is conformant with ISO 21185.
G08	Enrol into external authorization service	The IDX device can optionally participate in an external authorization service to enable cloud-supported authorization transactions between the accessing entity and the SCN owner/operator.
Scenario 1 activities		
S1-01	Request/retrieve public diagnostics data from SCN	Pre-requisites: G01-G07 The IDX device connection to the SCN is complete after the SCN has determined that a valid IDX device has connected. The IDX device application performs read requests on public data elements (e.g. Diagnostic Test Codes) via the peer SCN-fronting ITS-SU.
S1-02	SCN ITS-S performs policy check to determine if non-repudiation is required based on request	Pre-requisites: S1-01 The peer SCN-fronting ITS-SU checks policy based on the resource requested by the IDX ITS-SU and makes a non-repudiation decision. If non-repudiation is needed, the SCN peer ITS-SU digitally signs the requested data using the correct 1609.2 certificate. If non-repudiation is not needed, the peer ITS-SU returns the response over the ISO/TS 21177 secure session without application layer non-repudiation.
S1-03	SCN device provides report for IDX device (user) consumption only	Pre-requisites: S1-01, S1-02 The application provides the requested resources in a report, e.g. an emissions/diagnostics test report. In this activity, the read service invoked via the SCN-fronting ITS-SU assumes the requested data is for local user consumption only, i.e. does not contain data needed in a mandated or formalized regulatory report. The data is read and displayed to the user using the IDX device. Data is not further disseminated.
S1-04	SCN reports data for remote consumption by regulatory authority	Pre-requisites: S1-01, S1-02 The IDX device in this activity requests data via the SCN-fronting ITS-SU. The peer ITS-SU generates report data for an external regulator. In this activity, the IDX device is an intermediary facilitating the transaction between the SCN and the regulator, for example, passing a road weather data report from the SCN to to a weather management centre. It is assumed the regulator requires reasonable assurance the data came from a valid SCN or SCN-fronting ITS-SU.
S1-05	IDX device reports public data for remote consumption by regulatory authority	Pre-requisites: S1-01, S1-02 The IDX device in this activity requests and receives diagnostics data from the SCN-fronting ITS-SU. The IDX device performs additional analysis over the data/report and generates its own report for external regulator consumption. It is assumed the regulator requires non-repudiation and high assurance the data came from the indicated user, shop or IDX device (not the SCN).

Table 1 *(continued)*

ID	Activity	Description
Scenario 2 activities		
S2-01	IDX device ITS-SU requests from SCN ITS-SU a resource listing	Pre-requisites: G01-G07 The IDX device ITS-S should be capable of asserting its privileges to request details related to what privacy-protected data it is allowed to access. This activity consists of the IDX device requesting a resource listing after completing its TLS handshake with the peer ITS-SU. The resource listing should list only the privacy-protected resources that the IDX device, based on its presented 1609.2 certificate, should be allowed to know about. Note that an available resource listing can include resources that the user's certificate role is allowed to discover but not access. Some SCN resources, after discovered, can require additional authentication/authorization checks prior to accessing. The existence of other resources can be highly sensitive and not disclosed in the resource listing unless the user initiated the secure session with the appropriate role.
S2-02	Request privacy-sensitive data from SCN	Pre-requisites: G01-G07 and optionally G08 and S2-01 The IDX device connection to the SCN-fronting peer ITS-SU is complete after the peer has determined that a valid IDX device has connected. The user initiates via the IDX device a read-request on privacy-sensitive data elements or files. The IDX device checks if the user is authorized to perform this request. If yes, the IDX device sends the request to the SCN peer ITS-SU. If not, the IDX device indicates to the user that authorization is denied.
S2-03	SCN processes service-device request for privacy-sensitive data	Pre-requisites: G01-G08, S2-02 The peer SCN ITS-SU receives the request for privacy-sensitive data and using ISO/TS 21177 checks the application's security policy to determine if the secure session's client certificate is authorized to access the data. The response is either 1) yes, 2) no, 3) enhanced authentication required, or 4) extended authorization required via third-party authorization service. If yes: Provide data directly as a response to the request. If no: Return authorization error. If enhanced authentication is required: Utilize ISO/TS 21177 primitive to indicate to the IDX device that enhanced authentication is required. If extended authorization required: Utilize ISO/TS 21177 primitive to indicate to the IDX device extended authorization is required. The two ITS-SUs make use of the third-party authorization service as indicated by the peer ITS-SU to the IDX device displaying to maintenance worker the message "waiting for owner approval...".)

Table 1 *(continued)*

ID	Activity	Description
S2-04	SCN uses third-party authorization service to obtain data owner consent for local data release	Pre-requisites: S2-03 If third-party authorization is required, the SCN initiates the third-party authorization service provider with a request for owner consent. It summarizes or references the type of data being requested by the IDX device. By policy, it can also request from the SCN owner a) a one-time access, b) a time block over which the access can be performed or c) some other conditions on the access. The third-party authorization service sends a secure request to the SCN owner (e.g. smart phone). The owner responds with a grant or denial of the request. The owner response is indicated back to the SCN (or IDX device, depending on implementation) with the result of the requested access. If granted, the SCN application via the peer ITS-SU provides the privacy protected data to the IDX device.
S2-05	SCN uses third-party authorization service to obtain SCN owner/operator consent for remote data release	Pre-requisites: S2-02 This activity is identical to S2-03 with the exception that the disclosure, i.e. read access, of the privacy-protected data will only be to a known, remote entity, not the IDX device. It is expected in this case that the SCN-fronting ITS-SU provides the protected data to the IDX device in encrypted form so that the IDX device cannot read it. The IDX device is then expected to relay this data to a remote consumer that is able to decrypt and read the data.
Scenario 3 activities		
S3-01	IDX device ITS-SU stages a SCN software image load via SCN ITS-SU	The IDX device downloads critical software/firmware or configurations for the SCN. (Note: The IDX device ITS-S will perform a pre-validation including ISO21177 access control check on certificate ITS-AID prior to allowing a software image to be loaded to a SCN).
S3-02	IDX device performs write-execute functions in the SCN	The IDX device performs the load of a software image file to a SCN and receives a confirmation status in return.
S3-03	IDX device manages SCN access policy (application-specific)	The IDX device initiates a request to manage/update the SCN ITS-SU access control policy. This results in the download of the existing policy and request to make an update to that policy.

6.3 Device assets

This subclause details the assumed assets related the IDX device, the device environment, including assets of the connected SCN application, and the scenarios to which the assets pertain. Threats can impact any one or more of the listed assets.

The scenario-specific asset listing is intended to provide the security practitioner the assumed assets of the IDX device that can be threatened in the operational environment.

Asset	Description	S1	S2	S3
IDX device firmware (platform)	Firmware for the IDX device.	x	x	x
IDX device application software	Software applications on the IDX device. IDX device applications access resources from SCN applications via the SCN ITS-SU.	x	x	x
IDX device user account data	Names, identifiers, logins, role assignments, account terms and other account data that controls which users can access the IDX device and what local applications they can accessing via the SCN ITS-SU.	x	x	x

Asset	Description	S1	S2	S3
IDX device access control policies	Policies that control access to IDX device applications and services based on the accessor's identity. These policies can also map certificates to specific users. The existence of some policies and related services can be sensitive, therefore there can be a confidentiality sensitivity of this asset.	x	x	x
User passwords and other authenticators	Authenticators that authorized users use to prove their identities.	x	x	x
IDX device platform configuration data	Configurable parameters that can change the behaviour of the IDX device. This can be in the form of flat files, MIBs, etc. The existence of some policies and related services can be sensitive, therefore there can be a confidentiality sensitivity for this asset.	x	x	x
IDX device firewall settings	A type of configuration data that controls external access to the device.	x	x	x
SCN ITS-SU access control policy(s)	Access control policy resident on the SCN ITS-SU that controls access to SCN application and data resources. This policy can be used by the SCN ITS-SU to make grant/deny requests to requested resources over ISO/TS 21177.	x	x	x
IDX device application and data access control policy(s)	Access control policy resident on the IDX device that controls access to IDX device's client applications and data resources. This policy can be used by the IDX device to make grant/deny requests when the user using the IDX device requests certain operations or data accesses.		x	x
Certificate provisioning reachability settings	Network/application location information used by the IDX device certificate provisioning logic to obtain/refresh certificates and trust credentials.	x	x	x
IDX device platform logs	Logs that are accessible by a device administrator or possibly vendor.			
SCN ITS-SU access control policies	These access control policies are resident on the SCN ITS-SU. They can control the following: 1) which certificates/SSPs can invoke which services or access certain data; and 2) which certificates/SSPs can access which vehicular applications.	x	x	x
SCN firmware, software and applications	SCN executables that originate from the SCN or aftermarket vendor and can be invoked or loaded by an IDX device.			x
SCN public diagnostics data	Low-sensitivity application data originating from the SCN. Typically, these are diagnostics test codes, emissions or other diagnostics-related data that are not privacy-sensitive. Integrity and source authentication potentially needed in some applications.	x		
SCN public diagnostics ata Rerports	Assume that public diagnostics data are components of a report, in addition to other metadata. These assets can be meant for consumption by the diagnostics user or by some type of regulator or other organization. Reports can be directly generated by the SCN or can be generated by the IDX device. Integrity and source authentication potentially needed in some applications. It is assumed that any metadata in the report is also not privacy-sensitive.	x		

Asset	Description	S1	S2	S3
SCN PII / tracking data / proprietary data	Data originating from the SCN that is privacy-sensitive or required to be privacy-protected. This can include metadata about the SCN owner, data that links the SCN to the owner, tracking-related data, PII/SPII or OEM/vendor-proprietary data. This information has an expectation of confidentiality and should require either a policy or technical control to restrict who has access to it. Regulatory test reports originating from the SCN can fall into this category if they contain data or metadata that is privacy-sensitive. Potentially sensitive to confidentiality, integrity and authenticity lapses.		x	
Writable or executable data	This data asset can include: — Commands or API calls from the IDX device to the SCN that enable writing of sensitive data. — SCN ECU firmware or applications. — Sensitive configuration files/data originating from the IDX device or vendor. This data can include confidentiality sensitivity if it contains vendor-proprietary information.			x
Cloud services trust anchors	Roots of trust for device, application or cloud service vendors that enable the IDX device to trust service endpoints.	x	x	x
Cloud services TLS public keys (incl. certificates)	These are public keys/certificates associated with the IDX device and also associated with the cloud service (e.g. third-party authorization service).	x	x	x
Cloud services TLS private keys	Private keys belonging to the IDX device that can be used to authenticate itself to cloud/vendor services.	x	x	x
IDX device application certificates	1609.2 certificates used by the IDX device to authenticate via ISO/TS 21177 TLS to the SCN ITS-SU.	x	x	x
IDX device application private keys	Private keys (pairwise to the IDX device application certificate public keys) belonging to the IDX device that can be used to authenticate itself to the SCN ITS-SU using ISO/TS 21177.	x	x	x
Application encryption public keys	An entity's application-specific 1609.2 encryption public key, typically embedded in a 1609.2 certificate.			x
Application encryption private keys	The pairwise private key for the encryption public key. This key is not shared/disclosed by the owner. It is used to perform an ECIES encryption over data.			x
SCN ITS-SU application certificates	Certificate(s) presented by the SCN ITS-SU when establishing secure TLS sessions with the IDX device. These certificates and the SCN ITS-SU's private keys can also be used for non-repudiation on transactions/data coming from the SCN ITS-SU.	x	x	x
Revocation information	Information sourced from applicable PKIs that indicates the revocation status of CA and end entity certificates. Can be in the form of CRLs or OCSP-type services/structures.	x	x	x

6.4 Threats

6.4.1 General

This subclause provides a listing and description of threats against the IDX device, its assets and environment according to the different data types that the IDX device is intended to access in the SCN.

Threats are considered from the following perspectives:

- **Threat sources/actors**: Some threats are device-oriented and can be local or remote to the device and its intended operational environment, i.e. threats are not necessarily local to the device or its immediate physical proximity.
- **Device threats are data threats**: The purpose of this threat model is to ascertain required protections for a generic ITS-S that can or cannot be vehicular in nature. Accordingly, the three data access scenarios are threat modeled both from a device and data perspective, i.e. the impact of certain device compromises necessarily assumes that the data it was protecting is also compromised.
- **System threats**: System-level threats are considered for overall impact to either the owner or other stakeholders of the SCN (e.g. surrounding drivers) and the larger ecosystem. For example, the document also addresses device-oriented PKI threats.
- **PKI ecosystem threats**: Associated with the TOE environment is the larger PKI ecosystem serving that environment. To that end, this document highlights threats to the PKI, which, if not mitigated, can flow down substantial impacts to the TOE and its operating environment.

The threats identified in this section are neither exhaustive nor tailored for any particular ITS domain. Given that the scenarios are genericized in terms of data sensitivities, the threats identified for each are based on assumptions and rationale listed for each. Developers of specific applications are expected to carry out their own analysis of security assumptions and requirements, using this section as a template.

6.4.2 Threat modelling process

The threat modelling in this technical report loosely follows the formalized process of the common criteria (ISO 15408-1 and ISO 15408-2). However, it is abbreviated due to the genericized nature of the data types and devices. The threats are principally attack-oriented.

Prior to starting the threat modelling, the following pre-requisite actions are performed:

a) Characterize the data sensitivities for each Scenario, based on generic examples of those data types; see 6.2.3.

b) Model and depict the IDX device as a generic ITS-S; see 6.2.2.

c) Model and depict the operational environment; see 6.2.2:

   1) peer devices,

   2) assumed communication modes, i.e. the ability to communicate both X.509 TLS and ISO/TS 21177 secure communications, and

   3) assumed capabilities and/or services, such as an external authorization service.

The threat modelling process relevant to the development or validation of PP material includes the following steps:

a) Identify threat categories and attack vector types.

b) Characterize different attack motivations.

c) Identify the threats.

d) For each threat, provide a qualitative risk rating based on impact and probability.

6.4.3 Threat categories and actor motivations

Threat categories factored into modelling are presented in Table 2:

Table 2 — Threat categories and vectors

Threat category or attack vector	Attack vector type
Supply chain	Cyber
Maintenance environment	Cyber
External network connection	Cyber
External shared or infrastructure services	Cyber
Trusted or partner network connection	Cyber
Internal network	Cyber
Internal shared or infrastructure services	Cyber
Mobile or transiently connected devices	Cyber
Authorized actions of non-privileged user	Cyber
Authorized actions of privileged user	Cyber
Device port (e.g. removable media)	Cyber
Data	Cyber
Immediate physical proximity	Physical
Indirect attack (e.g. tampering with HVAC, etc.)	Physical
Privileged user	Human (intended effects: coercion, subversion, deception, incapacitation)
Normal user	Human (intended effects: coercion, subversion, deception, incapacitation)
External actor	Human (intended effects: coercion, subversion, deception, incapacitation)
Maintainer	Human (intended effects: coercion, subversion, deception, incapacitation)
Developer/integrator	Human (intended effects: coercion, subversion, deception, incapacitation)
Denial of service	Human or Cyber
System error	Other
Misconfiguration (intentional or accidental)	Human

Threat/Attacker motivations factored into the modelling are provided in Table 3.

Table 3 — Possible attacker motivations

Motivation	Description
Ideology	The attacker is motivated by a fundamental belief that the attack is needed and justifiable.
Coercion	The attacker seeks for compel some action or behaviour of the victim or a stakeholder to the attack.
Notoriety	The attacker seeks publicity and fame, whether or not the attacker is individually credited.
Personal satisfaction	The attacker's motivation is to satisfy a basic urge or inclination to perform the attack.
Organizational gain	Competitive advantage.
Financial gain	Financial gain (fraud against or theft from the organization, acquisition of sellable/usable PII, acquisition of sellable/usable competitive information), extortion, fraud against or theft from the organization's customers, suppliers, or partners).
Disgruntlement	Personal motives (attention, malice/resentment, acquisition of PII about targeted individuals).
Accidental	The realization of the threat is accidental, i.e. not purposeful.

Table 3 *(continued)*

Motivation	Description
Dominance	The attacker seeks to overcome and/or overwhelm the victim or another stakeholder.
Unpredictable	The attacker is unstable or otherwise unpredictable in his/her motivations.
Positional/stepping-stone	Acquisition of a launching point for targeted attacks, acquisition of resources that can be used in targeted attacks, acquisition of intelligence about other entities.
Tracking/stalking	Exploitation of the trackability of the device or its data in order to stalk.

6.4.4 Scenario comparison of threats

Table 4 identifies threats relevant to the three data sensitivity scenarios. For each, a description is provided in addition to a mapping to one or more of the three data sensitivity scenarios.

Table 4 — Scenario-based applicability and comparison of threats

Threat	Description	S1	S2	S3
T.THEFT	An attacker could try to gain illicit knowledge of the IDX design and thereby more easily mount an attack to compromise the integrity or authenticity the data processed by the device.	x	x	x
T.PHYSICAL_TAMPER	An attacker could attempt to access the internal components of the TOE to bypass software security controls and extract data including firmware which could lead to exposure of default passwords and other information.	x	x	x
T.INTERFACE_EXPLOITATION	Attacker is able to gain access to an IDX device through misconfigured or insecure physical or logical interfaces. Note that this threat is not indicated for Scenario 1 data sensitivities. However, it is possible for exploitation of the interface to target resource exhaustion of the ITS-S. If resource exhaustion is a threat to a design and implementation, then it should be included for Scenario 1.		x	x
T.UNAUTHORIZED_LOCAL_TRUST_CHAIN_MODIFICATION	An attacker is able to gain write access to the IDX device trust store, resulting in the ability to delete existing root or other CA certificates that would cause message validation failure or to add new root/CA certificates that would introduce unexpected and unauthorized trust relationships.	x	x	x
T.ENVIRONMENT_CA_COMPROMISE	An attacker is able to compromise one of the SCMS CAs resulting in illegitimate issuance of trusted certificates that allow SCN ITS-SU and IDX devices to spoof legitimate system components.	x	x	x
T.ENVIRONMENT_UNREPORTED_MISBEHAVIOUR_MALFUNCTION	Misbehaving components are not reported in a timely manner for revocation action, resulting in the ability for a compromised IDX device or SCN ITS-SU to establish trusted relationships with other components until the revocation process is completed.	x	x	x
T. PRIVATE_KEY_DISCLOSURE	An attacker is able to obtain the private keys for authentication of the IDX device, allowing the attacker to masquerade as the device and perform trusted functions with the SCN ITS-SU. Note that this threat is not indicated for Scenario 1 data sensitivities. However, it is possible for disclosure of private key material to be a prerequisite to targeting resource exhaustion of the ITS-S. If resource exhaustion is a threat to a design and implementation, then disclosure of private key material should be included for Scenario 1.		x	x
T.EPHEMERAL_OR_SESSION_KEY_DISCLOSURE	An attacker is able to intercept the session keys generated using ECIES resulting in a loss of confidentiality for data exchanges between the IDX device and SCN ITS-SU.		x	x

Table 4 *(continued)*

Threat	Description	S1	S2	S3
T.UNAUTHORIZED_PRIVATE_KEY_USE	An unauthorized user is able to gain access to an unprotected private key and use the key for messaging and transactions between the IDX device and SCN ITS-SU, allowing an unauthorized user to perform potentially privileged transactions or masquerade as authorized system components. Note that this threat is not indicated for Scenario 1 data sensitivities. However, it is possible for unauthorized private key use to be a prerequisite to targeting resource exhaustion of the ITS-S. If resource exhaustion is a threat to a design and implementation, then disclosure of private key material should be included for Scenario 1.		x	x
T.SOFTWARE_FIRMWARE_TAMPER_IN_TRANSIT	An attacker could alter the firmware of the IDX device during or prior to the update process to introduce malicious functionality or backdoors or identify ways to take root permissions on the IDX device. In this way, an attacker achieves potentially privileged operation of the IDX device.		x	x
T.SOFTWARE_FIRMWARE_TAMPER_AT_REST	An attacker could gain access to IDX device firmware by extracting from the device and then alter the firmware to a malicious one or identify ways to take root permissions on the device. In this way an attacker achieves potentially privileged operation of the IDX device.		x	x
T.SOFTWARE_FIRMWARE_INTEGRITY_ERROR	IDX firmware validation results in an integrity error, disrupting the operations and availability of the IDX device.	x	x	x
T.UNAUTHORIZED_ACCESS_POLICY_MODIFICATION	An attacker is able to obtain write access to the IDX device access policy configurations, resulting in the ability for the attacker to grant him/herself permissions for performing sensitive functions.		x	x
T.LACK_OF_REVOCATION_INFORMATION_AVAILABILITY	The IDX device is unable to retrieve the latest revocation information, resulting in the device potentially trusting compromised/revoked certificates until revocation information is available.	x	x	x
T.PRIVILEGE_ESCALATION	A logged in user to the IDX device is able to escalate privileges on the device to perform functions that were not authorized for his/her role.		x	x
T.UNAUTHORIZED_ACCOUNT_ASSUMPTION	Users could try to access functions not allowed to them when previous user has not properly logged out, to modify or delete user data, download sensitive SCN information or modify downloaded SCN data.		x	x
T.EAVESDROPPING	An attacker exposed to the communications medium between the IDX device and SCN ITS-SU is able to tap into the communications and decode messaging resulting in a loss of confidentiality.		x	x
T.ENIVIRONMENT_NETWORK_ATTACKER_IN_THE_MIDDLE	A successful modification of data exchanged between the IDX device and SCN ITS-SU would allow an attacker to view or change configurations, upload new software, or download sensitive data.		x	x
T.ENVIRONMENT_SCN_SPOOFING	SCN is masquerading as a genuine SCN ITS-SU sending false information in messages that are otherwise valid.		x	x
T.DEVICE_SPOOFING	Equipment masquerading as a genuine IDX device is able to upload new software or change configurations in the SCN.		x	x
T.ENVIRONMENT_UNKNOWN_COMPROMISED_PEER	The IDX device is threatened due existence of a compromised SCN ITS-SU peer; data loss, privacy losses, integrity losses.		x	x
T.ENVIRONMENT_EXECUTION_SENSITIVE_COMMANDS_ON_COMPROMISED_PEER	The IDX device connects to a compromised SCN ITS-SU system and executes sensitive commands on it. In this case, a loss of integrity of the peer device could mean that the sensitive command invoked by the IDX device is modified, doesn't work correctly, or is otherwise obstructed or hijacked. Commands executed on integrity violated/compromised SCN ITS-SU peers are modified prior to execution resulting in a loss of its functionality.			x
T.ENVIRONMENT_ACCESS_PRIVACY_PROTECTED_DATA_WITHOUT_CONSENT	Privacy protected data is transmitted from the SCN ITS-SU to the IDX device and accessed without data owner's explicit permission.		x	

Table 4 *(continued)*

Threat	Description	S1	S2	S3
T.ENVIRONMENT_SCN_SOFTWARE_IMAGE_TAMPER	A tampered software image is received by the IDX device from the OEM or supplier. This threat can leverage local or remote access to firmware files prior to the IDX device's download from the manufacturer and staging to the SCN via the SCN ITS-SU. Includes access to OEM/supplier web services repositories. If the device cannot identify that the software image has been altered, then it will stage a potentially malicious image for loading into the SCN.		x	x
T.ENVIRONMENT_SCN_SOFTWARE_IMAGE_REVERSE_ENGINEER	IDX device is physically accessed allowing extraction of staged software image from storage. Once extracted from storage, the binary can be reverse engineered to support security analysis by an attacker leading to identification of vulnerabilities, default passwords, software library versions (e.g. vulnerable versions) and other security intelligence.		x	x
T.MALICIOUS_SOFTWARE_LOAD_TO_SCN	An IDX device loads a malicious software image into the SCN because it did not either 1) validate the software signature itself prior to staging, or 2) maintain a trusted storage environment on itself for all software image staging.		x	x
T.LOAD_KNOWN_VULNERABLE_SOFTWARE_VERSION	IDX device loads a previous version of software to vehicle. Previous software version contains known vulnerabilities, thus constituting a downgrade attack.		x	x
T.SOFTWARE_METADATA_MODIFICATION	IDX device allows an attacker to access and modify the metadata associated with a software image file. Metadata can be used by vehicle components to support loading and validation of cryptographic signatures.		x	x
T.ENVIRONMENT_SCN_ACCESS_POLICY_UPDATE	An attacker is able to modify the access control policy of the SCN ITS-SU, allowing anyone to load firmware or read/write sensitive data and/or functions.		x	x
T.DATA_LEAKAGE	Attackers could try to leak data from the IDX device to third parties who should not receive this data. Data includes user account information, sensitive vehicle/owner data, keys and certificates.		x	x
T.MALICIOUS_CODE_INJECTION	A compromised IDX device could introduce malicious code into the SCN firmware or software executable.		x	x
T.SCN_DATA_TAMPERING	An authorized attacker can access data and reports and change details (e.g. emissions logs, maintenance history) prior to providing them to a vendor, regulatory authority, etc.		x	x
T.USER_ACCOUNT_MASQUERADE	An unauthorized user is able to access the IDX device due to no password protections or weak password protections.		x	x
T.DATA_TO_DEVICE_OWNER_LINKAGE	Asset: PII/tracking data. It is assumed the SCN or its fronting SCN ITS-SU has data, which if accessed could link an individual to the data. If this data is intended to be accessed for external use, some applications or policies can require that it either 1) be accessed by an authorized party, or 2) undergo some type of sanitization prior to the access (i.e. to make the data public).		x	x
T.ENVIRONMENT_PKI_SPOOFING	IDX device stores reachability information and trust anchors for connecting to a PKI. An attacker can modify the stored PKI location (e.g. URL) and/or the trust anchors needed to connect to the PKI, thus allowing the attacker to perform MITM attacks against the PKI.		x	x

6.5 Security objectives

6.5.1 Summary and comparison by scenario

This subclause provides a set of derived security objectives that security practitioners can use in building or comparing PPs. Table 5 provides the security objectives, based on threats identified in 6.4, required to counter the threats specific to each data sensitivity scenario.

