ETSI TS 118 116 V4.0.1 (2023-11)

oneM2M;
Secure Environment Abstraction
(oneM2M TS-0016 version 4.0.1 Release 4)

Reference
RTS/oneM2M-000016v4

Keywords
IoT, M2M, security

ETSI

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

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

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights 6
Foreword 6
1 Scope 7
2 References 7
2.1 Normative references 7
2.2 Informative references 8
3 Definition of terms, symbols and abbreviations 8
3.1 Terms 8
3.2 Symbols 8
3.3 Abbreviations 8
4 Conventions 9
5 SE Abstraction Architecture 9
5.1 Introduction 9
5.2 AE and CSE access security services within the SE 10
5.3 AE residing within the SE 11
6 Secure Environments 12
6.1 Secure Environments capabilities 12
6.2 Secure Environments security levels 12
6.3 Tamper resistant hardware SE implementation 12
6.4 Hardware isolated SE implementation 13
6.5 Software based SE implementation 13
7 Logical Abstraction - McsReference point 13
7.1 Overview 13
7.2 Mcs reference point 13
7.2.1 Secure Environment Identifier (M2M-SE-ID) 13
7.2.2 Differences between Mcs and Mcc/Mca reference points 14
7.2.3 Namespaces used for resource and data types 14
7.2.4 Mcs Resource type definitions 15
7.3 Resource SE 15
7.3.0 Overview 15
7.3.1 Resource *SEReboot* 17
7.4 Sensitive Data Storage 18
7.4.1 <sensitiveDataObject> resource 18
7.4.2 <sensitiveDataObject> Resource Procedures 20
7.4.2.1 CREATE <sensitiveDataObject> 20
7.4.2.2 RETRIEVE <sensitiveDataObject> 20
7.4.2.3 UPDATE <sensitiveDataObject> 21
7.4.2.4 DELETE <sensitiveDataObject> 21
7.5 Sensitive Cryptographic Functions 21
7.5.1 <cipher> resource 21
7.5.1.0 Introduction 21
7.5.1.1 <cipher> Resource Procedures 23
7.5.1.1.1 CREATE <cipher> 23
7.5.1.1.2 RETRIEVE <cipher> 24
7.5.1.1.3 UPDATE <cipher> 24
7.5.1.1.4 DELETE <cipher> 25
7.5.1.2 <encrypt> Resource 25
7.5.1.3 <decrypt> Resource 25
7.5.1.4 <generateKey> Resource 25
7.5.1.5 <algorithmSpecificParameter> Resource 25
7.5.2 <rand> resource 27
7.5.2.0 Introduction 27
7.5.2.1 <rand> Resource Procedures 28

7.5.2.1.1 CREATE <rand> 28
7.5.2.1.2 RETRIEVE <rand> 28
7.5.2.1.3 UPDATE <rand> 29
7.5.2.1.4 DELETE <rand> 29
7.5.2.2 <generateRand> Resource 29
7.5.3 <hash> resource 29
7.5.3.0 Introduction 29
7.5.3.1 <hash> Resource Procedures 31
7.5.3.1.1 CREATE <hash> 31
7.5.3.1.2 RETRIEVE <hash> 31
7.5.3.1.3 UPDATE <hash> 32
7.5.3.1.4 DELETE <hash> 32
7.5.3.2 <calculateHash> Resource 32
7.5.4 <signature> resource 33
7.5.4.0 Introduction 33
7.5.4.1 <signature> Resource Procedures 35
7.5.4.1.1 CREATE <signature> 35
7.5.4.1.2 RETRIEVE <signature> 35
7.5.4.1.3 UPDATE <signature> 36
7.5.4.1.4 DELETE <signature> 36
7.5.4.2 <calculateSignature> Resource 36
7.5.4.3 <verifySignature> Resource 36
7.5.4.4 <generateKey> Resource 37
7.6 Secure Connection Establishment 37
7.6.1 <secureConnection> resource 37
7.6.2 <secureConnection> Resource Procedures 38
7.6.2.1 CREATE <secureConnection> 38
7.6.2.2 RETRIEVE <secureConnection> 39
7.6.2.3 UPDATE <secureConnection> 39
7.6.2.4 DELETE <secureConnection> 40
7.6.3 <connectionInstance> resource 40
7.6.4 <connectionInstance> Resource Procedures 41
7.6.4.1 CREATE <connectionInstance> 41
7.6.4.2 RETRIEVE <connectionInstance> 42
7.6.4.3 UPDATE <connectionInstance> 42
7.6.4.4 DELETE <connectionInstance> 43
7.6.5 <connect> Resource 43
7.6.6 <send> Resource 43
7.6.7 <generateKey> Resource 43
7.7 Authentication and Identification 43
7.7.1 <identity> resource 43
7.7.2 <identity> Resource Procedures 45
7.7.2.1 CREATE <identity> 45
7.7.2.2 RETRIEVE <identity> 45
7.7.2.3 UPDATE <identity> 46
7.7.2.4 DELETE <identity> 46
7.7.3 <authenticate> Resource 46
7.7.4 <generateKey> Resource 46