NOTE The security objectives are high level definitions of security capabilities of a TOE, and one or more common criteria SFR are designed to map to each; see 6.6 for SFR mappings. A PP includes the full traceability of a) threats to security objectives, and b) security objectives to SFR (and in some cases organizational policy).

In addition to security objectives, Table 6 provides a listing and brief description of organizational policies needed to counter threats. This document does not map organizational policies to SFRs, as this level of granularity is much more specific to the applications and environments at hand. Annex A provides the full mapping of both security objectives and organizational policies to the defined threats.

If an IDX device is expected to host applications of more than one of the identified types, the device will be expected to be conformant with the PP requirements for each application.

Table 5 — Scenario-specific security objectives

Security objective (SO)	Description	S1	S2	S3
General platform objectives				
OT_PLATFORM_USER_MGMT	Securely manage users including the creation, update and deletion of user accounts and allocation to roles/groups.		x	x
OT_PLATFORM_PRIVILEGED_USER_AUTHORIZATION	Ensure that only registered legitimate users are able to gain access to system's file system and services with privileged rights.		x	x
OT_PLATFORM_SECURE_KEY_MGMT	Provide secure key management and provisioning of platform cryptographic key material.	x	x	x
OT_EXT_FW_PROTECT	Device validates signatures on all software/firmware loads. It is assumed that this is applied only to external firmware intended to be loaded to the SCN.			x
OT_PLATFORM_AUTHENTICATION	The TOE requires authenticated access for all services.		x	x
OT_PLATFORM_SECURE_BOOT	Device performs secure boot.		x	x
OT_PLATFORM_DATA_AT_REST_PROTECT	Device uses trusted storage to secure data at rest.		x	x
OT_PLATFORM_MALWARE_PROTECT	Device guards against the installation and execution of malware.			x
OT_PLATFORM_LOG	Device logs security-sensitive events.		x	x
OT_PLATFORM_TAMPER_PROTECT	Device guards against tamper.		x	x
OT_PLATFORM_COMM_INTERFACE_PROTECT	The TOE does not expose insecure network services, has firewall functionality and enforces against a policy.	x	x	x
OT_PLATFORM_PHYSICAL_INTERFACE_PROTECT	Access to all physical ports is authenticated.	x	x	x
OT_PLATFORM_REMOTE_CONNECTION_PROTECT	All remote connections are secured.		x	x
OT_PLATFORM_TRUST_CHAIN_PROTECT	The TOE ensures the integrity of the local trust chain.	x	x	x
OT_PLATFORM_CERT_REVOCATION	Device monitors for updates to certificate revocation information (CRLs, OCSP, etc.)	x	x	x
OT_EXT_FW_PROTECT	Device firmware is signed by the developer.			x
OT_EXT_FW_PROTECT	Device firmware is encrypted.			x
OT_PLATFORM_ALERT_INFORM	The device issues alerts/notices for significant security events including lack of available revocation information.		x	x
OT_PLATFORM_SESSION_PROTECT	Service device applications enforce screen timeouts/locks and require re-authentication after a configurable period of timeout.		x	x
OT_PLATFORM_AC_POLICY_PROTECT	Access policy is integrity protected and requires authentication for access.		x	x
OT_PLATFORM_RELIABLE_TIME	Reliable timestamping required for secure authorization checks for applications.		x	x
Device application security objectives				
OT_APP_AUTHENTICATION	Users authenticate before the application provides any services.		x	x
OT_APP_SECURE_KEY_MGMT	Provide secure key management and provisioning of application-specific cryptographic key material.	x	x	x
OT_APP_VALIDATE_PEER_STATE	Device can validate the current peer's security state through attestation transaction prior to loading or retrieving sensitive data.			x
OT_APP_ANONYMIZATION	Data is anonymized to ensure that privacy protected data cannot be later tied to a specific SCN operator or data owner.		x	x
OT_APP_THIRD_PARTY_AUTHORIZATION	Owner consent is required for access to all sensitive information.		x	x
OT_APP_ALERT_INFORM	Owners are notified of all attempts to access sensitive information.		x	x
OT_APP_USER_MANAGEMENT	Ensure that only registered legitimate users are able to gain access to system's file system.		x	x
OT_APP_DATA_AT_REST_PROTECT	Data received from a SCN ITS-SU as well as device-internal data (e.g., Configuration files, firmware) is encrypted and access controls applied.		x	x
OT_APP_EXT_FW_PROTECT	Service device validates signatures on firmware to be loaded.			x

Table 5 *(continued)*

Security objective (SO)	Description	S1	S2	S3
OT_APP_CERT_REVOCATION	Application monitors for updates to CRL.	x	x	x
OT_APP_TRUSTED_STORAGE	IDX implements integrity protection on user data stored within the device.		x	x
OT_APP_SECURE_BOOT	IDX implements secure boot process to guard against malicious code and identify integrity failures.			x
OT_APP_AC_POLICY_PROTECT	System provides integrity protection for access policy and requires authentication for access or update.		x	x
OT_APP_ROLLBACK_RECOVERY	Rollback sensitive configuration changes or operations, especially during configuration of a SCN.			x
PKI provider security objectives				
OT_PROV_KEY_MGMT_AUTHORITY_REVOCATION	CA infrastructure processes and communicates authority revocation list information and revokes all certificates issued under a revoked CA.	x	x	x
OT_PROV_KEY_MGMT_AUTHORITY_MISBEHAVIOUR_DETECTION	Key management infrastructures that manage keys for device implement misbehaviour detection systems.	x	x	x

Table 6 provides a listing of organizational policies that should be incorporated to counter threats to IDX devices and their environments.

Table 6 — Organizational Policies

Organizational Policy	Description
P. LOG_ASSET_INVENTORY	The environment that a TOE is used or stored within should implement procedures to log any removal of the device from the environment.
P.PHYSICAL_ACCESS	The environment that a TOE is used or stored within provides proper monitoring/escort for unauthorized users.
P.PHYSICAL_MONITORING	The environment that a TOE is used or stored within is be monitored for unauthorized entry after hours.
P.SECURE_MANAGEMENT	The TOE provides management means for the authorized administrator to manage the IDX device in a secure manner.
P.DATA_UNLINKABILITY	The IDX device protects data that is linkable to individuals.
P.UNLINKABILITY_OF_TRANSMITTED_DATA	The IDX device and SCN ITS-S secure session and transport protocol disassociate session information and artifacts from data owners.
P.MISBEHAVIOUR_REVOCATION_REPORTING	The organization implements policies and procudures to 1) detect misbehaviour, 2) obtain misbehaviour metadata, 3) report misbehaviour and, as needed, 4) request misbehaviour-based revocation of devices.

6.5.2 Analysis

The security objectives defined in Table 5 provide a defence-in-depth approach towards securing an IDX device that operates under the constraints of one of the three defined scenarios.

Scenario 1 is focused on processing only non-sensitive data, and it was assumed that any connectivity to the SCN ITS-SU requires a secure connection as per ISO/TS 21177 and also to satisfy a basic need in the environment to promote honest actors and devices that have undergone basic provisioning. This drives the allocation of security objectives that support secure key management and secure interface/connectivity. Additionally, security objectives have been allocated to Scenario 1 that focus on good cyber hygiene in order to limit the likelihood that a legitimate but infected requestor device attempts to connect to sensitive SCN systems. Security objectives include malware protection and session protection. Also identified are application-layer security objectives that would be allocated to the various software applications expected to be installed on an IDX device. These security objectives align with the objectives allocated to the IDX device itself, with the understanding that some objectives can be demonstrated by the host (IDX device) instead of the application.

Scenario 2 security objectives build upon the objectives allocated to Scenario 1. Scenario 2 is based on the receipt, storage and/or processing of sensitive data, privacy-protected data, and data that can support regulatory decision making (e.g. emissions test results). Thus, Scenario 2 security objectives are focused heavily on the design/implementation of a secure platform and secure management of users and roles. Data at rest protections, tamper protections, and remote connectivity protections are included within the allocated objectives. Additionally, alerting objectives are incorporated to provide enhanced situational awareness to operators of the IDX device of its associated infrastructure that a device can be under attack or compromised. Also note that anonymization was included as a security objective to scenario 2 IDX applications. The rationale for including an anonymization capability is that data collected from a SCN should, in some Scenario 2 cases, not be traceable back to the SCN, its owner and frequently a vendor associated with it.

Scenario 3 objectives again build upon those allocated to Scenarios 1 and 2, but add additional protections related to the security of firmware that might be stored and passed for installation to a sensor network. Note that although scenario 3 objectives can be thought of as more stringent than scenario 2, each scenario can be viewed as standalone from the perspective of security objectives. For example, security controls can be identified at a later time that should be allocated to Scenario 3, but not to Scenarios 1 and 2.

6.6 SFR and rationales

Table 7 identifies and maps common criteria SFRs[9] to SOs defined for the device and environment. The intent of the SFRs is to fulfil the security objectives' mitigation of the associated threats.

Table 7 maps the SFRs required to satisfy each indicated security objective, based on the threats that security objective counters according to each scenario. For each, a rationale is briefly provided along with scenario-specific constraints, exclusions and/or justifications.

SFRs are frequently found across more than one security objective. Like the definition of security objectives, a PP developer aggregates and reconciles the list of applicable SFRs for the device's determined set of security objectives. Rows in the following table can be directly used when developing and characterizing various ITS-SU PPs.

Table 7 — Security functional requirements for each security objective

Security objectives (SO)	SFR description and rationale
OT_PLATFORM_USER_MGMT	FIA_ATD: Required to identify and provide metadata on users of the device. Scenario exceptions: Not needed for Scenario 1 as it is assumed no user accounts are needed to access public data. FIA_USB: Required to maintain association of user's security attributes to subject information contained in a 1609.2 credential. Scenario exceptions: Not needed for Scenario 1 as it is assumed user to subject security attributes are non-existent or informal. FIA_UAU: Required to enforce user authentication rules. Scenario exceptions: Not needed for Scenario 1 as it is assumed user to subject security attributes are non-existent or informal. FMT_SMF: Required for definition of security management functions allocated to device administrators. Scenario exceptions: Not needed for Scenario 1 as it is assumed security management is minimal, non-existent or performed on the backend by the device manufacturer or service provider. FMT_SMR: Required for managing users, security roles and accesses for Scenarios 2 and 3. Scenario exceptions: Not needed for Scenario 1 as it is assumed no user accounts are needed to access public data.

Table 7 *(continued)*

Security objectives (SO)	SFR description and rationale
OT_PLATFORM_PRIVILEGED_USER_AUTHORIZATION	This is not needed for a Scenario 1 application since there are no access controls needed or enforced on the IDX device. FIA_ATD: Required to identify and provide metadata on users of the device. FIA_UAU: Required to enforce user authentication rules and how these are implemented with respect to ISO/TS 21177 and its authentication/authorization functions. Scenario exceptions: Not needed for Scenario 1 as it is assumed user to subject security attributes are non-existent or informal. FIA_UID: Required to establish conditions for users to identify themselves as part of an authentication or authorization function. FIA_USB: Required to maintain association of user's security attributes to subject information contained in a 1609.2 credential. One such attribute is the user's security role, which can be directly encoded in a 1609.2 certificate used in ISO/TS 21177. FMT (all): Security management required to establish and maintain access controls from subject users to resources in the ISO/TS 21177 context. Revocation (FMT_REV) also needed from both a user management and PKI perspective to remove bad actors (accounts and/or certificates) from the system. FDP_ACC: Required in order to establish, maintain and enforce an access control policy which addresses, minimally, a subset of the resources to which subjects have access. In some cases, a complete access control capability can be needed to stipulate all operations to which a subject has access on a given resource. FDP_ACF: Required to establish a rich set of access control rules on privacy-protected or high-sensitivity operations that the IDX user(s) can invoke.
OT_PLATFORM_SECURE_KEY_MGMT	FCS_CKM: Secure key management required for any ISO/TS 21177-enabled IDX device in order to securely generate and handle 1609.2 certificates and associated crypto material. While this also applies to Scenario 1, in many cases secure key management in Scenario 1 would likely be performed by the manufacturer vs. the users of the device. FCS_CKM.4 (key destruction) is not likely needed for Scenario 1. FCS_COP: Includes all of the relevant 1609.2/TLS/ISO21177 cryptographic operations.
OT_EXT_FW_PROTECT	FDP_DAU: Required to authenticate external firmware the IDX will accept and load to a SCN system. This is specifically allocated to external firmware, not firmware intended for the IDX device itself. FCS_COP: Specifies all cryptographic operations pertaining to the authentication and integrity checks on external firmware. FDP_ITC: Required in order to import and manage data, for example firmware, outside of the IDX device's control. Needed to protect the association of user data and security attributes associated with external firmware the device manages. FDP_ETC is not deemed necessary in Scenario 3 since it is assumed that external firmware loads and their security attributes are fully protected/authenticated from the manufacturer. If some security attributes are added by the IDX device during firmware export to a SCN, then FDP_ETC becomes necessary. FDP_ITT: Potentially required in Scenario 3 to protect the staging processes of SCN firmware within the IDX device. This is not likely necessary in all Scenario 3 instances, however.
OT_PLATFORM_AUTHENTICATION	NOTE Not needed in Scenario 1 due to lack of user authentication to the IDX device. FIA_UAU: Required to enforce user authentication rules and how these are implemented with respect to accessing sensitive services in Scenarios 2 and 3. FMT_SMR: Required for managing users, security roles and their associated authentications in Scenarios 2 and 3. FIA_ATD: Required to identify and provide metadata on users of the device. FIA_UID: Required to establish conditions for users to identify themselves as part of authentication. FIA_USB: Required to maintain association of a user's security attributes to its subject role information. Authentication data is associated as such.

Table 7 *(continued)*

Security objectives (SO)	SFR description and rationale
OT_PLATFORM_SECURE_BOOT	This is not needed for Scenario 1 as secure boot is assumed unnecessary. A secure boot capability in the SCN is necessary, however, as it is likely to perform sensitive access control checks even for an exclusively public data-accessing IDX device. FIA_AFL: This is required if user authentication is a component precursor to a secure boot operation. For example, if the user fails a repeated number of times to successfully authenticate, FIA_AFL conditions are needed to manage the condition. Also, vendor authentication can be embedded into the secure boot operation. FIA_USB: Required to maintain association of a user's security attributes to its subject role information. Any authentication data evaluated in secure boot can require this. FPT_TST: A component of the secure boot operation is the self-testing associated with secure boot. For example, memory checks and integrity checks are needed. FPT_SBT **: This is required to enforce secure boot. FDP_DSK **: This is required to perform secure data encryption **It is to be noted that:** 1) FPT_SBT would be an extension to the common criteria SFR families. It would require secure management of KEK for disk decryption, verification of the integrity of the OS and handling upon failure of verification. 2) FDP_DSK would be an extension to the common criteria SFR families. It would require data encryption in accordance with approved cryptographic algorithms and secure management of DEKs.
OT_PLATFORM_DATA_AT_REST_PROTECT	This is not needed for Scenario 1 as data protection of any kind is not deemed necessary. FDP_DSK **: This is an extension SFR required to perform secure data encryption. It would require data encryption in accordance with approved cryptographic algorithms and secure management of DEKs. FDP_DAU: Required to authenticate external firmware in Scenario 3 and authenticate data and/or reports in Scenario 2, for example for regulatory purposes. FDP_ITT: Potentially required in Scenario 3 to protect the staging processes of SCN firmware within the IDX device. In Scenario 2, the device can stage reports or other data from the SCN for regulatory purposes. FDP_SDI: Required in Scenario 3 to protect the staging processes of SCN firmware within the IDX device. For example, the device can verify the integrity of staged data as a prerequisite to loading/exporting it, thereby providing further isolation and protections to SCN systems.
OT_PLATFORM_MALWARE_PROTECT	FPT_ITT: Internal data transferring protections within the device are needed to reduce the likelihood of malware-infected data or firmware entering sensitive processes or being accessed or transferred. This is not likely needed in Scenario 1 or 2. FAU_ARP: Process isolation and alarms are needed for IDX devices to help protect against malware threats. FPT_PHP: Malware protections can be augmented by physical protections and alarms associated with physical access to memory and/or unexposed or disabled physical interfaces. FAU_SAA: Violation and anomaly detection and response are needed in Scenario 3, given the ability of the device to access and load/execute on a SCN system.
OT_PLATFORM_LOG	This control is not needed in Scenario 1. FAU_GEN: Security-relevant events are recorded. Specific events will be application-dependent. FAU_ARM: Automatic audit response can be needed in high-sensitivity devices such as used in Scenario 3.
OT_PLATFORM_TAMPER_PROTECT	FPT_PHP: Tamper detection and response are needed when handling Scenarios 2 and 3 sensitive data. FPT_TST: A component of tamper detection and response is the self-testing needed to provide a guarantee of platform integrity. For example, memory checks and integrity checks are needed.

Table 7 *(continued)*

Security objectives (SO)	SFR description and rationale
OT_PLATFORM_COMM_INTERFACE_PROTECT	FPT_ITT: Internal data transferring protections within the device are needed to reduce the likelihood of accessing or corrupting data transferred using one interface with that of another interface. This ensures internals of the platform help enforce external interface separations. FTP_TRP: Trusted path is needed for secure management interfaces exposed to users, especially when the IDX device is accessing sensitive SCN resources via the SCN ITS-SU. FDP_UCT: This is required to provide confidentiality to remote user management services as well as the TLS connections between the IDX device and SCN ITS-SU and the IDX device and other sensitive, external services. FTA_TSE: The IDX device needs to be able to enforce secure session establishment when external users/services connect to it.
OT_PLATFORM_PHYSICAL_INTERFACE_PROTECT	FPT_PHP: Physical protections and alarms are needed to protect physical access to exposed and/or unexposed or disabled physical interfaces.
OT_PLATFORM_REMOTE_CONNECTION_PROTECT	This is assumed to be unnecessary for Scenario 1 as it does not handle any sensitive data and is likely locally accessed and used. FPT_ITT: Internal data transferring protections within the device are needed to reduce the likelihood of accessing or corrupting data transferred using one remote access interface with that of another interface. This ensures internals of the platform help enforce the separation of remote interfaces. FTP_TRP: Trusted path is needed for secure management interfaces exposed to remote users. FDP_UCT: This is required to provide confidentiality to remote user management services as well as the TLS connections between the IDX device and SCN ITS-SU and the IDX device and other sensitive, external services. FTA_TSE: The IDX device needs to be able to enforce secure remote session establishment when external users/services connect to it. FIA_UID: Required to establish conditions for users to identify themselves as part of authentication when establishing a remote session with the IDX device.
OT_PLATFORM_TRUST_CHAIN_PROTECT	FCS_CKM: Secure key management is required for any ISO/TS 21177-enabled IDX device in order to securely manage roots of trust and validate certificates according to those roots. While this also applies to Scenario 1, FCS_CKM.4 (key destruction) is not likely needed for Scenario 1.
OT_PLATFORM_CERT_REVOCATION	FCS_CKM: Secure key management revocation checking functions are required for any ISO/TS 21177-enabled IDX device in order to securely validate endpoints against known roots of trust. While this also applies to Scenario 1, FCS_CKM.4 (key destruction) is not likely needed for Scenario 1. FMT_REV: Revocation capabilities are needed when needing to manually remove a trust relationship within the IDX device.
OT_PLATFORM_ALERT_INFORM	This is not applicable to Scenario 1 given the lack of sensitive data. FAU_ARP: Event detection and response functions are implemented to provide alerts and inform users of security incidents. FAU_STG: Audit data storage requirements are specified as a prerequisite to collecting and storing alert data.
OT_PLATFORM_SESSION_PROTECT	This is not applicable to Scenario 1 given the lack of sensitive data. FTA_SSL: An IDX device has the ability to restrict inactive sessions from being resumed and assumed by an unauthenticated individual. This is needed to protect the SCN from unattended IDX devices that can be in an authenticated state.

Table 7 *(continued)*

Security objectives (SO)	SFR description and rationale
OT_PLATFORM_AC_POLICY_PROTECT	This is not applicable to Scenario 1 given the lack of sensitive data and lack of need for an access control policy on the device. FMT_MSA: The management of security attributes associated with subjects and resources is a necessary prerequisite to protecting the access control policy itself. FMT_MTD: This control is needed to manage the security settings and the accesses to the access control policy. FMT_SMF: This control is needed to manage the security settings and the accesses to the access control policy. FMT_SMR: This control is necessary to establish and manage the security roles and their accesses to the access control policy. FDP_ACC: Unique naming, structure and attributes are needed for the access control policy and its protection. FDP_ACF: Secure access control functions are implemented in order to protect the device and the sensitive access control policy it enforces.
OT_PLATFORM_RELIABLE_TIME	FPT_STM: The IDX device, in order to securely perform some access control functions or perform 1609.2 relevance and consistency checks, will require a reliable timestamp capability. This is also needed for reliable audit generation.
Device application security objectives These objectives should apply to application developers in cases where the ITS-S platform supports third party applications on the IDX device platform.	
OT_APP_AUTHENTICATION	Not needed in Scenario 1 due to lack of user authentication to the IDX device. FIA_UAU: Required to enforce user authentication rules and how those are implemented with respect to accessing sensitive application-specific services in Scenarios 2 and 3. FMT_SMR: Required for managing application-specific users, security roles and their associated authentications in Scenarios 2 and 3. FIA_ATD: Required to identify and provide metadata on users of the application. It is assumed that applications maintain and enforce their own access control policies, or leverage capabilities of the platform to do so. FIA_UID: Required to establish conditions for users to identify themselves as part of application-specific authentication. FIA_USB: Required to maintain association of a user's security attributes to its subject application role information. Application-specific authentication data is associated as such.
OT_APP_SECURE_KEY_MGMT	Assume applications have their own unique key management needs and that the platform key management capabilities support them, for example via HSM, secure storage and relevant key management application interfaces. FCS_CKM: Secure key management revocation checking functions are required for any ISO/TS 21177-enabled IDX device in order to securely validate endpoints against known roots of trust. While this also applies to Scenario 1, FCS_CKM.4 (key destruction) is not likely needed for Scenario 1 FCS_COP: All cryptographic operations pertaining to secure key management functions are specified.
OT_APP_VALIDATE_PEER_STATE	FPT_TEE: The IDX device, prior to performing sensitive functions via the SCN ITS-SU, validates that the SCN is in an appropriate state. For example, it can validate the trust level and integrity of the endpoint prior to connecting and performing a firmware upgrade or sensitive configuration change.
OT_APP_ANONYMIZATION	FPR_ANO: Some services accessed on the IDX device by the SCN ITS-SU can require anonymity protections for the user or other stakeholder associated with the SCN.

Table 7 *(continued)*

Security objectives (SO)	SFR description and rationale
OT_APP_THIRD_PARTY_AUTHORIZATION	FDP_ACC: Unique naming, structure and attributes are needed for the application's access control policy and its protection while enforcing authorization checks using a third-party authorization provider. FDP_ACF: Secure access control functions are implemented in order to protect the application and the sensitive access control policy it levies and enforces on itself, including policies related to checking against a third-party access control provider. It is assumed that functions associated with third-party access control are also policy-mandated. FIA_UID: Required to establish conditions for users to identify themselves as part of an authentication or authorization function, in this case a third-party authorization call made by the IDX device or as initiated by the SCN ITS-SU to which it is connected. This functionality is supported in ISO/TS 21177. FIA_UAU: Required to enforce user authentication rules within the application. In this case, the user authentication is via a proxied third-party authorization provider.
OT_APP_ALERT_INFORM	This is not applicable to Scenario 1 given the lack of sensitive data. FAU_ARP: Application-specific event detection and response functions are implemented to provide alerts and inform users of security incidents. These would accommodate rules that are application-specific vs. platform-only. FAU_STG: Audit data storage requirements are specified as a prerequisite to collecting and storing alert data for specific applications.
OT_APP_USER_MANAGEMENT	This does not pertain to scenario 1 due to lack of sensitive data. FIA_ATD: Required to identify and provide metadata on users of the application. This is unique from users or administrators of the platform itself. FIA_USB: Required to maintain association of user's security attributes to subject information contained in an application-specific 1609.2 credential as identified by its application identifier. FIA_UAU: Required to enforce user authentication rules within the application. FMT_SMF: Required for definition of security management functions within an application. FMT_SMR: Required for managing application-specific users, security roles and accesses for Scenarios 2 and 3.
OT_APP_DATA_AT_REST_PROTECT	This is not needed for Scenario 1 as data protection of any kind is not deemed necessary. This SFR targets protection of configuration files and other sensitive platform data for confidentiality. FDP_DAU: Required to authenticate external software and configuration files in Scenario 3 and authenticate data and/or reports in Scenario 2, for example for regulatory purposes. FDP_ITT: Potentially required in Scenario 3 to protect the staging processes of SCN software applications and configurations within the IDX device. In Scenario 2, the device can stage reports or other data from the SCN for regulatory purposes. FDP_SDI: Required in Scenario 3 to protect the staging processes of SCN application software within the IDX device. For example, the device can verify the integrity of staged data as a prerequisite to loading/exporting it, thereby providing further isolation and protections to SCN systems. FDP_DSK **: This is required to perform secure data encryption **It is to be noted that:** 1) FDP_DSK would be an extension to the common criteria SFR families. It would require data encryption in accordance with approved cryptographic algorithms and secure management of DEKs.