8 Physical Interface 47

9 Resource type definitions for the Mcs reference point 47
9.1 Mcs specific enumeration values of m2m:resourceType 47
9.2 senv:SEType 47
9.3 senv:securityLevel 47
9.4 senv:rebootType 48
9.5 senv:cipherLabel 48
9.6 senv:cipherAlgorithm 48
9.7 senv:rngType 48
9.8 senv:hashAlgorithm 48
9.9 senv:signatureAlgorithm 49
9.10 senv:connectionTypeID 49

10 Short Name definitions for the Mcs reference point 49
10.1 Short Names for Mcs specific resource attributes 49
10.2 Short Names for Mcs specific resource types....50

History 52

Intellectual Property Rights

Essential patents

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

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

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Partnership Project oneM2M (oneM2M).

1 Scope

The present document specifies mechanisms and interfaces to abstract from different technical implementations of a secure environment as defined in ETSI TS 118 103 [1].

2 References

2.1 Normative references

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

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

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

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

[1] ETSI TS 118 103: "oneM2M; Security solutions (oneM2M TS-0003)".

[2] ETSI TS 118 101: "oneM2M; Functional Architecture (oneM2M TS-0001)".

[3] ETSI TS 102 221: "Smart Cards; UICC-Terminal interface; Physical and logical characteristics".

[4] GlobalPlatform: "Device Technology Device API Access Control", v1.0.

[5] GlobalPlatform: "Card Specification", version 2.3 (including its Amendments).

[6] IETF RFC 5705: "Keying Material Exporters for Transport Layer Security (TLS)".

[7] ISO/IEC 7816-3: "Identification cards -- Integrated circuit cards -- Part 3: Cards with contacts -- Electrical interface and transmission protocols".

[8] IETF RFC 5116: "An Interface and Algorithms for Authenticated Encryption".

[9] IETF RFC 6655: "AES-CCM Cipher Suites for Transport Layer Security (TLS)".

[10] ISO 9797 (2011): "Information technology -- Security techniques -- Message Authentication Codes (MACs)".

[11] NIST FIPS PUB 186-4: "Digital Signature Standard (DSS)".

[12] IETF RFC 2104: "HMAC: Keyed-Hashing for Message Authentication".

[13] ISO/IEC 18031:2011: "Information technology -- Security techniques -- Random bit generation".

[14] ETSI TS 118 104: "oneM2M; Service Layer Core Protocol Specification (oneM2M TS-0004)".

[15] ETSI TS 118 108: "oneM2M; CoAP Protocol Binding (oneM2M TS-0008)".

[16] ETSI TS 118 109: "oneM2M; HTTP Protocol Binding (oneM2M TS-0009)".

[17] ETSI TS 118 110: "oneM2M; MQTT Protocol Binding (oneM2M TS-0010)".

[18] ETSI TS 118 120: "oneM2M; WebSocket Protocol Binding (oneM2M TS-0020)".

[19] ETSI TS 118 122: "oneM2M; Field Device Configuration (oneM2M TS-0022)".