Table 7 *(continued)*

Security objectives (SO)	SFR description and rationale
OT_APP_EXT_FW_PROTECT	FDP_DAU: Required to authenticate external application software or configurations the IDX will accept and load to a SCN system. This is specifically allocated to external software, not software intended for the IDX device itself. FCS_COP: All cryptographic operations pertaining to the authentication and integrity checks on external firmware are specified. FDP_ITC: Required in order to import and manage data, for example software, outside of the IDX device's control. Needed to protect the association of user data and security attributes associated with external firmware the device manages. FDP_ETC is not deemed necessary in Scenario 3 since it is assumed that external software loads and their security attributes are fully protected/authenticated from the manufacturer. If some security attributes are added by the IDX device during software loading to a SCN, then FDP_ETC becomes necessary. FDP_ITT: Potentially required in Scenario 3 to protect the staging processes of SCN application software within the IDX device. This is not likely necessary in all Scenario 3 instances, however.
OT_APP_CERT_REVOCATION	FCS_CKM: Secure key management needed in some user or security attribute revocation functions. FMT_REV: Some applications to which the IDX device application can connect can be revoked. This can be certificate or account revocations.
OT_APP_TRUSTED_STORAGE	This is not pertinent to Scenario 1 given its lack of data sensitivity. This is meant for user data, specifically the integrity of the user data and any application-layer sensitive information. FDP_SDI: The application or its host is expected to monitor application data integrity within IDX device storage. The absence of this control poses risks to SCN equipment to which the IDX device connects and writes/executes.
OT_APP_SECURE_BOOT	FPT_TST: A component of necessary self-tests are those specific to the application. In some cases, the platform cannot perform these functions directly, though in some cases it can. This control is specific to the application and its self-testing associated with conditions defined for the application. FPT_SBT **: This is required to enforce secure boot. **It is to be noted that:** 1) FPT_SBT would be an extension to the common criteria SFR families. It would require secure management of KEK for disk decryption, verification of the integrity of the OS and handling upon failure of verification.
OT_APP_AC_POLICY_PROTECT	This is not applicable to Scenario 1 given the lack of sensitive data and lack of need for an access control policy on the device. FMT_MSA: The management of security attributes associated with subjects and resources is a necessary prerequisite to protecting an application's own access control policy. This is potentially the same mechanism as the platform AC policy, or is a different one. FMT_MTD: This control is needed to manage the application's unique security settings and the accesses to the application's access control policy. FMT_SMF: This control is needed to manage the application's unique security settings and the accesses to the application's access control policy. FMT_SMR: This control is necessary to establish and manage the security roles within the application and their accesses to security functions associated with the application. FDP_ACC: Unique naming, structure and attributes are needed for the application's access control policy and its protection. FDP_ACF: Secure access control functions are implemented in order to protect the application and the sensitive access control policy it levies and enforces on itself.
OT_APP_ROLLBACK_RECOVERY	FDP_ROL: Especially when accepting access commands from the user and invoking those commands on a connected SCN, many scenarios can exist where an unsafe or mistaken setting could cause significant harm. To mitigate these, the IDX device in Scenario 3 provides rollback capabilities per FDP_ROL. This is necessary, for example, if a firmware update command on the IDX device needs to be rolled back, or if it needs to support the rollback of firmware changes made on the connected SCN endpoint.

6.7 Comparison to other common criteria PPs

6.7.1 General

This subclause provides a high-level gap analysis on the coverage afforded by an existing set of PPs against the threats indicated for an IDX device performing in the three data sensitivity scenarios.

6.7.2 Summary and analysis of gaps

The threat analysis and resultant set of security controls were gap-analysed against existing PPs:

- C2CCC HSM PP
- V-ITS-S Base
- V-ITS-Comms PPs

The PPs were generally well aligned with typical threats associated with the three IDX device scenarios input to the threat analysis. However, there were several missing SFRs that can be required to support a secure implementation of an IDX device. The gaps in SFRs were not common across all of the scenarios in this document, however.

A key set of requirements missing from the current requirement allocations for the HSM[29] and V-ITS-Base PP[30] were associated with user identity and authentication.

Similarly, gaps were identified across PPs related to granular authorization capabilities in addition to detecting and responding to the presence of malware. Today's threats against V-ITS systems require security controls such as these in order to secure not only the component itself but also to keep bad actors from using that component as a launching point for attacks against a wider system.

6.7.3 Gap analysis with Car2Car HSM PP

The HSM PP was analysed in the context of Scenarios 2 and 3 described within this document. This drove requirements to protect the confidentiality and integrity of data processed by a host system embedding the HSM. An assumption was made that the host system would implement many of the defence-in-depth security controls needed for the system to operate, thereby alleviating many of the controls that would be allocated to the HSM embedded within the host. Even so, there were SFRs associated with Scenarios 2 and 3 that were identified as gaps in the HSM PP. These included:

- Identity and access management SFRs supporting the identification of a user prior to authenticating to the HSM. An HSM can frequently be used to help authenticated users in higher sensitivity scenarios.
- Role-based SFRs that require specific permissions assigned to roles that manage critical processes within the HSM, for example a crypto user role. Roles in the context of the HSM can be human or machine based.
- SRFs allocated to certificate revocation checking. HSMs have a role in the protection of trust chain management processes.
- Restrictions on who is allowed to access and manage restricted data.
- SFRs associated with the authentication process designed to limit feedback on authentication errors and enforce session timing restrictions.
- SFRs that enable an HSM to identify/mitigate attacks such as malware, especially those in critical embedded, hardware-protected areas.
- The protection of audit data on the HSM through its enablement of protected storage.

— The ability of an HSM to react to the identification of attempted intrusion, through zeroization, for example.

There are many possible uses and constraints placed on HSMs, based on the environment and host. Therefore, the gaps highlighted herein cannot be applicable in all cases.

Table 8 identifies potential gaps within the C2C HSM PP[29].

Table 8 — Possible gaps in Car-2-Car HSM PP

SFR	SFR family	SFR description	Gap
FIA_UAU.2	User authentication	User authentication before any action, requires that users are authenticated before any other action will be allowed by TSF.	HSM can require authenticated access and mapping to a user role.
FIA_UID	User identification	User identification before action, requires that users identify themselves before any other action will be allowed by the TSF.	HSM can require user identification, either to itself or its host, prior to it providing any services.
FMT_SMR.1	Security management	Security roles specifies the roles with respect to security that the TSF recognizes.	Privileged operations can require HSM users to be mapped to security roles, unless the host is designated a single role.
FMT_SMR.2	Security management	Restrictions on security roles specifies that in addition to the specification of the roles, there are rules that control the relationship between the roles.	Assume roles such as crypto officer that support creation of user processes, e.g. applications, and mapping to roles.
FMT_SMR.3	Security management	Assuming roles, requires that an explicit request is given to the TSF to assume a role.	Add ability for HSM to require authenticated access to any role.
FMT_REV.1	Revocation	Revocation provides for revocation of security attributes to be enforced at some point in time.	Add ability for HSM to validate certificates, including revocation status. Some designs can 1) allow for signature verification in the HSM and/or 2) allow the HSM to check the public key and its trust chain as part of certificate validation.
FMT_MSA.1	Management of security attributes	Management of security attributes allows authorized users (roles) to manage the specified security attributes.	HSMs should include security-specific user roles.
FMT_MTD.1	Management of TSF data	Management of TSF data allows authorized users to manage TSF data.	Assume roles such as crypto officer that support creation of user processes, e.g. applications, and how their metadata maps to roles.
FMT_MTD.3	Management of TSF data	Secure TSF data ensures that values assigned to TSF data are valid with respect to the secure state.	HSM should monitor its own security state.
FDP_DAU.2	Data authentication	Data Authentication with Identity of Guarantor additionally requires that the TSF is capable of establishing the identity of the subject who provided the guarantee of authenticity.	HSM should be capable of establishing the identity of the root of trust that issued the certificate used for authentication, for example.

Table 8 *(continued)*

SFR	SFR family	SFR description	Gap
FDP_ITT.3	Internal TOE transfer	Integrity monitoring, requires that the TSF monitor user data transmitted between parts of the TOE for identified integrity errors.	Data processed within the HSM should be monitored for integrity anomalies.
FCS_CKM.3	Cryptographic key management	Cryptographic key access, requires access to cryptographic keys to be performed in accordance with a specified access method which can be based on an assigned standard.	HSMs limit access to cryptographic keys.
FIA_UAS.6	User authentication	Re-authenticating, requires the ability to specify events for which the user needs to be re-authenticated	HSM should identify and enforce re-authentication as required.
FIA_UAU.7	User authentication	Protected authentication feedback, requires that only limited feedback information is provided to the user during the authentication	HSM should limit feedback on failed authentications; can be augmented/supported by host.
FPT_ITT.1	Internal TOE TSF data transfer	Basic internal TSF data transfer protection, requires that the TSF data be protected when transmitted between separate parts of the TOE.	Add ability for TSF data to be integrity protected while in transit internal to the HSM or across the HSM / host boundary.
FAU_ARP.1	Security audit automated response	Security alarms, the TSF takes action if a potential security violation is detected.	Assume host level support, however an HSM should be capable of taking action (e.g. zeroization) based on detection of a security-critical event.
FAU_SAA.1	Security audit analysis	Potential violation analysis, basic threshold detection on the basis of a fixed rule set is required.	HSM should be capable of identifying basic attacks against itself.
FAU_STG.1	Security audit event storage	Protected audit trail storage, requirements are placed on the audit trail. It will be protected from unauthorized deletion and/or modification.	HSM is expected to provide protected audit storage.

6.7.4 Gap analysis against V-ITS base PP

The V-ITS Base PP[30] is a modular PP that can be utilized on its own or together with other PPs. The Base module borrows its cryptographic security requirements from the C2C HSM PP[29]. Because of this, we have excluded cryptographic and key management SFRs from this gap analysis due to the expected integration of the C2C HSM PP for any system being developed in conformance with this V-ITS Base PP. Gaps in the C2C HSM PP are already noted earlier. Disregarding the cryptographic and key management requirements, a number of high-level requirements were still identified as gaps. These include:

- identity and access management SFRs supporting the identification of a user prior to authenticating to the HSM;
- role-based security management requirements;
- revocation checking requirements that can be handled by the host (base) vs. the HSM;
- data authentication requirements;
- integrity monitoring requirements;
- host-level attack detection and response requirements;
- secure boot requirements;

— stored data security requirements;

— audit event data security requirements;

— session security requirements (this is user interface-related sessions).

Table 8 identifies the specific items identified as gaps within the V-ITS Base PP.

Table 9 — Possible gaps in V-ITS Base PP

SFR	SFR family	SFR description	Gap
FIA_ATD.1	User attribute definition	User attribute definition, allows user security attributes for each user to be maintained individually.	Assumed that unique use attributes are necessary to distinguish unique users' accesses to a generic IDX device.
FIA_USB.1	User-subject binding	User-subject binding, requires the specification of any rules governing the association between user attributes and the subject attributes into which they are mapped.	Assumed that unique use attributes are necessary to distinguish unique users' accesses to a generic IDX device.
FIA_UID.2	User identification	User identification before action, requires that users identify themselves before any other action will be allowed by the TSF.	A generic IDX device conservatively should not perform security functions on behalf of a user prior to that user identifying him/herself.
FMT_REV.1	Revocation	Revocation, provides for revocation of security attributes to be enforced at some point in time.	The check on the revocation of a user's rights should be performed on a periodic basis because it is assumed that some user sessions in the scenarios can be long-lived.
FMT_MTD.1	Management of TSF data	Management of TSF data, allows authorized users to manage TSF data.	For the IDX-device, it is assumed that some TSF data will require secure default accesses and that only some role(s) will be able to modify (the data and data accesses).
FMT_MTD.2	Management of TSF data	Management of limits on TSF data, specifies the action to be taken if limits on TSF data are reached or exceeded.	For the IDX-device, it is assumed different settings can be needed to handle excesses of audit data. Rolling memory can be sufficient in some cases; in others, an indicator that TSF data has hit thresholds.
FMT_MTD.3	Management of TSF data	Secure TSF data, ensures that values assigned to TSF data are valid with respect to the secure state.	The assigned values should be such that the device will remain in a secure state. A complete STF will specify secure values. This is needed for threats against the access control policy itself.
FDP_DAU.1	Data authentication	Basic data authentication, requires that the TSF is capable of generating a guarantee of authenticity of the information content of objects (e.g. documents).	In Scenario 2, it is assumed that the IDX Device can be an information broker that needs to authenticate data, documents, files, etc. between the SCN and other entities, including regulatory authorities. One example is an emissions test report needed for regulatory reporting. Scenario 3 can also require this.

Table 9 *(continued)*

SFR	SFR family	SFR description	Gap
FDP_DAU.2	Data authentication	Data authentication with identity of guarantor, additionally requires that the TSF is capable of establishing the identity of the subject who provided the guarantee of authenticity.	In Scenario 2, it is assumed that the IDX device can be an information broker that needs to authenticate data, documents, files, etc. between the SCN and other entities. An example is an emissions test report. Scenario 3 can also require this capability in the process of uploading sensitive data (e.g. firmware) into a SCN.
FDP_ITC.1	Import from outside of the TOE	Import of user data without security attributes, requires that the security attributes correctly represent the user data and are supplied separately from the object.	Scenario 2 authentication of data/reports likely requires security attributes either directly or indirectly associated with the data (by the TSF). Scenario 3 FW management protocols can require security attributes to be associated with the data.
FDP_ITC.2	Import from outside of the TOE	Import of user data with security attributes, requires that security attributes correctly represent the user data and are accurately and unambiguously associated with the user data imported from outside the TOE.	Scenario 2 authenticated data/reports likely require security attributes either directly or indirectly associated with the data (by the TSF). Scenario 3 FW management protocols can require security attributes to be associated with the data.
FTP_ITT.3	Internal TOE transfer	Integrity monitoring, requires that the TSF monitor user data transmitted between parts of the TOE for identified integrity errors.	This is assumed to be needed for the TSF to copy, compile or otherwise prepare regulatory-type reports concerning public data retrieved from CSNs. This can be needed before generating a report according to a particular scheme, or as part of the process of verifying data integrity before copying into the new schema.
FDP_SDI.1	Stored data integrity	Stored data integrity monitoring, requires that the TSF monitor user data stored within containers controlled by the TSF for identified integrity errors.	Existing PP not finalized which subsections apply.
FDP_SDI.2	Stored data integrity	Stored data integrity monitoring and action adds the additional capability to the first component (FDP_SDI.1) by allowing actions to be taken as a result of error detection.	Existing PP not finalized which subsections apply.
FIA_ATD.1	User attribute definition	User attribute definition, allows user security attributes for each user to be maintained individually	This assumes a granular, identity-based account management feature is present.
TBT.1	Secure boot and operational continuity	For secure boot: a) the TOE is able to obtain the proper KEK for the disk encryption AND b) the TOE verifies the integrity of the OS.	Needed at minimum for powerup integrity tests on all executables and crypto functions within the HSM as well as functions within the host.

Table 9 *(continued)*

SFR	SFR family	SFR description	Gap
FDP_DSK.1	Protection of stored data	The TSF performs encryption of (assignment: data) in accordance with [assignment: cryptographic algorithm], such that no such data is otherwise stored as plain text within the TOE.	Base level PP should incorporate data security protections for stored data.
FDP_DSK.2	Protection of stored data	The DEK is encrypted with a KEK. The DEK only exists in persistent memory on the disk.	Base level PP should incorporate data security protections for stored data.
FDP_DSK.3	Protection of stored data	The TSF encrypts all data without user intervention.	Base level PP should incorporate data security protections for stored data.
FDP_DSK.4	Protection of stored data	The TSF supports decryption of the protected data once the corresponding KEK is presented.	Base level PP should incorporate data security protections for stored data.
FAU_SAA.1	Security audit automatic response	The TSF carries out potential violation analysis, using at least basic threshold detection on the basis of a fixed rule set.	Should be capable of detecting basic attacks/malware.
FAU_STG.1	Security audit event storage	Protected audit trail storage, requirements are placed on the audit trail. It is protected from unauthorized deletion and/or modification.	TOE should be capable of protecting the integrity of audit data.
FAU_STG.2	Security audit event storage	Prevention of audit data loss, specifies actions in case the audit trail is full.	TOE should be capable of protecting the integrity of audit data.
FTA_TSE.1	TOE session	TOE session establishment includes applying an access control policy based on attributes.	TOE should require authentication/authorization based on user attributes for Scenarios 2 and 3. Session establishment is part of the authorization function to a resource in the model of ISO/TS 21177 and checking user attributes such as permissions indicated in a client/server credential supports this authorization function.
FTA_SSL.1	Session locking and termination	TSF-initiated session locking, includes system-initiated locking of an interactive session after a specified period of user inactivity.	TOE should include session locking capabilities for Scenarios 2 and 3.
FTA_SSL.2	Session locking and termination	User-initiated locking, provides capabilities for the user to lock and unlock the user's own interactive sessions.	TOE should include user-initiated session locking capabilities for scenarios 2 and 3.
FTA_SSL.3	Session locking and termination	TSF-initiated termination, provides requirements for the TSF to terminate the session after a specified period of user inactivity.	TOE should include inactivity-based sessions locking capabilities for Scenario 2 and 3.
FTA_SSL.4	Session locking and termination	User-initiated termination, provides capabilities for the user to terminate the user's own interactive sessions.	TOE should include user-initiated session locking capabilities for Scenarios 2 and 3.

6.7.5 Gap analysis against V-ITS Comms Module PP

The Comms Module PP was reviewed for gaps with an understanding that the majority of SFRs would have been allocated to the V-ITS-S Base PP[30]. Minimal gaps are presented in Table 10 relating to:

- rollback of firmware;
- physical protection against tampering /unauthorized access to test ports.

Table 10 — Possible gaps in V-ITS Comms Module PP

SFR	SFR family	SFR description	Gap
FDP_REV.1	Rollback	Basic rollback supports rolling back or undoing a limited number of operations within the defined bounds.	Add support for rollback control if independent firmware is used within the implemented module.
FPT_PHP	Physical protection	Resistance to physical attack, provides for features that prevent or resist physical tampering with TSF devices and TSF elements.	An IDX device should protect against unauthorized access to test ports or other physical ports on device.

7 ISO/TS 21177 access control implementation guidance

7.1 General

ITS-SUs are rapidly maturing with regard to specification, use and security conformance standards. In support of the ITS-S ecosystem are standards such as IEEE 1609.2 and ISO/TS 21177 which provide a framework for device-to-device secure sessions and resource access authorization. However, these standards provide only a framework for secure communications: they define interfaces and activities across those interfaces but leave a large amount of detail to be specified on an application-by-application basis. Additionally, early experiences in integrating ISO/TS 21177 have indicated that some extensions to the functionality in that standard would be useful for supporting additional authentication use cases.

Clause 7 of this document is provided to assist device and application designers in implementing an ISO/TS 21177-based access control implementation. This is accomplished by illustrating through automotive access control use cases the steps in which a maintenance and diagnostics device and vehicle can securely connect to provide ISO/TS 21177 controlled access.

This section is divided into the following subsections:

- **High level architecture and access scenario** - A description is provided for an architecture in which an example application based on the Unified Gateway Protocol provides communications between an automotive diagnostics device and vehicle. Additionally, a high-level application scenario is provided as a backdrop to illustrating access control use cases.
- **Application protocol architecture and ISO/TS 21177 integration** – This section depicts an approach for architecturally integrating the UGP communications protocol with ISO/TS 21177. This is only provided as an example to implementers of how to approach integrating any application-layer protocol with the ISO/TS 21177 framework.
- **Access control policy structure** – This section provides an example access control policy data structure in ASN.1 that is used in the example access control use cases.
- **Access control approach** – This section depicts the client and server roles as well as the structures and usage of their access control permissions within the access control use cases.
- **Access control use cases and sequence diagrams** – This section provides essential access control use cases that exemplify how each of the example access control structures are used in the context of the ISO/TS 21177 framework. Each depicts the relevant access control checks that are performed against policy, and the specific flow and ISO/TS 21177 sequence of operations for each.

7.2 High level architecture and access scenario

This subclause describes an example UGP brokering application referred to as 'UGPApp'. This application and its architecture are provided to highlight a method of integration with the ISO/TS 21177 secure access control model.

UGP is a client-server protocol wherein a Vehicle ITS-Station Gateway (V-ITS-SG) 'server' provides gateway services to a nomadic device (ND) UGP 'client.' UGP provides baseline service-based access methods used by NDs or other application instances that need to access from the UGP server vehicle data.

The basic architecture is indicated in Figure 12.

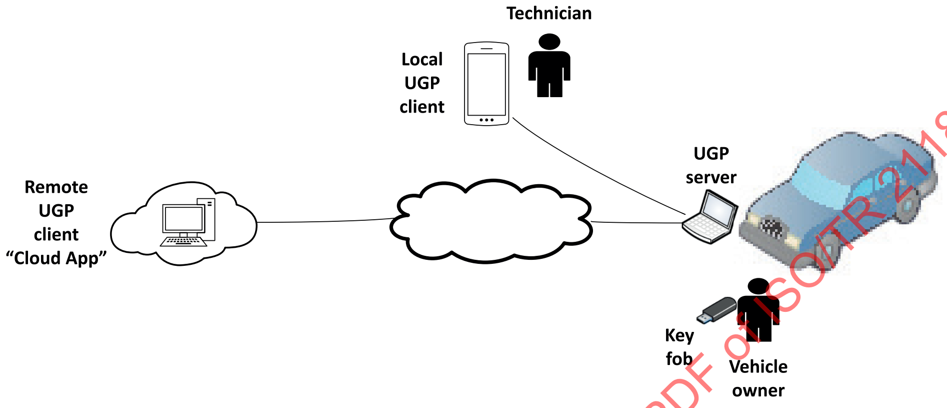

Figure 12 — UGP-based Vehicle Access Scenario

A typical UGP application scenario involves a technician holding a tablet or other ND executing a 'Local UGP client'. The UGP client is assigned a client 'role' that is indicated in its IEEE 1609.2 certificate SSP.

The UGP server, exemplified by UGPApp, is the vehicle's server-side broker from which the client requests and obtains access to vehicular data. Once the ND UGP client has securely connected to the UGPApp, UGPApp caches the ND's client role based on the SSP found in the certificate. The client role is used subsequently in access control lookups to determine if the technician's ND is authorized to access specific UGP services and resources. The UGPApp receives these requests, performs a lookup of the role-specific access control policy and determines:

1) Whether the role is permitted or denied access to the requested UGP service.

2) If service access is permitted, whether the role is permitted access to the one or more indicated resources accessed by the UGP service.

If the answer to both access control requests is 'permit', UGPApp invokes the requested UGP service actions on the vehicular resource(s) (e.g. GetValue, etc.). If access control is denied, based on role, UGPApp can either a) not respond, or b) return an ISO/TS 21177 access control error to the ND.

The high-level process is indicated in Figure 13.

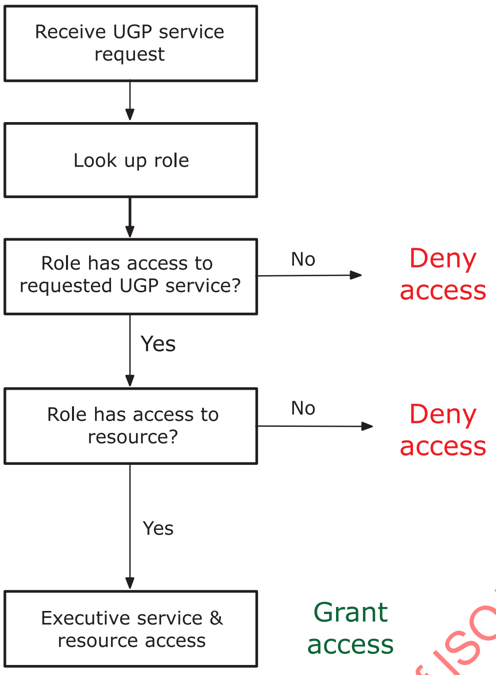
Receive UGP service request
Look up role
Role has access to requested UGP service?
No
Deny access
Yes
Role has access to resource?
No
Deny access
Yes
Executive service & resource access
Grant access

Figure 13 — High-level service/resource access control process

Note that some UGP services return a collection of resources, each with its own access policy. Therefore, a permitted UGP service can return a subset of the requested resources, based on the role indicated in the certificate SSP and the read, write, execute and other access types allocated to the role in the access control policy.

7.3 Application protocol architecture and ISO/TS 21177 integration

7.3.1 General

The example application protocol is UGP. UGP is specified in ISO 13185-2 and provides data request/response services to access vehicle information resources. A UGP endpoint can be a 'Server' located on the vehicle, a client as in the case of a diagnostics data collection device, a cloud-based application or some other machine endpoint able to generate or respond to UGP requests.

This subclause first describes at a high level the UGP architecture and protocol services, then provides the example integration of the protocol with ISO/TS 21177.

7.3.2 Example protocol architecture

The high-level UGP architecture is indicated in Figure 14.

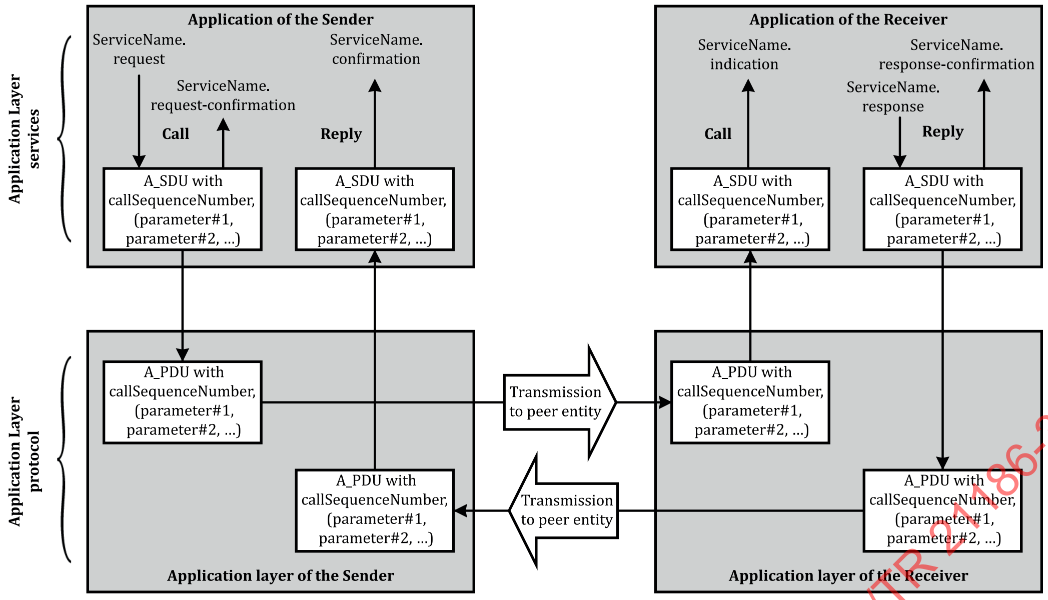
Application Layer services
Application Layer protocol
Application of the Sender
ServiceName. request
ServiceName. confirmation
ServiceName. request-confirmation
Call
Reply
A_SDU with callSequenceNumber, (parameter#1, parameter#2, ...)
A_SDU with callSequenceNumber, (parameter#1, parameter#2, ...)
Application of the Receiver
ServiceName. indication
ServiceName. response-confirmation
ServiceName. response
Call
Reply
A_SDU with callSequenceNumber, (parameter#1, parameter#2, ...)
A_SDU with callSequenceNumber, (parameter#1, parameter#2, ...)
A_PDU with callSequenceNumber, (parameter#1, parameter#2, ...)
Transmission to peer entity
A_PDU with callSequenceNumber, (parameter#1, parameter#2, ...)
A_PDU with callSequenceNumber, (parameter#1, parameter#2, ...)
Transmission to peer entity
A_PDU with callSequenceNumber, (parameter#1, parameter#2, ...)
Application layer of the Sender
Application layer of the Receiver

Figure 14 — UGP application architecture (see ISO 13185-2)

UGP architecturally separates the 'Application' from the 'Application Layer', providing request, request-confirmation, and confirmation primitives which the application uses in request/response messaging. Request/response messages over the wire are wrapped in a header generated by the application layer.

A UGP client can invoke the following services from a UGP Server:

- GetValue: Obtains a data value from the V-ITS-SG
- SetValue: Sets a data value to the V-ITS-SG
- ControlValue: Invokes a control function on the V-ITS-SG
- Getdtcinfo: Obtains diagnostic trouble code data from the V-ITS-SG
- Cleardtcinfo: Clears diagnostic trouble code data on the V-ITS-SG
- EnablePassThrough: Provides a bypass capability to enable an organization to bypass normal access controls (e.g. as needed by a manufacturer) on the V-ITS-SG
- ListFile: Used to request a download of core, configuration, or log files from the V-ITS-SG
- ManageFile (upload): Uploads a file to the V-ITS-SG
- ManageFile (download): Downloads a file from the V-ITS-SG
- ManageFile (delete): Deletes a file on the V-ITS-SG
- ResetAccess: Used to reboot or to shut down the V-ITS-SG

The interplay between UGP (a data brokering application) and ISO/TS 21177 (a security broker framework) necessitates integration and proper use of each protocol's functional units. This section documents an integration strategy enabling the two protocols to layer correctly and provide enhanced access control and secure connectivity to the UGP application.

7.3.3 Protocol integration strategy

This subclause provides an implementation strategy for integrating the ISO/TS 21177 functional elements with an example UGP-architected application. ISO/TS 21177 stipulates the following functional elements be added to the UGP application:

- ISO/TS 21177 Security Subsystem
- ISO/TS 21177 Adaptor Layer
- ISO/TS 21177 Secure Session
- An access control policy interpretable by the ISO/TS 21177 Security Subsystem

Figure 15 provides a pictorial view of the UGP-and-21177 integrated architecture.

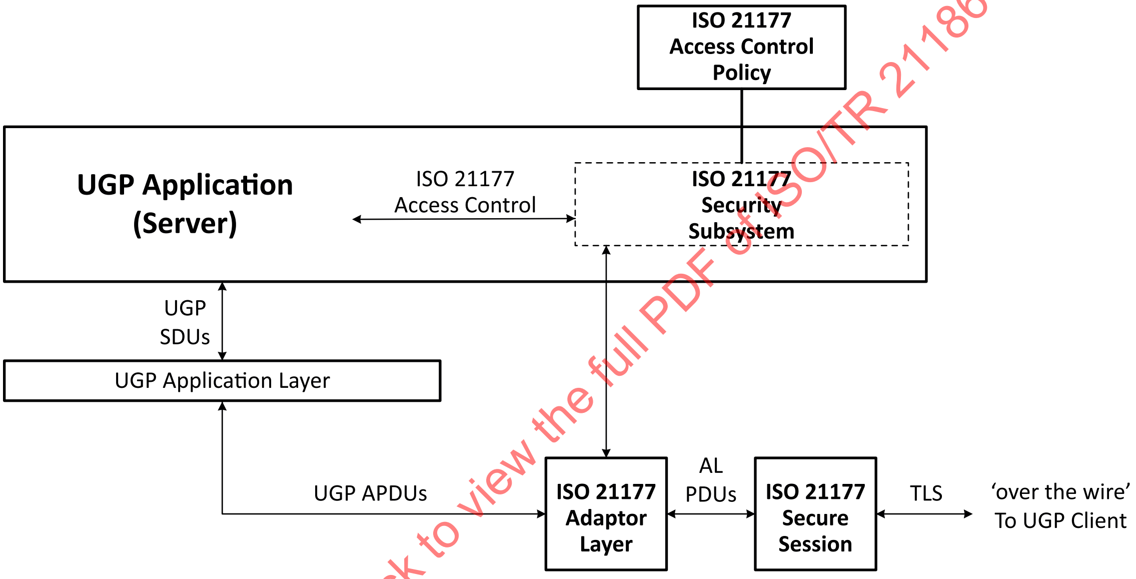

Figure 15 — Integrated architecture between ISO/TS 21177 and UGP

The integration strategy is as follows:

1) **Front the ISO 13185 Application Layer with the ISO/TS 21177 Adaptor Layer**. This implies that the 'over-the-wire' UGP PDUs are no longer those of the UGP Application Layer, but instead the ISO/TS 21177 Adaptor Layer. The ISO/TS 21177 Adaptor Layer header therefore wraps the inbound/outbound UGP Application Layer A_PDU with an ISO/TS 21177 ALPDU before sending it. Likewise, when the ISO/TS 21177 Adaptor Layer receives a data ALPDU from the UGP client, it decapsulates it to a UGP A_PDU and passes it to the UGP Application Layer. The Adaptor Layer, if it needs to send or receive access control messages with the UGP client application, uses an ISO/TS 21177 access control header that causes the PDU to be routed directly to the opposing application's Security Subsystem.

2) **Front the ISO/TS 21177 Adaptor Layer with the ISO/TS 21177 secure session, as per ISO/TS 21177 architecture**. This enhancement adds ISO/TS 21177-conformant TLS to all outbound/inbound APDUs in order to provide confidentiality, integrity and data origin authentication to all messages sent between the UGP Client and UGP Server.

3) **Add the ISO/TS 21177 conformant security subsystem to the UGP Application**. The UGP Application makes ISO/TS 21177 access control calls to the security subsystem and receive responses, according to ISO/TS 21177. The security subsystem can retain some access control provisions already defined in ISO 13185-2, but implements the ISO/TS 21177 primitives used for internal component messaging or communicating with the peer application's security subsystem.

7.4 Access control policy structure

This subclause provides a close representation of the access control policy ASN.1 structure from ISO/TS 21184:—[1], A.3. The ASN.1 structure provided in this subclause is intended only to illustrate for the UGPApp a role-based access control policy file called "V_ITS_G-ACCESS-CONTROL-POLICY". In conjunction with the ASN.1 extension structure, called TlsPolicyConfig (included here and fully defined in 7.6.4) UGPApp could be securely configured both with data access control and TLS Secure Session policies.

Many possible access control file structures are possible in ITS application deployments. Each should be separated by an application identifier and vendor-unique policies should maintain a minimum, core set of policies that support interoperability goals of the application.

The ASN.1 included in this subclause and subsequent sections is not intended for compilation and use; it is only intended to illustrate example structures for implementing a resource access control framework in an ITS-SU.

```
Identifier ::= SNUM32 -- imported from ISO/TS 21184 Annex A.1
Version ::= INTEGER(0..255) -- imported from ISO/TS 21184 Annex A.1
AccessType::= BIT STRING {
    r   (0),
    w   (1),
    x   (2),
    i   (3),
    u   (4)
} (SIZE(5, ...)) -- imported from ISO/TS 21184 Annex A.1

acVersion Version ::= 1

AcConfig ::= SEQUENCE {
    configFile             VisibleString("V_ITS_G-ACCESS-CONTROL-POLICY"),
    configName             VisibleString,
    configVersion          VisibleString,
    role                   SEQUENCE OF ACRole, -- see ISO/TS 21184 Appendix A.3
    entrySet               SEQUENCE OF ACEntrySet,
    tlsPolicyConfig        TlsPolicyConfig, -- Extended from ISO/TS 21184
    ...
}

ACRole ::= SEQUENCE {
    id                    Identifier,
    name                  VisibleStrin,
    serviceMask           ACService,
    entrySetId            Identifier,
    ...
}

ACService ::=  BIT STRING {
    getsupported        (0),
    getvalue          (1),
    setvalue          (2),
    controlvalue         (3),
    getdtcinfo        (4),
    cleardtcinfo         (5),
    enablepassthrough   (6),
    listfile          (7),
    managefile-download (8),
    managefile-upload   (9),
    managefile-delete   (10),
    reset               (11),
    sendMsg             (12),
} (SIZE(14, ...))

ACEntrySet ::= SEQUENCE {
    id      Identifier,
    entry   SEQUENCE OF ACEntry,
```

[1] Under preparation. Stage at the time of publication: ISO/PRF TS 21148:2021.

```
    ...
}
ACEntry ::= SEQUENCE {
    ecuId                  Identifier,
    dpId                   Identifier,
    accessType             AccessType,
    ...
}
```

7.5 Access control approach

This subclause describes the example access control approach for the sample ISO/TS 21177 and UGP integrated server application 'UGPApp.' The access control approach is exemplified in the use cases provided in 7.6.

In practice, the permissions content of the certificates can be defined by a variety of different structures. A common structure defined in IEEE 1609.2 is the BitmapSsp which holds a simple bit field in which roles and permissions can be encoded and appropriately described in an application specification. The BitmapSsp is the structure used herein to elaborate the permissions of the UGP client ND.

The access control model described here for UGPApp consists of a simple role-based access control mechanism. UGPApp functions in the server role for TLS and UGP, and in that role services requests by a party acting in the client role.

UGP clients are intended to initiate TLS connections to the UGPApp server to start ISO/TS 21177 sessions.

UGPApp's IEEE 1609.2 server certificate contains the following parameters:

- ITS-AID = 0x10203c7f
- A COER-encoded SSP, containing the following information:
  - endpoint type = 'server'

The ND UGP client initiating communication checks during the Secure Session TLS handshake with UGPApp, that UGPApp's 1609.2 server certificate contains a PsidSsp structure consisting of a known and allowed ITS-AID, endpoint type and protocol. If not, the ND abandons the connection.

The UGPApp 'server' endpoint type is indicated by an entry in UGPApp's IEEE 1609.2 certificate, called appPermissions. This field contains a 1609.2 sequence of PsidSsp, each a tuple of (ITS-AID, Sequence of SSPs for the ITS-AID). UGP clients of the UGPApp server inspects this field to ensure that there is at least one PsidSsp entry of the correct ITS-AID and COER-encoded SSP, where the SSP definition indicating a 'serve' role for the endpointType is as follows:

```
UgpServerSsp ::= BitmapSsp   --from 1609.2
```

The first octet of UgpServerSsp is the SSP version (e.g. 0x01).

The second octet of UgpServerSsp is the indication of the endpoint type:

- 0x00 is 'server'
- 0x01 is 'client'
- The third and fourth octets are used for client role types and are absent in the server's certificate.

Therefore, the example UGPApp server's SSP is: 0x0100

UGP client roles: The ND UGP client can be one of several roles defined for UGPApp's application ITS-AID; see Table 11.

Table 11 — Access control client roles and identifiers

Client role	ID
DefaultRole	0
ConfigRole	1
VehDiagRole	2
VehSecurityDiagRole	3
VehRemoteDiagRole	4
VehDriverProfileRole	5
VehInsuranceRole	6
VehRoadsideAssistRole	7
VehInspMaintRole	8
PAPPolicyManagerRole	9

UGPApp can be configured, by policy, to permit only a subset of client roles, or all roles, to connect and access services in the use cases described in the next section. Use cases presented in 7.6 invoke the following client roles.

- ConfigRole: This role is authorized to upload a file to the vehicle's UGPApp server.
- VehDiagRole: This role is authorized to access certain UGP-provisioned data resources.
- PAPPolicyManagerRole: This role is authorized to create an access control policy and activate it by generating a 1609.2 signature over the access control policy file.

Client role is indicated in the UGPApp client's IEEE 1609.2 certificate appPermissions. The UGPApp server, in this case, is designed to accept only one role for each client certificate, therefore each client 1609.2 certificate contains only a single PsidSsp entry consisting of the following parameters:

- ITS-AID 0x10203c7f
- a COER-encoded SSP with an endpointType='client' and one of the client roles defined in the access control policy.

The UGP Client SSP structure is defined as follows:

```
UgpClientSsp ::= BitmapSsp   --from 1609.2
```

The first octet is the SSP version (e.g. 0x01).

The second octet is the endpoint type:

- 0x00 is 'server'
- 0x01 is 'client'

The third and fourth octets indicate the allowed role(s) to which the client certificate is entitled.

Octet 3:

- Bit 0: '1' indicates the certificate can act in the ConfigRole
- Bit 1: '1' indicates the certificate can act in the VehDiagRole
- Bit 2: '1' indicates the certificate can act in the VehSecurityDiagRole
- Bit 3: '1' indicates the certificate can act in the VehRemoteDiagRole
- Bit 4: '1' indicates the certificate can act in the VehDriverProfileRole

- Bit 5: '1' indicates the certificate can act in the VehInsuranceRole
- Bit 6: '1' indicates the certificate can act in the VehRoadsideAssistRole
- Bit 7: '1' indicates the certificate can act in the VehInspMaintRole

Octet 4:

- Bit 0: '1' indicates the certificate can act in the PAPPolicyManagerRole
- Bits 1-7: Reserved – Set to zero

If octets 3 and 4 are all zeroes, the client assumes the role DefaultRole.

An example client SSP for an entity authorized to act in the role ConfigRole is: 0x01010100. This indicates version 1 of the SSP, entity type of 'client', and client role = ConfigRole.

Both the UgpServerSsp and UgpClientSsp are subtypes of AutomotiveSsp, defined as follows:

```
AutomotiveSsp ::= SEQUENCE {
   version              Uint8,
   protocol             ApplicationProtocol, --application-specific
   endpointType         EndpointType,
   role                    SubjectRole OPTIONAL,
    ...
}
ApplicationProtocol ::= SEQUENCE {
   protocolVersion      Uint8,
   protocolRef          ProtocolReference
}
ProtocolReference ::= CHOICE {
   ugp                     INTEGER(0),-- ISO13185
   uds                     INTEGER(1),-- ISO14229
   dxm                     INTEGER(2)
}
EndpointType ::= CHOICE {  -- Specific to this application ITS-AID
   client               INTEGER(0),
   server               INTEGER(1),
   ...
}
SubjectRole ::= INTEGER {
   defaultRole         (0),
   configRole          (1),
   vehDiagRole         (2),
   vehSecurityDiagRole       (3),
   vehRemoteDiagRole         (4),
   vehDriverProfileRole      (5),
   vehInsuranceRole          (6),
   vehRoadsideAssistRole     (7),
   vehInspMaintRole          (8),
   papPolicyManagerRole      (9)
}
```

The root SSP type AutomotiveSsp is COER-encoded and inserted into the single 1609.2 PsidSsp SSP field. Within the AutomotiveSsp type:

- version indicates the version of the SSP definition;
- protocol indicates the protocol (ugp, uds or dxm) and version of the protocol that the certificate holder is allowed to communicate;
- endpointType indicates whether the certificate holder is client or server. Servers are expected to check, by policy, that a client SSP indicates endpointType=client;
- role indicates the Role of the certificate holder. The role is used in all subsequent access control decisions enforced by the Server. Note that the role definitions applied to the client certificates overlaps the role definitions in the access control policy file, described later. If the endpointType=client, then the role entry is present. If the endpointType=server, then the role entry is absent.

7.6 Access control use cases and sequence diagrams

7.6.1 General

This subclause provides a sequence of access control use cases for each of the flows and related messaging structures used in accordance with ISO/TS 21177.

The following configuration use cases are provided:

- Define an access control policy
- Load an access control policy
- Configure TLS
- Start a secure TLS session
- Secure access controlled-resource discovery
- Server controls access to UGP service based on role

7.6.2 Define an access policy

Objective

The objective of this use case is to establish and secure an UGP server access control policy unique to each role. The policies will enable the server to enforce role-based access to specific UGPApp-brokered resources.

Description

The use case access controls are based on the role-based service access control scheme indicated in Table 12.

Table 12 — Role-based access control to UGP services

Service/User role	Default Role (ID:0)	Config Role (ID:1)	VehDiag Role (ID:2)	VehSecruri-tyDiagRole (ID:3)	VehRemote DiagRole (ID:5)	VehDriver ProfileRole (ID:6)	VehINsur-anceRole (ID:7)	VehRoadside-AssistRole (ID:8)	VehINspMain-tRole (ID:9)	PAPPolicy-ManagerRole (ID:10)
tls-connect	y	y	y	y	y	y	y	y	y	y
ugp-getsupported	y	y	y	y	y	y	y	y	y	n
ugp-getvalue	y	n	y	y	n	y	y	y	y	n
ugp-setvalue	n	n	y	y	n	y	y	y	n	n
ugp-controlvalue-access	n	n	n	y	y	n	n	y	n	n
ugp-getdtcinfo	n	n	y	y	y	n	n	y	Y	n
ugp-cleardtcinfo	n	n	y	y	y	n	n	y	n	n
ugp-enablepassthru	n	n	n	y	n	n	n	n	n	n
ugp-listfile	n	y	y	y	n	n	n	n	n	y
ugp-managefile-download	n	y	n	n	n	n	n	n	n	y
gup-managefile-upload	n	y	n	n	n	n	n	n	n	y
ugp-managefile-delete	n	y	n	n	n	n	n	n	n	y
ugp-reset	n	y	n	n	n	n	n	n	n	n
sendDXM	n	n	n	n	n	n	n	n	n	n
activate-policy	n	n	n	n	n	n	n	n	n	y

TLS connection access: The three roles that are allowed to connect to UGPApp's secure session are the three for whom the TLS-connect service is enabled in the access policy.

UGP service access: The roles that are permitted to TLS-connect are also given permissions to UGP services, those that are prefixed with 'ugp-'.

Policy activation access: The one role that is allowed to both connect and activate an access control policy is the PAPPolicyManager role.

Resource access: The low-level resource access controls are defined in substructures of the policy file, populated as described in this section.

Actors

— Policy Administration Point Policy Manager (PAPPolicyManager)

Prerequisites

— The PAPPolicyManager possesses a policy editing utility for creating, editing and digitally signing, as per IEEE 1609.2

Flow

The process steps for this use case are as follows:

The PAPPolicyManager:

a) generates an access control policy file V_ITS_G-ACCESS-CONTROL-POLICY by performing the following:

1) Populates an AcConfig structure within the file, defined as:

```
AcConfig ::= SEQUENCE {
    fileName              VisibleString("V_ITS_G-ACCESS-CONTROL-POLICY"),
    fileVersion             Uint8,
    configName              VisibleString,
    configVersion           Uint8,
    configDescription       VisibleString,
    roles                   SEQUENCE OF ACRole,
    entrySet                SEQUENCE OF ACEntrySet,
    secureSessionConfig     TlsPolicyConfig OPTIONAL,
    provide21177Errors      BOOLEAN, -- if TRUE, Security Subsystem sends access
control status to requestor
    ...
}
```

i) fileName is "V_ITS_G-ACCESS-CONTROL-POLICY". This filename is used by UgpApp as a flag to recognize that the configuration file is:

I) a sensitive access control configuration

II) a file signed by the PAPPolicyManager role, the single role that is authorized to sign access control files.

ii) fileVersion, configName, configVersion, configDescription are populated, as appropriate.

iii) roles provide the listing of ACroles indicated in Table X. For each ACrole:

I) id is the role identifier, matching those in Table X.

II) name is the name listed for each role.

III) serviceMask provides the BIT STRING of 0s and 1s indicating the services accessible to the role (0=denied, 1=permitted).

IV) entrySetId is an ACEntry, each containing a sequence of the role's access control entries, one for each resource that is accessible to the role when it invokes one of its allowed services. Called ACEntry, each indicates:

V) — ecuID: ECU identifier

— rvID: UGP registered value identifier

— accessType: The role's access type for the indicated resource. This can be one or more of the values read (r), write (w), execute (x), accompanied by one of the values internal (i) or user (u).

iv) entrySet provides the listing of all resources available, a superset of those potentially accessible by any one role. entrySet provides a sequence of one or more ACEntrySets. Each ACEntrySet contains:

I) an ID

II) an entry, containing a sequence of ACEntry

v) secureSessionConfig provides the role-based TLS access control policy for the UGPApp's ISO/TS 21177 secure session. This structure is comprised of a set of TLS connection policies, each with a unique identifier and a set of SSPs that are allowed to connect. Only one policy can be 'active' at a time, as selected by UGPApp. More information about how to configure the secureSessionConfig TLS policy structure is provided in the use case **Configure TLS**.

vi) provide21177AcErrors is a Boolean type. If it is TRUE, then the ISO/TS 21177-conformant access control behaviour of the Security Subsystem will send access control status and feedback in the form of an Iso21177AccessControlPdu of type accessControlResultId. This value, if TRUE, will only send access control errors, i.e. a generic indication that a request for a specific resource or service is not authorized. If this value is FALSE, the UGPApp server will not send application-specific access control feedback using the ISO/TS 21177 method, but can continue to use existing UGP protocol behaviour.

The PAPPolicyManager:

a) as per the 1609.2 permissions listed in its 1609.2 signing certificate, generates an IEEE 1609.22 signature over the completed ACConfig structure, resulting in the following 'signed' Ieee1609Dot2Data data structure:

```
SignedAcConfig ::= Ieee1609Dot2Data (WITH COMPONENTS {...,
  content (WITH COMPONENTS {...,
    signedData  (WITH COMPONENTS {...,
      tbsData (WITH COMPONENTS {...,
        payload (WITH COMPONENTS {...,
          data (WITH COMPONENTS {...,
             content (WITH COMPONENTS {
                unsecuredData (CONTAINING AcConfig)
            })
          })
        }),
        headerInfo (WITH COMPONENTS {...,
          generationTime PRESENT,
          expiryTime ABSENT,
          generationLocation ABSENT,
          p2pcdLearningRequest ABSENT,
          missingCrlIdentifier ABSENT,
          encryptionKey ABSENT,
          flags ABSENT
        })
      })
    })
  })
})
```

The PAPPolicyManager:

a) stores the 1609.2-signed structure in a file

b) provides the file with a filename identical to that given in the top-level AcConfig structure.