[20] ETSI TS 118 111: "oneM2M; Common Terminology (oneM2M TS-0011)".

2.2 Informative references

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

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

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

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

[i.1] oneM2M Drafting Rules.

[i.2] ETSI TR 118 508: "oneM2M; Security (oneM2M TR-0008)".

[i.3] ISO 7816-4: "Identification cards -- Integrated circuit cards -- Part 4: Organization, security and commands for interchange".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

SE-resource: resource that resides within the Secure Environment and that does not have a representation within an external CSE

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TS 118 111 [20] and the following apply:

AE	Application Entity
AEAD	Authenticated Encryption with Associated Data
AE-ID	Application Entity Identifier
AES	Advanced Encryption Standard
API	Application Programming Interface
CBC	Cipher Block Chaining
CCM	Counter with CBC-MAC
CMAC	Cipher-based Message Authentication Code
CSE	Common Services Entity
CSE-ID	Common Services Entity Identifier
DTLS	Datagram Transport Layer Security
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standard
GCM	Galois Counter Mode
HMAC	keyed-Hash Message Authentication Code
IANA	Internet Assigned Numbers Authority
ID	Identity
IV	Initialization Vector
LEN	Length
MAC	Message Authentication Code

NIST	National Institute of Standards and Technology
PKCS	Public Key Cryptography Standards
RFU	Reserved for Future Use
RNG	Random Number Generator
RO	Read-Only
RW	Read-Write
SE	Secure Environment
SEC	Security
SMS	Short Message Service
TEE	Trusted Execution Environment
TLS	Transport Layer Security
UICC	Universal Integrated Circuit Card
URI	Uniform Resource Identifier
WO	Write-Only

4 Conventions

The key words "Shall", "Shall not", "May", "Need not", "Should", "Should not" in the present document are to be interpreted as described in the oneM2M Drafting Rules [i.1].

5 SE Abstraction Architecture

5.1 Introduction

As defined in ETSI TS 118 103 [1], a Secure Environment (SE) provides protected sensitive functions and sensitive data to entities within the M2M system via the Mcs reference point. It serves the purpose of protecting secret or sensitive information (code or data) at rest and in use (i.e. while being used in computing processes). An SE is either implemented on a dedicated hardware component or on a trusted logical entity represented by a set of software functions on the supporting M2M node. An SE shall provide process isolation with respect to code and data residing outside of the SE.

In most M2M ecosystems, multiple stakeholders that do not necessarily trust each other (e.g. Underlying network operator, M2M Service Provider, M2M application provider and end user) need to protect their own sensitive data and functions, M2M nodes should therefore support multiple secure environments that shall provide process isolation between each other. Depending on deployment models, secure environments may be either pre-provisioned before deployment, or created dynamically during the enrolment phase, relying on SE management functionalities provided by the SE Abstraction Layer specified in the present document.

Depending on risk assessment specific to the use case and its associated security requirements several different integration scenarios are possible. They are described within this clause.

Regardless of the underlying instantiation(s) of secure environments implemented on an M2M node, the capability to create, personalize and manage secure environment areas shall be exposed by the local CSE to local AEs via the SE Abstraction Layer, as detailed in the present document. Furthermore, the local CSE itself shall use the locally available secure environment capabilities to protect sensitive information (see ETSI TS 118 103 [1]).

In general the following high level architecture as depicted in figure 5.1-1 applies. However AEs and CSEs may be spread between different processing environments within a node, including security sensitive parts running in local secure environments. The SE Abstraction Layer exposes over Mcs a common security interface to AEs and CSEs components within devices, facilitating independent deployment and management of such components in heterogeneous scenarios.

When an Mcs upstream API is exposed to a oneM2M entity, the CSE components shall rely on secure environment capabilities exposed over Mcs to implement their security services. The Mcs API enables a CSE to implement high level security services exposed on Mcc or Mca by relying on lower level services exposed to the SE Abstraction Layer by locally available secure environment implementations. Bindings of the Mcs functionalities to specific SE implementations can be specified by other organizations or provided as annex to the present document.