7.6.3 Load an access control policy

Objective

The objective of this use case is to upload and activate the access control policies to UGPApp.

Description

This use case can initiate after the Access Control Policy has been generated and signed. Once prepared, either the PAPPolicyManager role or Config role establishes a secure session with the UGPApp server, as described in use case '**Start a secure TLS session**', and then performs a UGP protocol file upload of the access control policy file.

Once the file is successfully uploaded, the UGPApp server evaluates the file name, performs a signature verification over the file, and if successful, loads and activates the new access control policies indicated in the file. Activation involves two activities: 1) sending an application-specific notification to the security subsystem that a new policy file exists, and 2) invoking an ISO/TS 21177 'Configure' operation if the policy indicates TLS secure session policy changes. The latter case is described in use case **Configure TLS**.

Actors

- ConfigRole: Connects to the UGPApp over TLS and uploads the access control policy file.
- PAPPolicyManager: Entity that signed the policy file. This role can also be authorized to connect and upload the policy file.
- UGPApp: Receives the policy file and activates it.

Prerequisites

a) Security Pre-configuration:

1) The UGPApp ISO21177 SecureSession component is configured with the following:

i) A 1609.2 server certificate 'Server TLS Cert', containing the UGPApp ITS-AID and SSP. The SSP is defined as:

```
UgpServerSsp ::= AutomotiveSsp (WITH COMPONENTS {...,
   version PRESENT,
   protocol (WITH COMPONENTS {...,
      protocolVersion PRESENT,  -- '1' for version 1
      protocolRef (WITH COMPONENTS {...,
         ugp PRESENT})}),
   endpointType (WITH COMPONENTS {...,
          server PRESENT}),
   role ABSENT
   }
)
```

ii) One or more 1609.2 CA certificates comprising a complete trust chain from the 'Server TLS Cert' to a self-signed root.

2) If no operational access policy has been established, there exists a 'bootstrapping' policy that enables 1) an authorized role to connect to a secure session, and 2) the role to perform a file upload and activation of the new policy file. Once loaded, operational policies (not bootstrap policies) are enforced.

3) The security subsystem is also pre-configured with the identical 'Server TLS Cert' and the CA certificates are given to the secure session component. The CA certificate(s) are also present.

4) The UGPApp server application has an existing access control policy file defining:

i) ConfigRole and its permission to:

I) Securely Connect

II) Perform a file upload

ii) PAPPolicyManager role and its permission to activate/sign a policy file

NOTE The UGPApp can define a bootstrapping state in which a manufacturer-provided device contains a bootstrapping policy enabling a custom role to perform a one-time connection/upload policy to infuse the UGPApp with the necessary roles and policies, especially those of ConfigRole and PAPPolicyManager.

b) The PAPPolicyManager has completed preparing and signing the policy file "V_ITS_G-ACCESS-CONTROL-POLICY" as per the flow and requirements of the use case **Define an access control policy.**

c) The role, ConfigRole, which is allowed file upload access, has been provided with the policy file V_ITS_G-ACCESS-CONTROL-POLICY.

NOTE Either the ConfigRole or PAPPolicyManager can be configured to perform the actual upload, by policy. Enabling one role to sign the file and another to upload it allows application designers to implement multi-party controls, such as separation of duties, if so needed.

Flow

The process steps for this use case are as follows:

The ConfigRole:

a) initiates a TLS connection with the UGPApp secure session, according to ISO/TS 21177:2019, 6.5.

The UGPApp SecureSession:

a) authenticates the ConfigRole and checks the list of TLS constraints either provided it by its own security subsystem during the ISO/TS 21177 configure operation, or given to it as a manufacturer bootstrapping default. These constraints list only the authorized ITS-AID and SSP tuples that are allowed to connect, in conformance with the provisioned V_ITS_G-ACCESS-CONTROL-POLICY file.

b) passes the ConfigRole's IEEE 1609.2 certificate to the UGPApp's security subsystem, as per ISO/TS 21177:2019, 9.3.3 (Sec-Sess-StartSession.indication), such that the UGPApp can determine conclusively whether the connection is permitted.

The UGPApp Security Subsystem:

a) checks the configured access control policy V_ITS_G-ACCESS-CONTROL-POLICY TlsPolicyConfig to determine if the received client certificate SSP matches one of the valid SSPs in the currently active ConfigTls.

b) (if the connection is permitted) passes the session ID to the UGPApp for subsequent UGP messaging over TLS. The TLS session is now active between the ConfigRole and UGPApp.

The ConfigRole:

a) generates a UGP ManageFileCall message, as follows:

— *activityType* = upload

— *fileType* = configuration

— *fileName* = "V_ITS_G-ACCESS-CONTROL-POLICY"

— *fileSize* = ABSENT

— *Data* = The COER-encoded OctetString of the file V_ITS_G-ACCESS-CONTROL-POLICY, containing the 1609.2-signed access control configuration data

— *crc* = ABSENT

b) wraps the ManageFileCall message in an Iso21177AdaptorLayerPDU with Adaptor Layer PDU type=Apdu.

c) sends the Adaptor Layer wrapped PDU (ALPDU) over its own Secure Session (TLS Client) to the UGPApp.

The UGPApp receives the file and determines if it needs an access control check:

a) performs the following pre-processing: Detects the message type as a 'UGP ManageFile', which it determines requires an access control check by the security subsystem (this can be performed by caching a list of services contained in the access control policy)

b) relays the APDU to the security subsystem in an ISO/TS 21177 App-Sec-Incoming.request primitive

The UGPApp security subsystem performs the access control check, as follows:

a) applies the access control policy to the requested UGP service call:

   1) identifies the role indicated in the 1609.2 certificate associated with the TLS connection (session ID). The role identifier indicates 'ConfigRole'.

   2) evaluates the policy file's serviceMask associated with the ConfigRole to determine if the ManageFile 'upload' service (ugp-managefile-upload) is permitted.

b) If the service is permitted: sends the UGPApp an App-Sec-Incoming.confirm primitive with result='*Success*' back to the UGPApp.

c) If the service is not permitted: sends the UGPApp an App-Sec-Incoming.confirm primitive with result='*Invalid APDU per access control policy/no request sent*'. UGPApp logs the error and stops processing.

The UGPApp, if the file upload access was permitted:

a) detects that the filename is "V_ITS_G-ACCESS-CONTROL-POLICY".

b) extracts the filename from within the authenticated file ASN.1 to ensure it is the same name:

   1) (if the names do not match): stops processing and logs the error.

   2) (If the names match): continues.

c) performs a 1609.2 signature verification over the file:

   1) if signature fails, stops processing and logs the error.

   2) if successful, continues as follows:

      i) If the signature was generated by any role other than PAPPolicyManager, the UGPApp does not load and activate the policy. In this case, it stops processing and logs an error.

      ii) If the signature was generated with a valid 1609.2 certificate indicating the PAPPolicyManager role SSP, UGPApp replaces the existing access control file in integrity-protected memory.

      iii) Provides a 'UGP access activation' indicator to the security subsystem that the new policy file is active. This indicator should be implemented by the UGPApp vendor to inform the security subsystem to:

         I) Refresh its role definitions.

         II) Refresh its UGP service access policies.

         III) Refresh its Resource access policies, per role.

IV) If a new TLS policy is indicated in the file, performs use case **Configure TLS**, in which the ISO/TS 21177 'Configure' operations are performed.

7.6.4 Configure TLS

Objective

The objective of this use case is two-fold:

1) Trigger activation of UGPApp's new internal TLS access policy based on information provided in the V_ITS_G-ACCESS-CONTROL-POLICY. This is performed by sending an ISO/TS 21177 App-Sec-Configure to the security subsystem.

2) Configure the UGPApp secure session with the newly activated policy. This is performed by the security subsystem following the App-Sec-Configure operation by invoking Sec-Sess-Configure with the secure session.

Description

Secure session policies are defined in a customized extension substructure of the ISO/TS 21184 ACConfig (configFile name is "V_ITS_G-ACCESS-CONTROL-POLICY") called TlsPolicyConfig, defined as follows:

```
-- TLS-related policy configuration extension for the Security Subsystem and Secure
Session

TlsPolicyConfig ::= SEQUENCE { -- this message is wrapped in a ISO21177 CONFIGURE
       policies        SEQUENCE OF ConfigTls,
       activePolicy    PolicyId, -- Identifier of the active TLS policy
       ...
}
ConfigTls ::= SEQUENCE {
    policyId            PolicyId, -- unique ID for these role-based TLS constraints
    ssps                SecureSessionAllowedSsps,
    endpointType        EndpointType,
    secureTransports    SecureTransports,
    timeoutParams       TimeoutParams,
   ...
}

PolicyId ::= VisibleString

EndpointType ::= CHOICE {
   client             INTEGER(0),
   server             INTEGER(1),
   ...
}

SecureSessionAllowedSsps ::= SEQUENCE OF ClientSsp

SecureTransports ::= BIT STRING {
    none                          (0),
    tls                           (1),
    dtls                          (2),
    reserved1                     (3),
    reserved2                     (4),
    reserved3                     (5),
    reserved4                     (6),
    reserved5                     (7)
}  (SIZE(8, ...))

TimeoutParams ::= SEQUENCE {
    inactivityTimeout    Time32, -- inactivity timeout in seconds
    sessionTimeout       Time32, -- max session timeout in seconds
    ...
}
```

Implementers of ISO/TS 21177 and ISO/TS 21184 require configuration for the TLS secure session to make granular policy decisions related to the TLS session and not just resources accessed through it.

This structure contains a sequence of configTls entries, each uniquely identified, and an indicator of which configTls is 'active.' The active configTls is set in the file, as received. However, some application designers can implement a dynamic capability to manage internally or externally, for example via API, which configTls is active without having to load a new V_ITS_G-ACCESS-CONTROL-POLICY each time. If this is done, it is configured to be invoked only by an authorized role.

Each ConfigTls TLS configuration policy contains:

- a policyId that the UGPApp or external management application can maintain and use to reference and activate a specific TLS access policy. An old policy can be used if the access control policy file does not contain a new ConfigTls.
- SSPs, consisting of a set of one or more ClientSsp that are allowed to connect. ClientSsp is a subtype instance of the AutomotiveSsp structure defined for the IEEE 1609.2 certificate. The security subsystem will send this list of permitted SSPs, and for each a ITS-AID, to the secure session service in the ISO/TS 21177 Sec-Sess-Configure primitive.
- an endpointType. This parameter simply tells the UGPApp whether it is a 'server' or 'client.' This information is provided to the UGPApp's security system as the 'Role' parameter in an App-Sec-Configure request, as per ISO/TS 21177:2019, 7.6.1.
- a secureTransports bit string indicating which options of secure transport, 'tls', 'dtls', a client with a valid SSP can use to connect.
- a timeoutParams structure, which provides both a timeout value for inactivity as well as one for maximum session time. Upon either timeout condition, a client establishes a new connection in order to continue services. TimeoutParams are provided by the security subsystem to the secure session when activating a new TLS access policy during the ISO/TS 21177 Sec-Sess-Configure operation.

Upon load and activation of the new V_ITS_G-ACCESS-CONTROL-POLICY in the **Load an Access Control Policy** use case, UGPApp 1) extracts and interprets the TlsPolicyConfig from the file, and 2) sends the security subsystem an App-Sec-Configure and signals the security subsystem to activate the new policy. Part of this activation is the Sec-Sess-Configure operation the security subsystem performs with the secure session.

Prerequisites

- V_ITS_G-ACCESS-CONTROL-POLICY has been uploaded and activated by UGPApp.
- V_ITS_G-ACCESS-CONTROL-POLICY contained a TlsPolicyConfig. If no TlsPolicyConfig is present, then UGPApp retains and continues to use the previous TlsPolicyConfig.

Actors

- UGPApp
- UGPApps security subsystem
- UGPApp secure session

Flow

The process steps for this use case are as follows:

The UGPApp:

a) upon receiving and activating an uploaded access control policy, V_ITS_G-ACCESS-CONTROL-POLICY, performs the following:

   1) If the file does not contain a TlsPolicyConfig:

      i) stops processing this use case,

      ii) only updates the policies listed in V_ITS_G-ACCESS-CONTROL-POLICY as per use case Load an Access Control policy,

      iii) continues to enforce the existing TlsPolicyConfig and the single, active ConfigTls it indicates.

   2) If the file contains a TlsPolicyConfig update structure:

      i) extracts and updates its internal TlsPolicyConfig,

      ii) uses the activePolicy contained within the TlsPolicyConfig to send an App-Sec-Configure. request to its security subsystem, as per ISO/TS 21177:2019, 7.6.1. This message to the security subsystem reflects the configuration of role-based SSPs within TLS policy indexed by the activePolicy.

The security subsystem:

a) sends the UGP server an App-Sec-Configure.confirm, as per ISO/TS 21177:2019, 7.6.2.

b) sends the secure session a Sec-Sess-Configure.request containing new information indicated in the TlsConfiguration in the TlsAccessPolicyFile:

   1) Endpoint type, i.e. 'server'.

   2) A CertificatePermissions comprising an array of ITS-AID and SSP tuples, one for each TlsRole in the active ConfigTls.

   3) Inactivity timeout parameter from the ConfigTls timeoutParams.

   4) Session timeout parameter from the ConfigTls timeoutParams.

The secure session:

a) creates a secure session instance with the indicated parameters as specified in ISO/TS 21177:2019, 6.4.

b) sends a Sec-Sess-Configure.confirm message to the security subsystem.

c) the secure session instance is now correctly configured with the new TLS access policy.

7.6.5 Start a secure TLS session

Objective

The objective of this use case is to start a TLS secure session and ensure only authorized clients possessing a valid 1609.2 certificate SSP can establish the secure session with the UGPApp server.

Description

The previously documented use cases **Define a TLS Access Control Policy, Load an Access Control Policy and Configure TLS** have resulted in a configured TLS security policy within the UGPApp secure session service that can now be used to establish a cryptographic secure TLS session (as per ISO/TS 21177) between a UGPApp 'server' and a UGP client. The UGPApp secure session service performs a TLS handshake with the UGP client based on the V_ITS_G-ACCESS-CONTROL-POLICY

TlsPolicyConfig. TlsPolicyConfig defines the authorized secureTransport method(s) and includes a sequence of authorized ClientSsps that the server will check exists in the client-presented 1609.2 certificate. If the access constraints defined within TlsPolicyConfig are violated, the Secure Session will not complete the handshake and the UGPApp and UGP client will have no connectivity. If the access constraints are satisfied, then 1) the handshake will complete; and 2) the secure session notifies via the Sec-Sess-StartSession.indication primitive that the session has started and passes the UGP Client's certificate and session Id to the UGPApp security subsystem. This information is then used in later use cases to determine if 1) the session is still active; and 2) the role asserted in the UGP client's certificate can be used to make access control decisions within the UGP application.

NOTE More advanced use cases can incorporate the security subsystem to initiate any additional access control activities that can be required per policy. See the section on Enhanced Access Control Functionality for more details.

Actors

- UGP Client Application: VehDiagRole
- UGPApp Secure Session
- UGPApp Adaptor Layer
- UGPApp ISO 21177 security subsystem

Prerequisites

1) The UGPApp secure session has been configured with a TLS Access Control Policy V_ITS_G-ACCESS-CONTROL-POLICY TlsPolicyConfig.

2) The client IEEE 1609.2 certificate contains an appPermissions field that specifies the UGP application and an authorized role for access to the requested resource.

3) The UGP Client is configured to communicate using the ISO/TS 21177 primitives listed in this use case.

Flow

The process steps for this use case are as follows:

The UGP Client, assuming the VehDiagRole:

a) transmits a TLS ClientHello message to the UGPApp secure session service.

The secure session of the UGPApp:

a) authenticates the VehDiagRole and checks the list of TLS constraints provided to it by its own security subsystem during the ISO/TS 21177 configure operation. These constraints list only the authorized ITS-AID and SSP tuples that are allowed to connect, in conformance with the provisioned V_ITS_G-ACCESS-CONTROL-POLICY file and the single, active ConfigTls.

b) passes the VehDiagRole's IEEE 1609.2 certificate to the UGPApp's security subsystem via an ISO/TS 21177 Sec-Sess-StartSession.indication primitive, which includes the application ID, session Id and client certificate. This allows the UGPApp to determine conclusively whether the connection is permitted.

The UGPApp security subsystem:

a) checks the configured access control policy V_ITS_G-ACCESS-CONTROL-POLICY TlsPolicyConfig to determine if the received client certificate SSP matches one of the valid SSPs in the currently active ConfigTls. For this use case, the UGPClient is asserting the VehDiagRole and as such, the TlsPolicyConfig SecureSessionAllowedSsps contains the VehDiagRole (2).

b) (if the connection is permitted) provides the session ID for subsequent UGP messaging over TLS via the App-Sec-StartSession.indication primitive.

c) if successful, generates an AccessControlRequest.success access control PDU and sends the AccessControlRequest.success PDU to the secure session service via the Sec-AL-AccessControl.result primitive.

AccessControlResult::=INTEGER{success(0)}

d) if unsuccessful, two options exist:

1) End the session without notification to the UGP Client via a Sec-AL-EndSession.request primitive.

2) Generate an AccessControlRequest access control PDU with status failure and send the PDU to the secure session service via the Sec-AL-AccessControl.Result primitive.

AccessControlResult::=INTEGER{failure(1)}

The Security Adaptor Layer:

a) Wraps the access control request PDU within an ALPDU, an Iso21177AdaptorLayerPdu with the component messageId equal to apduId and Data parameter from this service primitive.

b) Generates an AL-Sess-Data.request primitive with (Application ID, Session ID, ALPDU) as the application ID, session ID, data parameters and sends to the secure session service.

The secure session service:

a) Sends back to the UGP client a Server Hello message containing the server's IEEE 1609.2 certificate:

```
UgpServerSsp ::= AutomotiveSsp (WITH COMPONENTS {...,
   version PRESENT,
   protocol (WITH COMPONENTS {...,
      protocolVersion PRESENT,  -- '1' for version 1
      protocolRef (WITH COMPONENTS {...,
         ugp PRESENT})}),
   endpointType (WITH COMPONENTS {...,
          server PRESENT}),
   role ABSENT
   }
)
```

b) The TLS session is now active between the VehDiagRole and UGPApp via the UGPApp Secure Session.

A sequence diagram for this use case is given in Figure 16.

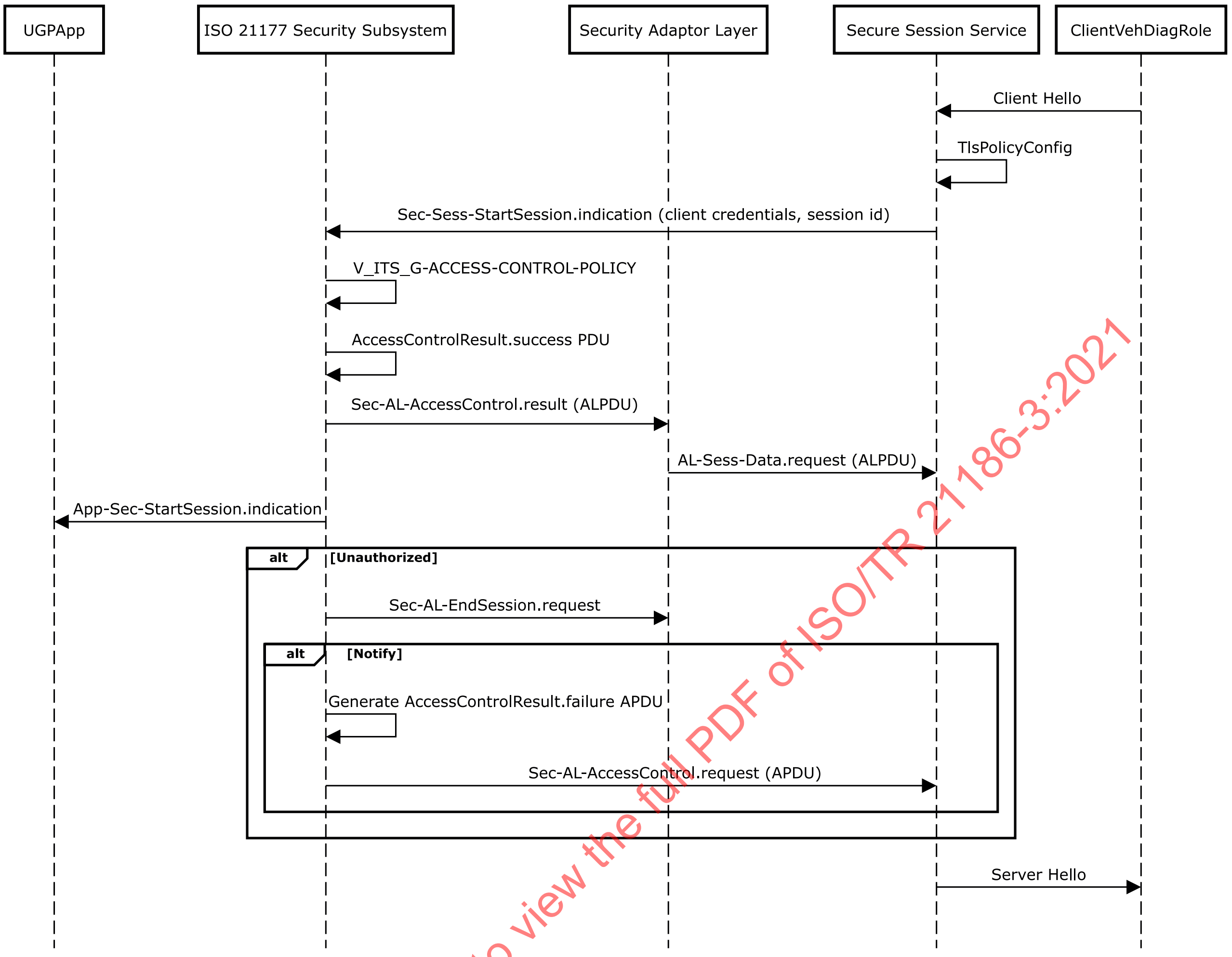
UGPApp
ISO 21177 Security Subsystem
Security Adaptor Layer
Secure Session Service
ClientVehDiagRole
Client Hello
TlsPolicyConfig
Sec-Sess-StartSession.indication (client credentials, session id)
V_ITS_G-ACCESS-CONTROL-POLICY
AccessControlResult.success PDU
Sec-AL-AccessControl.result (ALPDU)
AL-Sess-Data.request (ALPDU)
App-Sec-StartSession.indication
alt
[Unauthorized]
Sec-AL-EndSession.request
alt
[Notify]
Generate AccessControlResult.failure APDU
Sec-AL-AccessControl.request (APDU)
Server Hello

Figure 16 — Start TLS session sequence

7.6.6 Secure access-controlled resource discovery

Objective

The objective of this use case is to allow a client to the UGPApp server to discover only what UGP-accessible resources are available to it, based on 1) the access control policy defined for the client type, and 2) the client's 1609.2 certificate permissions obtained by the UGPApp security subsystem during establishment of the TLS secure session.

Description

In the previous use case, a UGPApp and a client established a secure TLS session. After completion of the TLS handshake, the UGPApp secure session service passed the client credentials and session Id to the ISO/TS 21177 security subsystem via the Sec-Sess-StartSession.indication primitive. This information is used by the ISO/TS 21177 security subsystem to perform a simple role-based access control for UGP service and resource requests.

The GetSupportedDataCall message requests the supported data parameters available from the IVN. Depending on the filters applied, the message can request the rvIDs or a mapping of ecuIDs to rvIDs for all supported data. A listing of ecuIDs can also be added to filter specifically ECUs within the IVN.

GetSupportedDataCall also supports an optional accessType. Access type is defined in ISO/TS 21184 as:

```
AccessType::= BIT STRING {
   r  (0),
    w  (1),
    x  (2),
    i  (3),
    u  (4)
} (SIZE(5, ...))
```

A GetSupportedDataCall message containing a simple request for ALL supported data parameters would, without access control checks, return a listing of all supported rvIDs and an optional mapping to the associated ecuIDs. UGPApp's access control policy evaluation of GetSupportedDataCall takes into account the role associated with message request and appropriately filters the GetSupportedDataReply message that is returned from the UGPApp to the Client. No rvIDs to which the client has access should be returned to the client.

For this example, the VehDiagRole is associated with the TLS session of the message and also with the ISO/TS 21177 security subsystem that is responsible for making the access control determination based on the role-based privileges defined in the current access control policy.

The access policy evaluation will consist of the following steps:

1) Identify whether the calling role (VehDiagRole) has access to the GetSupported UGP service. Based on the pre-configured policy, VehDiagRole is authorized to access the service.

2) Examine the policy file ACEntrySet which contains a sequence of each authorized ecuID, rvID and accessType mapping.

```
ACEntrySet ::= SEQUENCE {
     id   Uint16,
     entrySEQUENCE OF ACEntry,
     ...
}
ACEntry ::= SEQUENCE {
     ecuId                Identifier,
     rvId                 Identifier,
     accessType           AccessType,
     ...
}
```

Upon receipt of the GetSupportedDataCall message, the UGPApp determines that the message requires an access check (this can be performed by caching a list of services contained in the access control policy) and sends the security subsystem the wrapped APDU via an App-Sec-Incoming.request primitive. The UGPApp's security subsystem evaluates the request against the ACEntrySet for the calling role and based on its privileges, either:

a) Indicates to the UGPApp application that the APDU can be processed without any filtering by sending an App-Sec-Incoming.confirm (success) primitive.

b) Indicates to the UGPApp application that the APDU is processed with filtering by sending an App-Sec-Incoming.confirm (success) primitive along with an application-defined directive to perform filtering of the rvID/ecuID tuples and return only those to which the role has access. The directive indication is not currently supported in ISO/TS 21177 and can therefore be implemented in a manner custom to the UGPApp application. The Directive can consist of an instruction to:

    1) look up the role's resource access policy

    2) build a response message that contains rvID returned results where the rvID is both:

        i) accessible per the policy

        ii) accessible/present from the UGP application

    3) determines that no requested data is accessible by the role, by sending to the UGPApp an App-Sec-Incoming.confirm (invalid APDU) primitive.

The UGPApp then constructs a filtered or unfiltered GetSupportedDataReply message, according to the presence or lack of a filter Directive from the Security Subsystem, wraps it within an

Iso21177AdapterLayerPDU with Adapter Layer PDU type=APDU, and lastly sends it to the secure session for secure TLS transmission back to the client.

Actors

- Client Application: VehDiagRole
- UGPApp Secure Session Service
- UGPApp Security Adaptor Layer
- UGPApp ISO21177 Security Subsystem

Prerequisites

- A cryptographic secure session has been previously established between the client application and the server Secure Session Layer and the session has not timed out.
- The security subsystem has been configured with a valid V_ITS_G-ACCESS-CONTROL-POLICY as described in **DEFINE A RESOURCE ACCESS POLICY.**

a) The security subsystem is designed with a set of enumerated directives that it can use to indicate to the application that only qualified access is granted. The directive in this use case instructs the application to filter the content of the GetSupportedDataReply to only those rvID/ecuID tuples to which the client role has access. In general, the directive should consist of an instruction to:

  1) look up the role's resource access policy, and

  2) build a response message that contains rvID returned results where the rvID is both:

    i) accessible per the policy, and

    ii) accessible/present from the UGP application.

- The client IEEE 1609.2 certificate contains an appPermissions field that specifies the UGP application and an authorized role with access to the requested resource.
- The UGP client is configured to communicate using the ISO/TS 21177 primitives listed in this use case.

Flow

The process steps for this use-case are:

The client asserting the VehDiagRole:

a) prepares a GetSupportedDataCall message.

```
GetSupportedDataCall::= SEQUENCE {
  supportedDataFilter SupportedDataFilter,
  ecuList           SEQUENCE OF Identifier OPTIONAL,
  accessType           AccessType            OPTIONAL,
  dataParamProperty    DataParamProperty       OPTIONAL,
}

SupportedDataFilter::= ENUMERATED {vehicle-info-only}
```

The GetSupportedDataCall is used to request the supported data parameters within the IVN. Various filters can be applied to the message including the accessType filter which requests a return of only supported data parameters of the specified access type.

From an access control perspective, however, the access type field requests a specific set of supported data parameters (e.g. R, W, X). During an access control evaluation, the policy file is evaluated to determine whether a restriction on any supported data parameters exists based on access type. AccessType for each rvID and ecuID tuple is defined as a bit string within the policy file.

b) wraps the GetSupportedDataCall APDU message in an Adaptor Layer PDU with type=APDU.

c) sends the APDU from its secure session to the UGPApp Secure Session Service as TLSData.

The UGPApp secure session service:

a) decrypts, defragments and authenticates the APDU.

b) sends the APDU to the UGPApp's Security Adapter Layer via the AL-Sess-Data.indication primitive.

The UGPApp Security Adapter Layer:

a) checks that the Adapter Layer header subtype is 'APDU', removes the Adapter Layer header and passes the embedded APDU to the UGPApp using the App-AL-Data.indication primitive. This primitive includes the application ID and session ID.

The UGPApp performs the following:

a) detects the message type as a 'GetSupportedDataCall', which it determines requires an access control check by the security subsystem.

b) relays the APDU to the security subsystem in an App-Sec-Incoming.request primitive, requesting that the security subsystem perform the access control check.

The UGPApp security subsystem applies the access control policy corresponding to the GetSupportedDataCall service call, as follows:

a) identifies the role indicated in the 1609.2 certificate associated with the session ID. The role identifier from the SSP indicates 'VehDiagRole'.

b) evaluates whether the VehDiagRole is authorized to access the GetSupportedData, i.e. 'ugp-getsupported' service indicated in the policy. This is indicated in the first bit of the ACService bit string:

```
ACRole ::= SEQUENCE {
    id                    Uint16,
    assignedPsid          Psid,
    name                  VisibleString    OPTIONAL,
    serviceMask           ACService        OPTIONAL,
    entrySetId            Uint16           OPTIONAL,
    allowedTransports     SecureTransports OPTIONAL,
    ...
}

ACService ::=  BIT STRING {
    ugp-getsupported        (0),
    ugp-getvalue              (1),
    ugp-setvalue              (2),
    ugp-controlvalue          (3),
    ugp-getdtcinfo        (4),
    ugp-cleardtcinfo         (5),
    ugp-enablepassthrough   (6),
    ugp-listfile              (7),
    ugp-managefile-download (8),
    ugp-managefile-upload   (9),
    ugp-managefile-delete   (10),
    ugp-reset               (11),
    sendDXM                 (12),
    activate-policy         (13)
} (SIZE(14, ...))
```

c) If the service is permitted for the VehDiagRole, the security subsystem performs the following:

1) (if all resources are accessible by the role) sends UGPApp an App-Sec-Incoming.confirm (success) primitive.

2) (if no resources are accessible by the role) sends to the UGPApp an App-Sec-Incoming.confirm primitive result= "Invalid APDU per access control policy/no request sent'.

3) (if some resources are accessible by the role) sends an App-Sec-Incoming.confirm (success-with-directive) primitive to the UGPApp application, indicating a directive to filter the response message according to the policy.

UGPApp receives the App-Sec-Incoming.confirm, and performs the following:

a) (if it indicates 'success') sends an unfiltered UGP GetSupportedDataReply message to the VehDiagRole client.

b) (if it indicates 'Invalid APDU per access control policy/no request sent') sends VehDiagRole client a GetSupportedDataReply with no resource identifier content.

c) (if it indicates a 'success-with-directive'):

   1) executes the directive, for example based on a directive ID, to 'filter the GetSupportedDataReply' message according to the current access control policy.

   2) examines the policy file ACEntrySet which contains a sequence of each authorized ecuID, rvID and accessType mapping.

      i) determines the authorized rvID and ecuID tuples that can be passed back to the UGP client in the GetSupportedDataReply message.

```
ACEntrySet ::= SEQUENCE {
id       Uint16,
entry       SEQUENCE OF ACEntry,
    ...
}


ACEntry ::= SEQUENCE {
ecuId                    Identifier,
rvId                     Identifier,
accessType           AccessType,
...
}
```

      ii) populates a GetSupportedDataReply message based on the received APDU from the security subsystem.

```
GetSupportedDataReply::= SEQUENCE {
    dataParamList SEQUENCE OF Identifier OPTIONAL,
    dataParamMapping SEQUENCE OF DataParamMapping OPTIONAL,
```

      iii) sends the GetSupportedDataReply APDU to the adaptor layer.

The UGPApp adapter layer:

a) wraps the message in an Iso21177AdapterLayerPDU with Adapter Layer PDU type=APDU

b) sends the ALPDU to the secure session

The UGPApp secure session:

a) acknowledges the receipt of the ALPDU via an AL-Sess-Data.confirm primitive.

b) sends the ALPDU over its secure session to the VehDiagRole's secure session.

A sequence diagram for this use case is given in Figure 17.

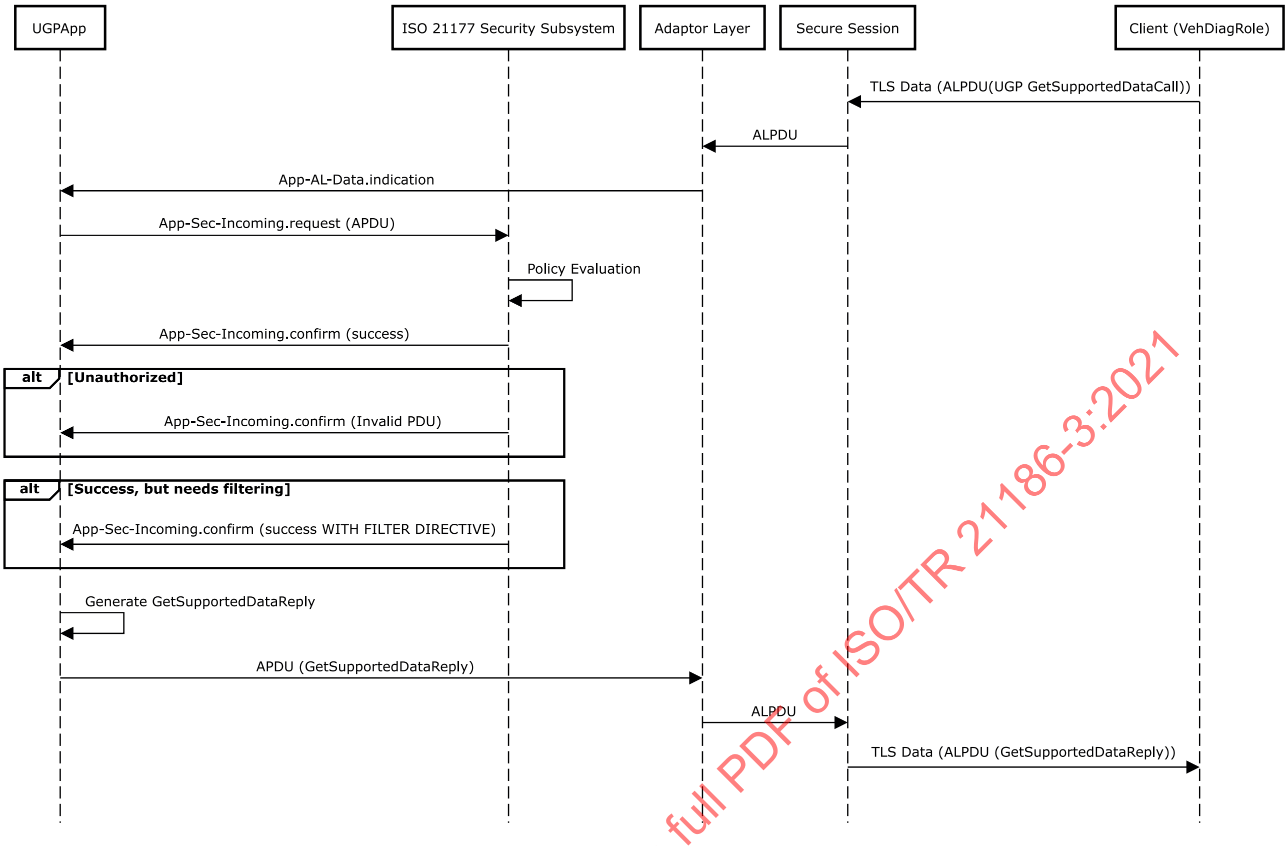
UGPApp
ISO 21177 Security Subsystem
Adaptor Layer
Secure Session
Client (VehDiagRole)
TLS Data (ALPDU(UGP GetSupportedDataCall))
ALPDU
App-AL-Data.indication
App-Sec-Incoming.request (APDU)
Policy Evaluation
App-Sec-Incoming.confirm (success)
alt
[Unauthorized]
App-Sec-Incoming.confirm (Invalid PDU)
alt
[Success, but needs filtering]
App-Sec-Incoming.confirm (success WITH FILTER DIRECTIVE)
Generate GetSupportedDataReply
APDU (GetSupportedDataReply)
ALPDU
TLS Data (ALPDU (GetSupportedDataReply))

Figure 17 — Secure access-controlled resource discovery

7.6.7 Server controls access to UGP service based on role

Objective

The objective of this use case is to restrict access to a UGP-accessible service/resource using:

- the stored access control policy defined for the client role, and
- the certificate permissions of the client, as indicated in its 1609.2 certificate.

Description

In this use case, following a successful creation of a TLS secure session and the successful identification of supported UGP data parameters as described in the use case **Secure Access Controlled Resource Discovery**, the UGP client asserting the VehDiagRole transmits a UGP GetValue message to the UGPApp. The UGPApp loads the previously configured V_ITS_G-ACCESS-CONTROL-POLICY to evaluate the request and make a PERMIT or DENY decision. Additionally, the security subsystem can apply a filter to the GetValue request that results in the UGPApp sending back only the authorized values. If authorized, a GetValueReply message is sent back to the UGP client.

Actors

- Client Application: VehDiagRole
- UGPApp Secure Session
- UGPApp Adaptor Layer
- UGPApp ISO 21177 Security Subsystem

Prerequisites

- A cryptographic secure session has been previously established between the client application and the server Secure Session Layer and the session has not timed out.
- The security subsystem has been configured with a valid V_ITS_G-ACCESS-CONTROL-POLICY, as described in **DEFINE A RESOURCE ACCESS POLICY.**
- The client IEEE 1609.2 certificate contains an appPermissions field that specifies the UGP application and an authorized role for access to the requested resource.
- The UGP client is configured to communicate using the ISO/TS 21177 primitives listed in this use case.
- The UGP client has completed a discovery session resulting in a listing of available and authorized UGP data parameters, each identified by a given rvID.

Flow

The process steps for this use case are:

The client asserting the VehDiagRole:

a) (From the **Secure access-controlled resource discovery** use case) receives the Iso21177AdapterLayerPDU, removes the applied header noting that the wrapped message is an APDU and parses the GetSupportedDataReply message. This message contains either a listing of the available and authorized rvIDs or a mapping of the available and authorized rvIDs to ecuIDs, as defined from ISO 13185-2:

```
GetSupportedDataReply::= SEQUENCE {
dataParamList SEQUENCE OF Identifier OPTIONAL,
dataParamMapping SEQUENCE OF DataParamMapping OPTIONAL,
```

b) uses the contents of the GetSupportedDataReply message to create a UGP GetValueCall message. The GetValueCall message includes a dataParamList containing the rvIDs or the data parameter mapping which contains the mapping between data parameter (by rvID) to ecuID.

```
GetValueCall::= SEQUENCE {
testInterval SNUM32 DEFAULT 0,
dataParamList SEQUENCE OF Identifier OPTIONAL,
dataParamMapping SEQUENCE OF DataParamMapping OPTIONAL,
condition ComplexCondition OPTIONAL,
}
```

c) wraps the UGP GetValueCall APDU in an Iso21177AdapterLayerPDU with Adapter Layer PDU type=APDU.

```
Apdu ::= OCTET STRING
```

d) sends the ALPDU over its secure session (TLS client) to the UGPApp secure session service as TLS data.

The UGPApp secure session service:

a) decrypts, defragments and authenticates the ALPDU.

b) sends the ALPDU to the UGPApp's Adaptor Layer in an AL-Sess-Data.indication primitive.

The UGPApp Adaptor Layer:

a) checks that the Adaptor Layer header subtype is 'APDU', removes the Adapter Layer header and passes the embedded UGP APDU to the UGPApp application using the App-AL-Data.indication primitive. This primitive includes the application ID and session ID.

The UGPApp:

a) receives the APDU.

b) detects the message type as a 'GetValueCall', which it determines requires an access control check by the security subsystem.

c) relays the APDU to the security subsystem in an App-Sec-Incoming.request primitive requesting that the security subsystem perform access control checks.

The UGPApp security subsystem:

a) receives the App-Sec-Incoming.request from UGPApp.

b) applies the access control policy to the requested UGP service call, as follows:

   1) identifies the role indicated in the 1609.2 certificate associated with the TLS connection (session ID). The role identifier indicates 'VehDiagRole'.

   2) evaluates the policy file's serviceMask associated with the VehDiagRole to determine if the ugp-getValue service is permitted.

```
ACRole ::= SEQUENCE {
    id                     Uint16,
    assignedPsid           Psid,
    name                   VisibleString OPTIONAL,
    serviceMask            ACService     OPTIONAL,
    entrySetId             Uint16        OPTIONAL,
    allowedTransports      SecureTransports OPTIONAL,
    ...
}
ACService ::=  BIT STRING {
    ugp-getsupported               (0),
    ugp-getvalue              (1),
    ugp-setvalue              (2),
    ugp-controlvalue          (3),
    ugp-getdtcinfo         (4),
    ugp-cleardtcinfo             (5),
    ugp-enablepassthrough        (6),
    ugp-listfile              (7),
    ugp-managefile-download        (8),
    ugp-managefile-upload          (9),
    ugp-managefile-delete          (10),
    ugp-reset                      (11),
    sendDXM                        (12),
    activate-policy                (13)
} (SIZE(14, ...))
```

c) If the service is permitted for the VehDiagRole, the security subsystem proceeds with an evaluation of the GetValueCall message's requested content. The security subsystem examines the policy file ACEntrySet which contains a sequence of each authorized ecuID, rvID and accessType mapping. The values requested in the GetValueCall message are compared against the values authorized for the role in ACEntrySet and the specific access types.

```
ACEntrySet ::= SEQUENCE {
id       Uint16,
entry SEQUENCE OF ACEntry,
...
}

ACEntry ::= SEQUENCE {
ecuId                Identifier,
rvId                  Identifier,
accessType         AccessType,
...
}
```

An example of an ACEntry is the following:

{ rvId 1002, ecuId 17, 0 },

{ rvId 2341, , 0 }

{rvId 461, , 0 }

This ACEntry list would provide the authorized role the ability to 1) read the engine control module voltage (ECMV) from the Engine Control Module ECU, 2) read the engine coolant temperature, and 3) read the VIN.

d) The actual method for performing the policy evaluation is implementation-specific. However, the basic process involves the security subsystem determining whether each requested rvID contained in the dataParamList or dataParamMapping fields of GetValueCall is accessible to the client. The typical steps involved in the evaluation are:

   1) Identify the first rvID specified in the GetValueCall dataParamList. Determine if the identified rvID is included in the ACEntrySet for the asserted role.

      i) If not included in ACEntrySet, stop processing. Request is not authorized.

      ii) If included in ACEntrySet continue processing.

   2) Identify the next and subsequent rvIDs specified in the GetValueCall dataParamList. Determine if the identified rvIDs are included in the ACEntrySet for the asserted role.

   3) If a dataParamMapping is provided within the GetValueCall, then evaluate whether ACEntrySet includes authorized access for both the rvID and associated EcuID specified in each dataParamMapping entry.

e) If the service is permitted:

   1) sends the UGPApp an App-Sec-Incoming.confirm primitive with result = success back to the UGPApp.

f) If the service is not permitted:

   1) sends the UGPApp an App-Sec-Incoming.confirm primitive with result="Invalid APDU per access control policy/no request sent'.

g) If only some results of the getValueCall are permitted:

   1) provides a 'success-with-directive' App-Sec-Incoming.confirm back to the UGPApp indicating the service can be performed, but the results filtered. The application-specific directive accompanying the App-Sec-Incoming.confirm then indicates a sequence of permitted rvIDs or tuples of rvID and ecuID to reflect only the authorized rvIDs and/or ecuIDs.

The UGPApp:

a) receives the App-Sec-Incoming.confirm indicating success or an App-Sec-Incoming.confirm with:

   1) an optional directive, and

   2) permitted list of rvID/ecuID

b) populates a GetValueReply message based on the received APDU from the security subsystem. The GetValueReply contains either a list of the requested data parameter values defined by ecuID, rvID and a value. The actual value is a sequence of DataParamValueTS which includes an optional timestamp.

```
GetValueReply::= SEQUENCE {
valueTS SEQUENCE OF DataParamValueTS,
...
}

DataParamValueTS::= SEQUENCE { -- timestamp
value DataParamValue,
timeInMillis UNUM64 OPTIONAL,
. . .
}
```

c) wraps the GetValueReply message in an Iso21177AdapterLayerPDU with Adapter Layer PDU type=APDU and sends to the Adaptor Layer in an App-AL-Data.request primitive.

The UGPApp Adaptor Layer:

a) adds an Adaptor Layer Data Header to the APDU indicating that the ALPDU is an Application datagram and sends the ALPDU to the UGPApp Secure Session Layer in an AL-Sess-Data.request primitive.

The UGApp Secure Session Layer:

a) acknowledges the receipt of the ALPDU via an AL-Sess-Data.confirm primitive.

b) sends the Adapter Layer wrapped PDU (ALPDU) over its secure session (TLS client) to the VehDiagRole (UGP client).

A sequence diagram for this use case is given in Figure 18.

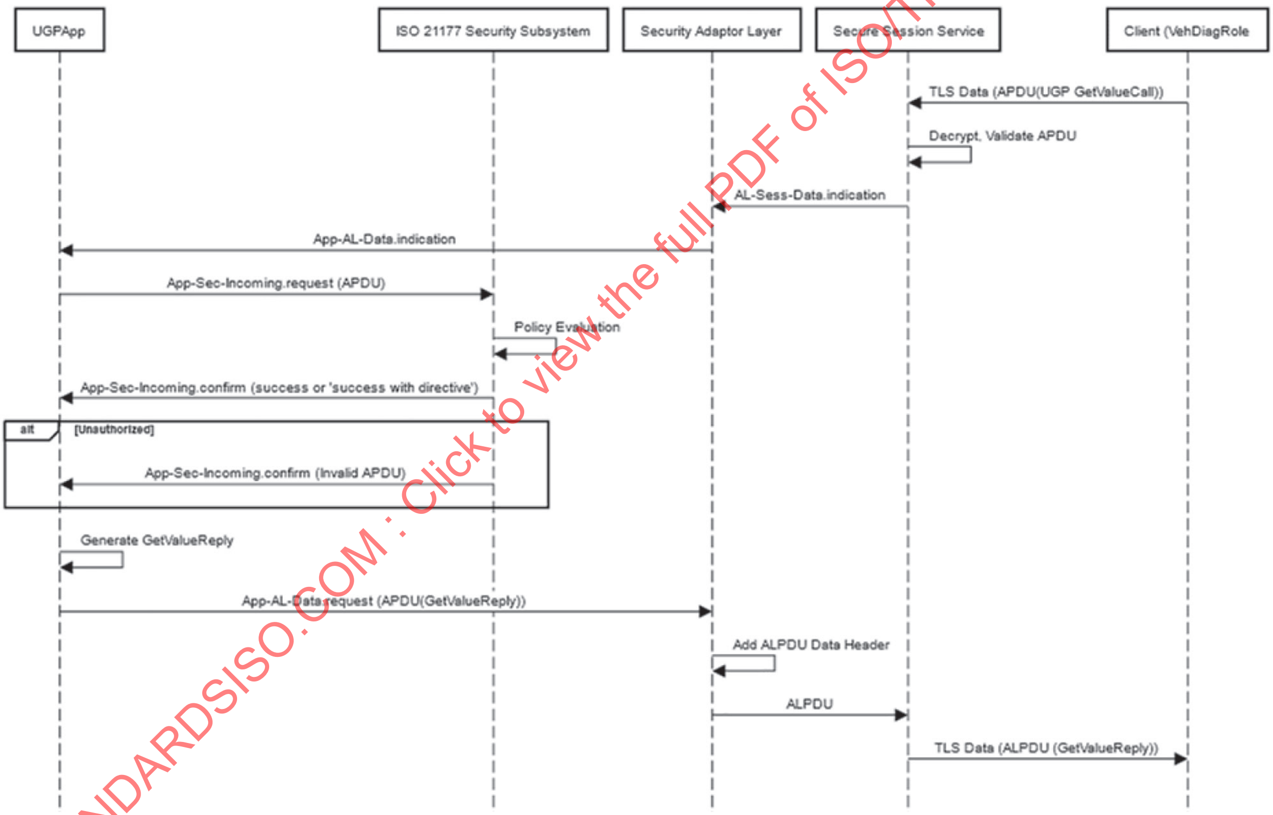
UGPApp
ISO 21177 Security Subsystem
Security Adaptor Layer
Secure Session Service
Client (VehDiagRole
TLS Data (APDU(UGP GetValueCall))
Decrypt. Validate APDU
AL-Sess-Data.indication
App-AL-Data.indication
App-Sec-Incoming.request (APDU)
Policy Evaluation
App-Sec-Incoming.confirm (success or 'success with directive')
alt
[Unauthorized]
App-Sec-Incoming.confirm (Invalid APDU)
Generate GetValueReply
App-AL-Data.request (APDU(GetValueReply))
Add ALPDU Data Header
ALPDU
TLS Data (ALPDU (GetValueReply))

Figure 18 — Server controls access to resource based on role

8 C-ITS CP security requirements gaps and needs

8.1 General

Critically important to device and ecosystem ITS-S security are the security provisions of the European ITS PKI undergoing implementation. This PKI establishes a top-level root of trust implemented in an entity called the TLM. The C-ITS PKI is responsible for provisioning the cryptographic and operational trust capabilities of the C-ITS trust infrastructure. Facing myriad threats, it will be needed to support many new types of C-ITS-Ss ranging from vehicles, test equipment and roadside infrastructure.

This clause features:

- a brief overview of Version 1.1 of the European C-ITS PKI;
- a determination of C-ITS PKI-related threats and methods to mitigate;
- a high-level gap analysis regarding the PKI's ability to support the demands of new ITS applications and their associated data sensitivities.

8.2 Overview of European C-ITS CP

The European C-ITS CP[27] defines a set of minimal, baseline security requirements for the centralized C-ITS PKI, its principal roles, participant Root CAs and their EAs and AAs. Architecturally, the PKI centralizes all trust of root CAs via Root CAs and their trusted association with the TLM. The TLM, via the CPOC, periodically updates, digitally signs and publishes the ECTL, containing each trusted root certificate (see Figure 19, where the red lines represent the trust relations). Interfaces, some technical and some procedural, are established between the following principle roles:

- C-ITS CP authority
- TLM – The authority for managing and maintaining the ECTL
- C-ITS CPOC – The organizational interface between Root authorities and the PKI's TLM
- Root CAs – Public or private organizations operating public key roots entered into the ECTL
- Enrolment Authorities – Responsible for enrolling end entities to a Root CA.
- Authorization Authorities – Responsible for issuing operational certificate to end entities possessing an Enrolment certificate acquired from an EA.

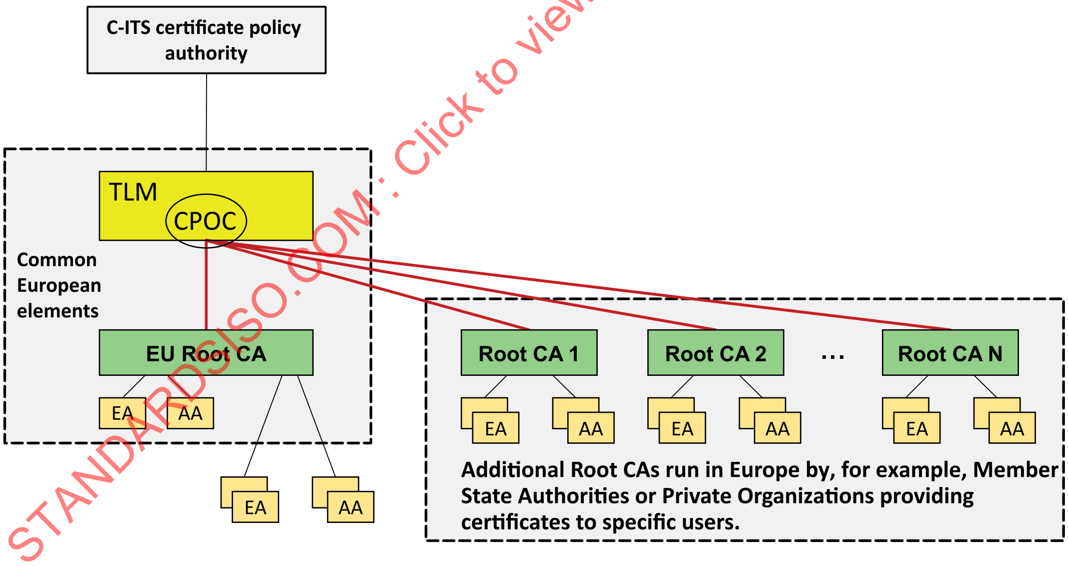
C-ITS certificate policy authority
TLM
CPOC
Common European elements
EU Root CA
EA
AA
EA
AA
Root CA 1
Root CA 2
...
Root CA N
EA
AA
EA
AA
EA
AA
Additional Root CAs run in Europe by, for example, Member State Authorities or Private Organizations providing certificates to specific users.

Figure 19 — European C-ITS PKI, High-level architecture[27]

The CP conforms to IETF RFC 3647, addressing the following PKI-related topics:

1) PKI participants

2) Certificate usage

3) CP administration

4) Publication and repository responsibilities

5) Identification and authentication

6) Certificate lifecycle operational requirements

7) Facility, management and operational controls

8) Technical security controls

9) Certificate profiles

10) Compliance, audit and other assessments

11) Other business legal matters

Subclause 8.3 highlights threat categories and for each, a set of threat vectors and mitigations for the European C-ITS PKI system and its operations. Unmitigated, the identified threats have the potential to severely impact the C-ITS ecosystem and the proper functioning of trust relationships afforded to end entity devices.

8.3 PKI threat categories and mitigations

This subclause provides more detailed, relevant PKI-related threat vectors in each threat category and details for each a set of possible mitigations.

Each listed threat includes:

- an enumerated reference starting with a "T.***" for ease of reference;
- a high-level description, including whether it is an organizational process or technical area of concern;
- one or more threat vectors associated with the threat;
- one or more proposed mitigations for relevant stakeholders to address.

T.COMPROMISE_TRUST_LIST_MANAGER

<table>
<tr><td colspan="2"><b>Description:</b> A variety of potential attack vectors exist to compromise the TLM, the top of the trust hierarchy. Organizational and technical compromises pose a centralized threat to the entire ITS trust ecosystem. A variety of security controls are required to mitigate a TLM compromise and its potential to result in unauthorized and unofficial modifications to the ECTL. A compromised ECTL provides the means for an attacker to add or remove Root CAs and thus remove trust where it is needed and add unauthorized trust with malicious intent. Additionally, the TLM can undergo DoS attacks. Thus, the potential fallout from a TLM compromise can be the full or partial corruption of the entire trust ecosystem, as evidenced by end entities' inability to validate the trust of peers.
A compromise of the TLM can broadly take a variety of forms:
<b>Organizational –</b> Modifications or substitutions of TLM artefacts can be performed by individuals in either the TLM, CPOC during normal processing and interacting with Root CAs, or they can be performed during internal processing, generation and signing of the ECTL. Internal TLM individuals would likely be involved and can be motivated from external sources.
<b>Technical –</b> TLM systems can be compromised through a variety of methods ranging from entity/device spoofing, exploitation of internal personnel and networks, host interfaces, physical access, corruption of the ECTL generation and signing HSM, or corruption or delayed publication of the ECTL. Denial or substitution attacks on the publication endpoint(s) also pose a significant risk.</td></tr>
<tr><th>Gap/issue</th><th>Proposed mitigation with rationale</th></tr>
</table>

Vector: Rogue ECTL detection and reporting. **Description:** The TLM ECTL is issued by the TLM. Each can be issued as a full CTL or a delta CTL, and each with a CTL sequence. A compromised TLM can result in the ability for the attacker to generate a new CTL with untrusted or removed CAs.	The TLM European Root can need to include detection and reporting requirements for devices or device vendors to indicate the detection of a rogue ECTLs, i.e. rogue or unduly revoked Root CA certificates. A rogue ECTL can be detected in the following ways: 1) Two ECTLs that are not identical and indicate the same ctlSequence Number. 2) A full ECTL and delta ECTL with the same ctlSequence Number, in this case the delta ECTL, when differenced against the previously-generated full ECTL does not result in the new, full CTL. It can be necessary to perform this check periodically to validate the full succession of ECTLs and to verify an entity or device hasn't installed a rogue one in its past.
Vector: Integrity loss on ECTL input. **Description:** An individual at the Root CA, CPOC, or TLM substitutes the approved, self-signed Root CA certificate to be added to the ECTL with a rogue Root CA certificate. The impact is the introduction of an untrusted community of devices into the C-ITS ecosystem for the period of time between the publication of the unauthorized root certificate and its detection.	Broad controls that should be implemented in the PKI include the following: Beyond extensive technical and administrative controls designed to mitigate this type of attack, additional guarantees can need to be provided the ensure the new root certificate being added to the ECTL is the same as: 1) The legitimate root certificate that was generated in a signing ceremony at the Root authority. 2) The legitimate root certificate that was applied for, and approved by the CP authority. 3) The legitimate root certificate whose private key is contained in the audited Root CA HSM. The PKI should implement procedural controls to cross-verify the hash of the generated root CA certificate at: the Root CA to CP authority interface, CP authority to TLM, Root CA to CPOC interface, and CPOC to TLM interfaces. The level of consistency checking between PKI entities can be also be determined by assurance level.

Vector: Substitution or availability loss on ECTL output/publication. **Description:** The CPOC is responsible for publishing the ECTL to a public website. Conventional network and web attacks on this location can result in substitution of the ECTL or lack of its availability for device manufacturers, devices and the general subscriber base.	The following controls can mitigate this threat: — The TLM and CPOC validate the signature of the ECTL after signing and before publication. This is necessary to detect an error or other problem in the signing process. — Automated checks on the availability/reachability of the ECTL publication repository in the event of CPOC DoS. — Publication of the ECTL via independent pathways to multiple publication locations to mitigate integrity and DoS attacks. — Back-channel publication and distribution of the ECTL to device manufacturers. — Rapid notification capability to indicate to subscribers of a compromised or invalid ECTL. — Public as well as back-channel communication to industry stakeholders on: — Changes to the TLM certificate used to sign the ECTL. — ECTL publication timing, cadence and any unusual changes. — Root Authority validation of their root certificate entries in newly published ECTLs.

	PKI subscribers should perform the following: — Validate during ECTL download that the public ECTL publication location: — resolves to known domain name and uses DNSSEC. — website certificate validates. — Notify multiple PKI entities of any reachability or validation failures. — Implement a notification capability to device owner/operators of the compromised ECTL. — Include update mechanisms to rapidly update the ECTL in the event of a compromised TLM.

T.COMPROMISE_CERTIFICATE_AUTHORITY

Description: A CA includes a Root CA, intermediate CA, enrolment CA and authorization CA. A root CA can be compromised by an attacker using a combination of technical or process-related vulnerabilities. A compromise means that the attacker can issue certificates, including those for EAs, AA Sub-CAs and a variety of end entities. The resulting impact of a compromised CA is a compromise of all parts of the C-ITS ecosystem that cryptographically chain to that particular CA.	
Gap/issue	**Proposed mitigation with rationale**
Vector: Insider threats, general. **Description:** The European C-ITS CP mandates 5-year periodic background investigations that cannot fully address personnel risks, which are conditional and event-based. Security/trust compromises of safety-of-life systems have a higher sensitivity to insider threats.	**PKI:** CPSs conforming to the European CP should include augmented policies for operators of CA systems that service ITS-S which interact with safety-of-life-related CSNs. Background vetting should not only include periodic checks, but also include specific reporting criteria for high-sensitivity positions in the CA. Reporting criteria can include financial or other forms of personal distress, including substance abuse, that can materialize outside of routine background re-investigations and contribute to higher risks of personal compromise.
Vector: CA public key substitution attack during issuance. **Description:** CA certificates are generated and then signed by Root authorities or parent CAs. Prior to the parent signing the CA certificate, an attacker can substitute the new CA's generated public key with one for which it has the private key, or the attacker can back up private key material during the access event. The C-ITS CP requires multi-person and split-level control over HSM access but does not include details on how to split access or for specific functions, function sequences or other events. For example, one individual with authorization to generate a key pair does not need access to back it up or possibly generate the certificate signing request from it..	PKI implementers conforming to the European C-ITS CP should dictate down to the specific HSM function type and access the access splits and how the separation of duties is maintained by authorizing those splits to different individuals. Accordingly, the auditing of the performance of these actions needs to be detailed along with 1) how the audit function is independent of the individuals accessing the HSM, and 2) how it is not possibly to violate the integrity of the audit trail by those accessing the HSM. In addition, redundant and independent internal checks should be performed in the CA to validate that the generated CA public key is the same as that which ends up in the signed CA certificate.

Vector: Rogue CA detection and reporting **Description:** CTLs are issued by Root authorities. Each can be issued as a full CTL or a delta CTL, each with a CTL sequence. A compromised TLM or Root CA can result in the ability for the attacker to generate a new CTL with untrusted or removed CAs.	PKI participant policy should include detection and reporting requirements for devices or device vendors to indicate the detection of rogue CTLs, i.e. rogue or unduly revoked CA certificates. Rogue CTLs can be detected in the following ways: 1) Two CTLs that are not identical and indicate the same ctlSequence Number. 2) A full CTL and delta CTL with the same ctlSequence Number – in this case the delta CTL – when differenced against the previous full CTL does not result in the new, full CTL. Device vendors/subscribers should implement the ability in devices to log and report the detection of root certificates unknown to the device. This information should be audited and reported to the PKI periodically.

T.CERTIFICATE_WRONG_PERMISSIONS

Description: Allocation of permissions to special vehicles (e.g. police cars) occurs by the member state. There is no binding however between the allowable permissions that can be provisioned by a member state CA and authorizations granted to that CA. This opens a vector of attack in which a rogue insider within a Root or Sub-CA organization is able to modify the allowable permissions that can be granted to an end entity ITS-S. For example, a Root or Sub-CA that is not authorized to grant special permissions can do so if the policy management database within that CA is compromised. This can result in a CA being fooled into issuing certificates with wrong permissions or permissions to which the device was not entitled.	
Gap/issue	**Proposed mitigation with rationale**
Vector: Unchecked binding of allowable permission granting rights through the certificate chain **Description:** There is a threat against the integrity of the permissions issuance policies constraining what permissions a PKI service provider (CA) is authorized to grant to either a Sub-CA or end entity. There are assurance mechanisms in place for ensuring that the policy defining the allowable permissions to be issued by a CA are safeguarded.	Layered defensive policies need to be applied to the management of Root or Sub-CA permissions issuance policies. — The PKI should include a policy management database that defines the allowable permissions to be granted by each Root or Sub-CA. — Best practices should include: — two-person integrity controls to mitigate insider-based attacks on the integrity protections applied to the policy management database. — logging of all actions that modify the policy and monitoring of any actions associated with authorized users or their privileges.

T.EXPLOIT_BOOTSTRAP_ENROLMENT_PROCESSING

<table>
<tr><td colspan="2"><b>Description:</b> Registering, bootstrapping and enrolling a device into the ITS PKI introduces a variety of risks, both technical and operational. An exploit of the enrolment processing function would likely seek to a) introduce a malicious device into the trusted ecosystem, or b) introduce a device that trusts endpoints it should not.
Exploits on this processing can be performed at both the subscriber and EA.</td></tr>
<tr><th>Gap/issue</th><th>Proposed mitigation with rationale</th></tr>
<tr><td><b>Vector:</b> Compromise of the subscriber or manufacturers bootstrap process.
<b>Description:</b> The bootstrap process that precedes an enrolment action is often a weak security point in the processing of certificate requests and binding of public keys to certificates. Standard security mechanisms should be defined to ensure that the bootstrap process occurs in a secure manner by either the manufacturer or the operator of the ITS-S device.
An insecure bootstrap process can introduce opportunities for an attacker to submit a rogue ITS-S for certification.</td><td>Processes for secure enrolment of ITS-S should be documented or referenced as part of certification practices.
The following are options:
— The device manufacturer or operator develop a secure bootstrap process for enrolment of ITS-S and document process in CPS.
— The device manufacturer or operator establish a monitoring programme to regularly audit that rogue devices have not been submitted for enrolment.</td></tr>
<tr><td><b>Vector:</b> Compromised or malicious trust chain installed in ITS-S during boostrap.
<b>Description:</b> A pre-requisite to secure enrolment is maintaining the integrity of all the certificate trust chain information pertinent to the ITS-S. All root public keys, and no others, are present in the ECTL. Likewise, CA and Sub-CA keys are present in the CTL pertinent to the device's own root. If the process of installing the CTLs is integrity-violated, the device will trust malicious devices or be precluded from trusting devices it should.</td><td>Manufacturers or subscribers should implement the following:
— Validate all pertinent CTLs prior to installation in devices:
    — Verify the signature on each CTL and verify the root is present in the ECTL.
    — Ensure the hash of ECTL is the same as what is publicly published and what is received out-of-band from the TLM/CPOC.
    — Validate the cryptographic binding of the current CTL to the previous CTL.
— Implement plans to repeat the certificate trust chain validations upon each CTL refresh.
— Implement multi-person control and auditing on the CTL/ECTL installation on devices yet to be enrolled.</td></tr>
</table>

Vector: Compromise of the subscriber's enrolment request process following bootstrap. **Description:** Accidental or malicious manipulation of the subscriber's enrolment process is an insider threat problem. An attacker can attempt to enrol a malicious device or perform an enrolment that enables a device to obtain certificate types to which it is not entitled. For example, a vehicle OBU could obtain RSU certificates along with the communications rights those devices hold.	The organization requesting the enrolment needs to implement multi-person integrity in validating the pre-enrolment state of the ITS-S. This includes information such as hardware and firmware versions, device type, applicable AIDs and other device metadata. Note that the device and device type map to 1) a canonical public key for the device, and 2) authorizations that will be included in the device's subsequent authorization tickets. Therefore, multiple independent checks need to be made to ensure the device is appropriately categorized. Subsequently, a multi-person check is needed for the generation of the enrolment request that is to be submitted to the EA. This check should ensure that critical metadata such as device category is consistent with the validated device data ascertained prior to the enrolment request.
Vector: Compromise of the EA's enrolment processing. **Description:** Upon receiving an enrolment request, the EA internally categorizes the device against a policy that determines the type of certificates to which the device is entitled. This information is shared ahead of time from the device manufacturer or operator and maps to a canonical public key. Malicious or accidental false categorization of the device would enable the device operator to obtain ITS-S certificates to which it is not entitled.	Possible controls in the EA to limit the potential for false or malicious device-categorization include the following: — Implementing an automated, integrity-protected and audited process that ensures the device's enrolment request metadata is protected from reception through the enrolment process. — Implementing strict access controls that prohibit write or modify access on device metadata that is stored persistently and mapped to a device's canonical or enrolment public key. If modification is necessary, this should be an audited and multi-person function.

T.EXPLOIT_REVOCATION_PROCESSING

Description: Revocation processing for ITS-S enrolment credentials is a vital response in the case of a device security compromise or, in some cases, a serious malfunction. Revocation of enrolment certificates is initiated by a PKI subscriber and processed by the EA. Upon processing the subscriber's revocation request, the EA adds the revoked device's enrolment certificate to an internal blacklist, such that the device can no longer request authorization tickets.

An exploit of the revocation processing function would likely seek to accomplish one or both of the following:

1) Prevent or delay a compromised device from being revoked that should be revoked.

2) Revoke a device that should not be revoked.

Exploits on this processing can be performed at both the subscriber and EA.

In addition, a residual threat to the ITS ecosystem is the lack of revocation processing on authorization tickets. Even if a device's enrolment certificate is revoked, it can contain 1-week validity certificates up to a pre-loading period of 3 months. Thus, the threat imposed by the device will exist for up to 3 months, despite its enrolment revocation and short AT validity period of one week.

Gap/issue	Proposed mitigation with rationale

Vector: Compromise of subscriber's revocation requesting process. **Description:** A compromise of a subscriber, its enterprise and personnel can take many forms and an insider threat can exist to compromise the integrity of the revocation requesting processes.	Possible controls to mitigate this threat include the following: — Multi-person integrity on the revocation decision-making process, such that no single individual can decide that a device needs to be revoked. — Institution of requirements that minimize the amount of time between a revocation decision and a revocation request. — Multi-person integrity on the revocation requesting process to ensure that 1) the enrolment credential is the same as the one that should be revoked, and 2) a revocation action is not initiated on a device that should not be revoked. — Strict auditing by an independent person to ensure 1) revocation actions are performed in a short amount of time, and 2) multi-integrity processes are carried out.
Vector: Compromise of EA's blacklist process. **Description:** A compromise of an EA, its enterprise and personnel can take many forms and an insider threat can exist to compromise the integrity of the blacklisting process.	Possible controls to mitigate this threat include the following: — Multi-person integrity on the blacklisting process, such that no single individual implements the revocation decision-making and process. — Institution of requirements that minimize the amount of time between a received revocation decision and the inclusion of the revocation on the internal blacklist. — Multi-person integrity on the blacklisting process to ensure that 1) the enrolment credential is the same as the one that was requested to be revoked, and 2) a revocation action is not initiated on a device that should not be revoked. — Strict auditing by an independent person to ensure 1) revocation actions are performed in a short amount of time, 2) multi-integrity processes are carried out, 3) each blacklisting operation maps to a subscriber's revocation request, and 4) each revocation request was appropriately authenticated and validated. — Implementation of a plan to coordinate with the device subscriber on the allowance of that device to re-enrol.

Vector: Pre-loading period set too long. **Description:** A 3-month pre-loading period can allow a compromised and revoked device to continue to operate long after the device was revoked by the EA.	Possible controls to mitigate this threat include the following: — EA: — Implement a requirement that a device subscriber reports when a device has been taken offline following a revocation event. — Implement a requirement that a device subscriber is notified if a revoked device requests authorization tickets. — All vendor subscribers: — Implement a rapid notification process to owners of the device. — Implement terms and conditions into device purchase contracts that indicate the processes and expectations that device owners follow in the event of a revocation action. This is needed to ensure the device owner ceases operating the device and indicates this to the vendor. — Implement a policy to indicate a confirmation to the EA that the device was taken offline. — Roadside ITS-S vendor subscribers: — Enhance roadside ITS-S connectivity to minimize preloading periods whenever possible. While a maximum permissible value, three months is an excessive preload period in many cases.

T.COMPROMISE_CRYPTO

Description: Crypto agility is addressed at the policy level in the European C-ITS CP and addresses issues such as selected cryptographic algorithms and key lengths and their adaptation to the latest cryptanalytic attacks. Compromised cryptography, however, can occur at a more tactical level in the cryptographic algorithm design, its specification or its implementation in software or hardware. While an error in implementation can compromise the security of a single ITS vendor or device family's communications, an error in cryptographic algorithm design or specification can undermine the integrity of the entire ITS ecosystem. Additionally, traditional PKI deployments can, in the near future, be susceptible to attacks using quantum computers that render the cryptographic algorithms used within the PKI vulnerable. Successful use of a quantum computer to attack elliptic curve-based algorithms used within root and EE certificates could result in the ability for an attacker to create forged Sub-CAs and provision illegitimate EE certificates.	
Gap/issue	**Proposed mitigation with rationale**

Vector: Inadequate responsiveness to CP authority changes to algorithm or key length changes. **Description:** An ITS device vendor or PKI node can be unable to 1) respond to a sunsetting algorithm or key length, or 2) respond to an emergency adaptation of the cryptographic design.	Crypto agility is turned into a plan vs. remain a policy. All participants in the PKI should implement policies and practices to implement the following: — Establishing and maintaining an inventory of all cryptographic hardware and software in their systems, including HSMs, server operating systems, applications, test systems and backup systems. Include vendor information, software and hardware version numbers and the cryptographic algorithms and key lengths used in each. The inventory should also identify the available algorithms along with the modes and key lengths available in each implementation. One possible response to weak cryptography is a simple mandate to rollover to a new key length or algorithm mode within a software or firmware update. — Establish a response plan to various cryptographic changeover timelines, both fast and slow turnaround. Include backup strategies if certain device vendors or implementations will not be upgradable within the designated period(s) of time. — Establish and designate the actuation of a response plan to a person or team.
Vector: Discovery of algorithm implementation errors. **Description:** The ITS-S manufacturer or a software or hardware vendor who supplies to the ITS-S device manufacturer discovers an error or weakness in a cryptographic implementation within the supply chain.	Rapid notification and response planning are in place in the event a cryptographic implementation error is discovered: — ITS-S vendors should formulate and maintain a contractual obligation for their cryptographic suppliers to rapidly notify them of cryptographic or any other security vulnerabilities. — ITS-S vendors should also implement a plan to notify device owners and operators of the discovered vulnerability to facilitate a community software or firmware update procedure. — ITS-S vendors should develop a response plan that specifies a course of action should the vulnerability be in cryptographic hardware, and what the response should be if the hardware is upgradable or not. — ITS-S vendors and other subscribers should likewise institute a response plan to notify the PKI if the cryptographic errors affect the enrolment of any devices, thus impacting the security of the wider ITS-S. — ITS-S vendors and operators have in place a plan to map a discovered cryptographic vulnerability to a device and enrolment certificate, should a PKI response become necessary.

Vector: Step function made in quantum computing cryptanalytic attack capabilities introduces distrust in certificates issued to all PKI certificates. **Description:** A successful demonstration of an attack against ECDSA using quantum computers would require an update of algorithms used within the PKI and render existing processes vulnerable to spoofing. The PKI issues certificates that are vulnerable to quantum computing-based attacks. Additionally, this threat vector can include the creation of new illegitimate certificates and signing of information made to seem like it was generated in the past. Certificates that use existing EC-based algorithms are currently used to digitally sign communications. A quantum attack against ECDSA would allow new information to be created and signed using illegitimate certificates that seem like they are signed by the trust chain. Similar in concept to quantum harvesting attacks.	A move to quantum-resistant algorithms should be begun immediately. The integration of crypto algorithm agility mechanisms within the PKI would support the quick change-over to new algorithms when necessary, in the case that successful demonstration of a quantum-based attack against ECDSA is achieved sooner rather than later. Planning, including quantum-resistant algorithm selection, should be made in anticipation of a rapid, staged transition to support a quantum resistant PKI and subscriber base.

T.COMPROMISE_END_ENTITY_IDENTITY

<table>
<tr><td colspan="2"><b>Description:</b> Compromise of a device's identity introduces substantial risk to the ITS ecosystem. It enables a variety of masquerading attacks whereby the attacking device is allowed to use certificates to which it was never entitled. Likewise, compromise of a device's trust list enables attackers to perform many attacks on the device.</td></tr>
<tr><th>Gap/issue</th><th>Proposed mitigation with rationale</th></tr>
<tr><td><b>Vector:</b> Compromise of ITS-S private keys during initial enrolment.
<b>Description:</b> Extraction of a device's private keys, both enrolment certificates and authorization tickets can occur in both the enrolment and operational lifecycle phases of the device.</td><td>The ITS-S manufacturer or vendor maintains the security of all canonical keys throughout their lifecycle. This includes HSM storage, split access requirements, and strict access controls to key stores.
The ITS-S manufacturer or operator maintains a canonical private and public keypair associated with each device to enable initial device enrolment. Unauthorized access to the canonical private key could allow multiple devices, including malicious ones, to receive copies of the same canonical key. The EA implements checks to ensure that 1) there are no duplicate canonical certificates across its entire repository, and 2) there are no two initial enrolment requests performed using the same canonical private key. A response plan is developed for both of these scenarios.</td></tr>
</table>

Vector: Compromise of ITS-S private keys during operations **Description:** Extraction of a device's enrolment private key and authorization tickets can occur during the device's operational lifecycle phases. This can occur through chip capping side channel analysis and other techniques. The result is that unauthorized devices can gain all of the rights and permissions entitled to the compromised device.	From a PKI perspective, the following controls mitigate the threat: — The EA should track and monitor for erroneous or abnormal requests for authorization tickets. This can include duplicate requests, abnormal request timing and cadence as well as other patterns that indicate multiple devices can be using the same enrolment certificate. — Both the device vendor and EA should maintain a notification and response plan with one another to mitigate any discovered or suspected private key compromises.

T.CRL_DISSEMINATION_ISSUES

<table>
<tr><td colspan="2"><b>Description</b>: Without the timely distribution and receipt of CRL data, end entities cannot be assured of the trust associated with a PKI. Any DoS associated with a CRL distribution point or other technology used to distribute CRL data can impact this trust. For example, if the receiver of a message is not able to check the updated revocation status of the sender, then the receiver is unable to validate that the message is legitimate.</td></tr>
<tr><th>Gap/issue</th><th>Proposed mitigation with rationale</th></tr>
<tr><td><b>Vector:</b> DoS attack against infrastructure CRL distribution technologies.
<b>Description:</b> Excessive delay in the publication/dissemination of a CRL can result in the establishment of trusted operations with untrusted/potentially compromised hosts.</td><td>The availability of CRL information is assured. The following practices provide higher assurance repository availability.
— Establish redundant and geographically dispersed CRL distribution points.
— C-ITS application designers and standards bodies should define maximum allowable timelines for checking for fresh CRL information based on the AID and in some cases the specific message.</td></tr>
</table>

8.4 European C-ITS CP changes to support news C-ITS applications

8.4.1 General

Version 1.1 of the European C-ITS CP is bound to a narrow range of C-ITS applications supporting C-ITS cooperative awareness and infrastructure applications. Given the range of anticipated C-ITS application domains enabled through granular ISO/TS 21177 access controls and/or implemented in devices accommodating broader ranges of data sensitivity, this section provides specific references in the C-ITS CP needed to support such enhancements.

8.4.2 CP Section 1.6.1

Issue: This section mandates conformance of the C-ITS security architecture to the current definitions in ETSI TS 102 940 V1.3.1. This standard currently only lists the following application groups and their communications characteristics:

- Cooperative awareness
- Static local hazard warning
- Interactive local hazard warning

- Area hazard warning
- Advertised services
- Local high-speed unicast service
- Local multicast service
- Low-speed unicast service
- Distributed (networked) service
- Multiple applications.

Possible future action: Update references with the ability to include other application groups and types not currently defined in ETSE TS 102 940 V1.3.1.

8.4.3 CP Section 1.6.2

Issue: Amongst other restrictions, this section does not permit a certificate's authorized use to include any circumstances where the use of certificates could lead directly to death, personal injury, or severe environmental damage (such as the operation of nuclear facilities, aircraft navigation or communication, or weapons control systems, where failure could lead directly to death, personal injury, or severe environmental damage).

Unauthorized use of a high-sensitivity IDX device that is either embedded within, or fronts, a SCN can lead to significant personal injury or death.

Possible future action: Suggest re-wording this element of the CP to include such high-sensitivity devices, although under a higher assurance certification.

8.4.4 CP Section 6.1.5.2

Issue: This section indicates that "... security certificates for the cryptographic module shall be issued applying the common criteria certification scheme by a certification body recognized by the Management Committee within the framework of the 'Mutual Recognition Agreement of Information Technology Security Evaluation Certificates' of the Senior Officials Group on Information Systems Security (SOG-IS)."

Currently, this SOG-IS mutual agreement drives the V2X gateway and the HSM of the C-ITS-Ss to be security certified against V2X Gateway and HSM PPs developed by the Car2Car Communication Consortium.

Today, the majority of C-ITS device and application types and associated higher or lower data sensitivities are non-conforming and unsupported by the European C-ITS CP. In terms of information security management requirements specified in the C-ITS Policy Framework[28], only the following application messages are addressed: CAM[19], DENM[20], IVIM[2], SPaTEM, MAPEM, SSEM and SREM[22].

Possible future action: The European C-ITS CP should be updated to permit a wider set of PPs and associated certification regimes, each allocated to minimum trust assurance level. In other words, new C-ITS certification options should be supportable that responsibly and flexibly manage application domains against minimum trust assurance level(s) and the devices to which they pertain.

Additional PPS, conceivably supporting both lower and higher assurance levels, should be supportable with limitations and rules on how application domains (AIDs) of varying safety, security and privacy sensitivity can cohabitate specific ITS-S devices and their defined assurance levels.

While ITS-S devices support a trust assurance level, a framework that allows applications restricted to utilizing specific data types and sensitivities should also be mapped to minimum assurance levels.

8.4.5 CP Section 4.1.2.4

Issue: This section indicates the following constraints on ITS-SU:

- Regular vehicles have only one ITS-S that is registered at one EA.
- Requirements for 'special purpose vehicles' (anything not a regular vehicle) are dictated by member states. For example, they can be registered at an additional EA or have one additional ITS-S for authorizations that are in scope of the special purpose.

The challenge for ITS-S is that anything not designated as a regular vehicle ITS-S is designated as 'special vehicle' and regular vehicles are restricted to a single ITS-S.

Possible future action: The CP needs to be modified to modernize the types of ITS-S that can be offered on regular vehicles. This should include liberalization of the number and type of ITS AIDs that any single or multiple ITS-S in a vehicle can implement without becoming a member nation defined 'special vehicle.' In other words, accommodation of various commercial industry application domains less bound or unbounded to member nation definitions would permit new and emerging types of ITS-S outside of today's vehicle and RSU form factors. This can include a variety of diagnostics devices, specialized gateways and virtualized services/devices, in addition to multiple ITS-S being fronted by a single gateway ITS-S wholly or partially responsible for the accesses to the applications and devices behind it.

Annex A
(informative)

Scenario threats

Table A.1 depicts the full list of identified threats. For each threat the following are provided:

- the affected asset of the environment,
- a brief description the threat concerned,
- a listing of the type of threat actors likely involved,
- one or more attack vectors likely to be associated with the threat,
- possible motive,
- objective(s) of the attacker,
- desired outcome(s) of the attacker,
- a rough probability and impact level – Low, Medium, High,
- one or more associated security objectives or organizational policy(s) to counter the threat,
- a mapping to the data sensitivity scenario(s) to which each threat pertains

S1: IDX device handling only Scenario 1 data sensitivities.

S2: IDX device handling only Scenario 2 data sensitivities.

S3: IDX device handling only Scenario 3 data sensitivities

Table A.1 — Scenario-mapped threat table

Threat	Details	S1	S2	S3
T.THEFT	Asset: IDX device	x	x	x
	Area of concern: An attacker could try to gain illicit knowledge of the IDX design and thereby more easily mount an attack to compromise the integrity or authenticity the data processed by the device.			
	Actors: Disgruntled insider; stalker; hackers, taggers and script kiddies; criminal individual.			
	Attack vectors: Maintenance environment; internal system; authorized actions of non-privileged users; authorized actions of privileged users; device port; immediate physical proximity.			
	Motive: Notoriety; personal satisfaction; disgruntlement; positional/stepping stone.			
	Objectives: Enable other operations; extract data.			
	Outcome: Disclosure (identification of TOE vulnerabilities that can be exploited).			
	Probability: M; Impact: L			
Security objectives	OT_PLATFORM_AUTHENTICATION OT_PLATFORM_TAMPER_PROTECT			
Organizational policies	P. LOG_ASSET_INVENTORY P.PHYSICAL_ACCESS P.PHYSICAL_MONITORING			

Table A.1 *(continued)*

T.PHYSICAL_TAMPER	Asset: IDX device	x	x	x
	Area of Concern: An attacker could attempt to access the internal components of the IDX device to bypass software security controls and extract data including firmware which could lead to exposure of default passwords and other information.			
	Actor: Disgruntled insider; stalker; hackers, taggers and script kiddies; criminal individual			
	Attack vectors: Maintenance environment; internal system; authorized actions of non-privileged users; authorized actions of privileged users; device port; immediate physical proximity.			
	Motive: Notoriety; personal satisfaction; disgruntlement; positional/stepping stone.			
	Outcome: Disclosure (identification of TOE vulnerabilities that can be able exploited or access to sensitive information stored within the device).			
	Probability: M; Impact: L			
Security Objectives	OT_PLATFORM_TAMPER_PROTECT OT_PLATFORM_LOG			
Organizational policies	P.PHYSICAL_ACCESS: The environment that a IDX device is used or stored within does not allow unauthorized users without proper monitoring/escort. P.PHYSICAL_MONITORING: The environment that a TOE is used or stored within will be monitored for unauthorized entry after hours (see Reference [33]).			
T.INTERFACE_EXPLOITATION	Asset: IDX		x	x
	Area of concern: Attacker is able to gain access to an IDX device through misconfigured or insecure physical or logical interfaces.			
	Actor: Criminal individual; hackers, taggers and script kiddies.			
	Attack vectors: External network connection; external shared or infrastructure services; rusted or Paprtner network connection; mobile or transiently connected devices; device port.			
	Motive: Personal satisfaction; personal financial gain; unpredictable; positional/stepping stone; tracking/stalking.			
	Outcome: Unauthorized use; disclosure.			
	Probability: L; Impact: L			
Security objectives	OT_PLATFORM_COMM_INTERFACE_PROTECT OT_PLATFORM_PHYSICAL_INTERFACE_PROTECT OT_PLATFORM_LOG OT_PLATFORM_REMOTE_CONNECTION_PROTECT			
T.UNAUTHORIZED_LOCAL_TRUST_CHAIN_MODIFICATION	Asset: IDX trust anchors.	x	x	x
	Area of concern: An attacker is able to gain write access to the IDX device trust store, resulting in the ability to delete existing root or other CA certificates that would cause message validation failure or to add new Root/CA certificates that would introduce unexpected and unauthorized trust relationships.			
	Actor: Hackers, taggers and script kiddies; criminal individual; insider.			
	Attack vectors: Supply chain; maintenance environment; trusted or partner network connection; authorized actions of privileged user.			
	Motive: Financial gain; positional/stepping stone.			
	Outcome: Modification; unauthorized use.			
	Probability: L; Impact: L			
Security objectives	OT_PLATFORM_TRUST_CHAIN_PROTECT OT_PLATFORM_LOG			

Table A.1 *(continued)*

Organizational policies	P.PHYSICAL_ACCESS: The environment that a TOE is used or stored within does not allow unauthorized users without proper monitoring/escort. P.PHYSICAL_MONITORING: The environment that a TOE is used or stored within will be monitored for unauthorized entry after hours. P.SECURE_MANAGEMENT: The TOE provides management means for the authorized administrator to manage the IDX device in a secure manner.			
T.ENVIRONMENT_CA_COMPROMISE	Asset: Cloud service trust anchors	x	x	x
	Area of concern: An attacker is able to compromise one of the SCMS CAs resulting in illegitimate issuance of trusted certificates that allow vehicle ITS-SCN and IDX devices to spoof legitimate system components.			
	Actor: Insider; former insider; criminal individual; nation-state-aligned professional.			
	Attack vectors: Supply chain; privileged user.			
	Motive: Ideology; organizational gain; disgruntlement; unpredictable; positional/stepping stone.			
	Outcome: Interruption; degradation; unauthorized use.			
	Probability: L; Impact: M			
Security objectives	OT_PROV_KEY_MGMT_AUTHORITY_REVOCATION OT_PROV_KEY_MGMT_AUTHORITY_MISBEHAVIOUR_DETECTION			
Organizational policies	P.SECURE_MANAGEMENT: The TOE provides management means for the authorized administrator to manage the IDX device in a secure manner.			
T.ENVIRONMENT_UNREPORTED_MISBEHAVIOUR_MALFUNCTION	Asset: Revocation information.	x	x	x
	Area of concern: Misbehaving components are not reported in a timely manner for revocation action, resulting in the ability for a compromised service device or vehicle to establish trusted relationships with other components until the revocation process is completed.			
	Actor: Insider; former insider; hackers, taggers and script kiddies.			
	Attack vectors: Supply chain; maintenance environment; privileged user; trusted or partner network.			
	Motive: Financial gain; organizational gain; disgruntlement; positional/stepping stone.			
	Outcome: Disclosure; loss/exfiltration; unauthorized use.			
	Probability: L; Impact: L			
Security objectives	OT_PLATFORM_CERT_REVOCATION OT_APP_CERT_REVOCATION			
Organizational policies	P.MISBEHAVIOUR_REVOCATION_REPORTING: Policies and procudures to 1) detect misbehaviour, 2) obtain misbehaviour metadata, 3) report misbehaviour and, as needed, 4) request misbehaviour-based revocation of devices are implemented.			

Table A.1 *(continued)*

T.PRIVATE_KEY_DISCLOSURE	Asset: Application encryption private keys; IDX application private keys; cloud services TLS private keys. Note that this threat is not indicated for Scenario 1 data sensitivities. However, it is possible for disclosure of private key material to be a prerequisite to targeting resource exhaustion of the ITS-S. If resource exhaustion is a threat to a design and implementation, then disclosure of private key material should be included for Scenario 1.		x	x
	Area of concern: An attacker is able to obtain the secret private keys for authentication of the IDX device, allowing the attacker to masquerade as the service device and perform trusted functions with the ITS-SCN.			
	Actor: Insider; former insider; hackers, taggers and script kiddies			
	Attack ectors: Supply chain; maintenance environment; internal network; authorized actions of privileged users.			
	Motive: Organizational gain; financial gain; accidental; positional/stepping stone; tracking/stalking.			
	Outcome: Dislosure; modification; destruction; loss/exfiltration; unauthorized use.			
	Probability: L; Impact: L			
Security objectives	OT_PLATFORM_SECURE_KEY_MGMT OT_APP_SECURE_KEY_MGMT			
T.EPHEMERAL_OR_SESSION_KEY_DISCLOSURE	Asset: TLS session keys		x	x
	Area of concern: An attacker is able to intercept the session keys generated using ECIES resulting in a loss of confidentiality for data exchanges between the IDX device and ITS-SCN.			
	Actor: Hackers, taggers and script kiddies; criminal individual.			
	Attack Vectors: ?			
	Motive: Personal satisfaction; unpredictable; positional/stepping stone; tracking/stalking.			
	Outcome: Disclosure; loss/exfiltration			
	Probability: L; Impact: L			
Security objectives	OT_PLATFORM_SECURE_KEY_MGMT OT_APP_SECURE_KEY_MGMT			
T.UNAUTHORIZED_PRIVATE_KEY_USE	Asset: Application encryption private keys; IDX application private keys; cloud services TLS private keys. This threat is not indicated for Scenario 1 data sensitivities. However, it is possible for unauthorized private key use to be a prerequisite to targeting resource exhaustion of the ITS-S. If resource exhaustion is a threat to a design and implementation, then disclosure of private key material should be included for Scenario 1.		x	x
	Area of Concern: An unauthorized user is able to gain access to an unprotected private key and use the key for messaging and transactions between the IDX device and ITS-SCN, allowing an unauthorized user to perform potentially privileged transactions or masquerade as authorized system components.			
	Actor: Insider; former insider; hackers, taggers and script kiddies; stalker.			
	Attack Vectors: Supply chain; maintenance environment; internal network; authorized actions of privileged users.			
	Motive: Organizational gain; financial gain; accidental; positional/stepping stone; tracking/stalking.			
	Outcome: Dislosure; modification; destruction; loss/exfiltration; unauthorized use.			
	Probability: L; Impact: L			
Security objectives	OT_PLATFORM_SECURE_KEY_MGMT OT_APP_SECURE_KEY_MGMT			

Table A.1 *(continued)*

Organizational policies	P.MISBEHAVIOUR_REVOCATION_REPORTING: Policies and procedures to 1) detect misbehaviour, 2) obtain misbehaviour metadata, 3) report misbehaviour and, as needed, 4) request misbehaviour-based revocation of devices are implemented.			
T.SOFTWARE_FIRMWARE_TAMPER_IN_TRANSIT	Asset: IDX device firmware		x	x
	Area of concern: An attacker could alter the firmware of the IDX device during or prior to the update process to introduce malicious functionality or backdoors or identify ways to take root permissions on the IDX device. In this way, an attacker achieves potentially privileged operation of the IDX device.			
	Actor: Criminal individual; hackers, taggers and script kiddies.			
	Attack vectors: Maintenance environment; external shared or infrastructure services; trusted or partner network; authorized actions of privileged user.			
	Motive: Ideology; notoriety; personal satisfaction; organizational gain; disgruntlement; positional/stepping stone.			
	Outcome: Modification; destruction; interruption; degradation.			
	Probability: L; Impact: M			
Security objectives	OT_EXT_FW_PROTECT			
T.SOFTWARE_FIRMWARE_TAMPER_AT_REST	Asset: IDX		x	x
	Area of Concern: An attacker could gain access to IDX device firmware by extracting from the device and then alter the firmware to a malicious one or identify ways to take root permissions on the device. In this way an attacker achieves potentially privielged operation of the IDX device.			
	Actor: Criminal individual; hackers, taggers and script kiddies.			
	Attack vectors: Maintenance environment; external shared or infrastructure services; trusted or partner network; authorized actions of privileged user.			
	Motive: Ideology; notoriety; personal satisfaction; organizational gain; disgruntlement; positional/stepping stone.			
	Outcome: Modification; destruction; interruption; degradation.			
	Probability: L; Impact: M			
Security objectives	OT_PLATFORM_USER_MGMT OT_PLATFORM_DATA_AT_REST_PROTECT OT_EXT_FW_PROTECT			
Organizational policies	P.PHYSICAL_ACCESS: The environment that a TOE is used or stored within does not allow unauthorized users without proper monitoring/escort. P.PHYSICAL_MONITORING: The environment that a TOE is used or stored within will be monitored for unauthorized entry after hours.			
T.SOFTWARE_FIRMWARE_INTEGRITY_ERROR	Asset: IDX firmware.	x	x	x
	Area of concern: IDX firmware validation results in an integrity error, disrupting the operations and availabiltiy of the IDX device.			
	Actor: System.			
	Attack vectors: Supply chain.			
	Motive: Accidental.			
	Outcome: Interruption; degradation.			
	Probability: M; Impact: L			
Security objectives	OT_EXT_FW_PROTECT			

Table A.1 *(continued)*

T.UNAUTHORIZED_ACCESS_POLICY_MODIFICATION	Asset: IDX access control policies.		x	x
	Area of concern: An attacker is able to obtain write access to the IDX device access policy configurations, resulting in the ability for the attacker to grant him/herself permissions for performing sensitive functions.			
	Actor: Insider; former insider; hackers, taggers and script kiddies.			
	Attack vectors: Maintenance environment; trusted or partner network connection; device port; privileged user.			
	Motive: Personal satisfaction; disgruntlement; positional/stepping stone; tracking/stalking.			
	Outcome: Modification; disclosure; interruption; degradaton; unauthorized use.			
	Probability: M; Impact: L			
Security objectives	OT_APP_AC_POLICY_PROTECT			
Organizational policies	P.SECURE_MANAGEMENT: The TOE provides management means for the authorized administrator to manage the IDX device in a secure manner.			
T.LACK_OF_REVOCATION_INFORMATION_AVAILABILITY	Asset: Revocation information	x	x	x
	Area of concern: The IDX device is unable to retrieve the latest revocation information, resulting in the device potentially trusting compromised/revoked certificates until revocation information is available.			
	Actor: System; insider			
	Attack vectors: System error; DoS; misconfiguration (intentional or accidental).			
	Motive: Accidental; positional/stepping stone.			
	Outcome: Interruption; degradation.			
	Probability: M; Impact: L.			
Security objectives	OT_PLATFORM_ALERT_INFORM OT_PLATFORM_RELIABLE_TIME			
Organizational policies	P.SECURE_MANAGEMENT: The TOE provides management means for the authorized administrator to manage the IDX device in a secure manner.			
T.PRIVILEGE_ESCALATION	Asset: IDX user account data; IDX application and data access control policy(s); user passwords and authenticators		x	x
	Area of concern: A logged in user to the IDX device is able to escalate privileges on the device to perform functions that were not authorized for his/her role.			
	Actor: Insider; former insider; hackers, taggers and script kiddies.			
	Attack vectors: Authorized actions of non-privileged users; maintainer.			
	Motive: Personal satisfaction; organizational gain; financial gain; accidental; tracking/stalking.			
	Outcome: Disclosure; modification; loss/exfiltration; unauthorized use.			
	Probability: M; Impact: L			
Security objectives	OT_PLATFORM_PRIVILEGED_USER_AUTHORIZATION			
T.UNAUTHORIZED_ACCOUNT_ASSUMPTION	Asset: IDX user account ata; user passwords and authenticators.		x	x
	Area of concern: Users could try to access functions not allowed to them when the previous user has not properly logged out, to modify, delete user data, download sensitive ITS-SCN information or modify downloaded ITS-SCN data.			
	Actor: Insider.			
	Attack ectors: Privileged user; normal user.			
	Motive: Personal satisfaction; disgruntlement; accidental; tracking/ stalking; unpredictable.			
	Outcome: Disclosure; modification; unauthorized use.			
	Probability: M; Impact: L			

Table A.1 *(continued)*

Security objectives	OT_PLATFORM_SESSION_PROTECT OT_PLATFORM_RELIABLE_TIME			
Organizational policies	P.SECURE_MANAGEMENT: The TOE provides management means for the authorized administrator to manage the IDX device in a secure manner.			
T.EAVESDROPPING	Asset: Data.		x	x
	Area of concern: An attacker operating within the range of the wireless communications between the IDX device and ITS-SCN is able to tap into the communications and decode messaging resulting in a loss of confidentiality.			
	Actor: Criminal individual; hackers, taggers and script kiddies; stalker.			
	Attack vectors: Internal network; mobile or transiently connected devices.			
	Motive: Tracking/stalking; personal satisfaction; positional/stepping stone.			
	Outcome: Disclosure; loss/exfiltration.			
	Probability: M; Impact: L			
Security objectives	OT_PLATFORM_COMM_INTERFACE_PROTECT			
T.ENIVIRONMENT_NETWORK_ATTACKER_IN_THE_MIDDLE	Asset: Configuration files.		x	x
	Area of concern: A successful modification of data exchanged between the IDX device and ITS-SCN would allow an attacker to view or change configurations, upload new software, or download sensitive data.			
	Actor: Criminal individual; hackers, taggers and script kiddies; stalker.			
	Attack vectors: Internal network; mobile or transiently connected devices.			
	Motive: Tracking/stalking; personal satisfaction; positional/stepping stone.			
	Outcome: Modification; destruction; Interruption; degradation.			
	Probability: L; Impact: H			
Security objectives	OT_PLATFORM_COMM_INTERFACE_PROTECT			
T.ENVIRONMENT_SCN_SPOOFING	Asset: ITS-SCN firmware, software and applications.		x	x
	Area of Concern: SCN is masquerading as a genuine ITS-SCN sending false information in messages that are otherwise valid.			
	Actor: Criminal Individual; Hackers, taggers and script kiddies			
	Attack vectors; Supply chain; mobile or transiently connected devices; developer/integrator; maintainer; external actor.			
	Motive: Notoriety; personal satisfaction; unpredictable; positional/stepping stone.			
	Outcome: Uncertainty.			
	Probability: L; Impact: L			
Security objectives	OT_APP_VALIDATE_PEER_STATE: Device can validate the current peer's security state through attestation transaction prior to loading or retrieving sensitive data. OT_EXT_FW_PROTECT			