Additionally, CSEs may rely on CSE_sec components running inside the secure environment to expose additional optional capabilities through the Mcs layer, to expose further domain specific functionalities over Mca or Mcc. Such extensions are not specified in the present document.

Similarly, AEs may rely on AE_sec components running inside the secure environment to expose additional application specific capabilities to the Mcs layer. Such services are application specific and therefore not specified by oneM2M.

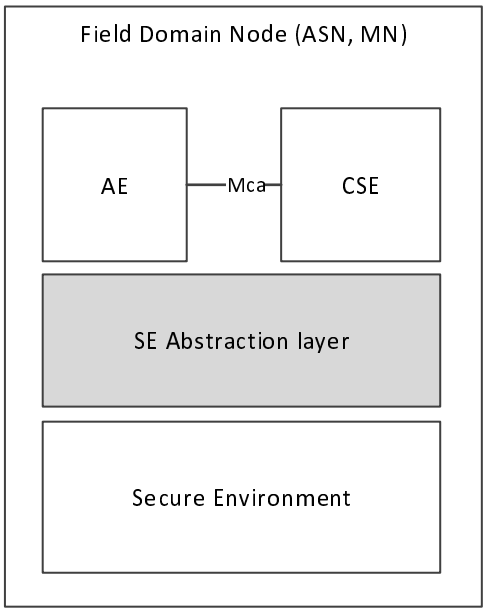

Figure 5.1-1: Secure Environment interworking on Field Domain Node

5.2 AE and CSE access security services within the SE

In this scenario, both the AE and the CSE reside within a Node as depicted in figure 5.2-1. The AE (or CSE) is split into a secure and a non secure part whereas the security relevant part AE_sec (respectively CSE_sec) resides within the SE and the corresponding non security relevant part AE_ (respectively CSE_) resides within the application space of the node. The AE_ (respectively CSE_ accesses AE_sec (respectively CSE_sec) via the Maa reference point. In addition the AE and the CSE can access security services offered by the SE via the Mcs reference point.

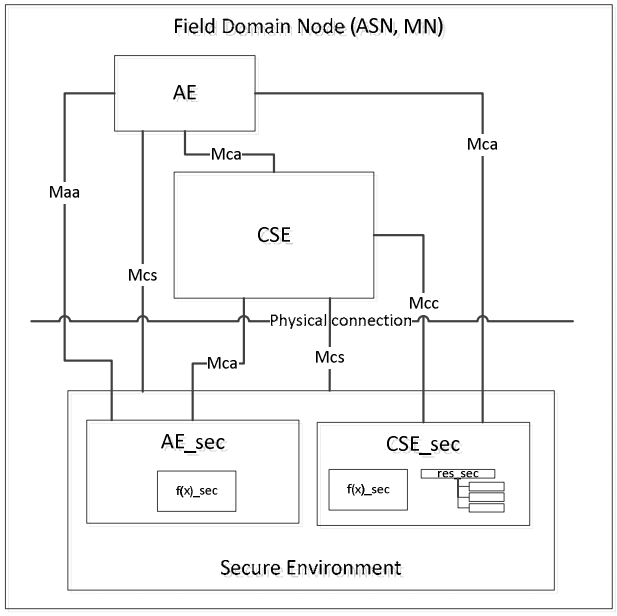

Figure 5.2-1: Secure Environment architecture of a Field Domain Node

The SE is integrated into the node as follows:

- Logically: the SE offers its sensitive functions f(x)_sec and its security resources res_sec to AEs and CSEs residing within the field domain node via the Mcs reference point. In addition the AE_sec or CSE_sec may offer corresponding services to the AE_ or CSE_ via a proprietary Maa reference point. AE_sec may also access the CSE via the Mca reference point.
- Physically: in case the SE is a dedicated hardware component, it has to be integrated into the node physically including low level drivers that enable logical access to the SE. The physical connection is superfluous in case the SE is implemented in software.

5.3 AE residing within the SE

In this scenario, the entire AE resides within the SE and utilizes security services provided by the SE. In addition the CSE may access the SE for dedicated security services via the Mcs reference point as depicted in figure 5.3-1. The AE may additionally access CSE resources via the Mca reference point.

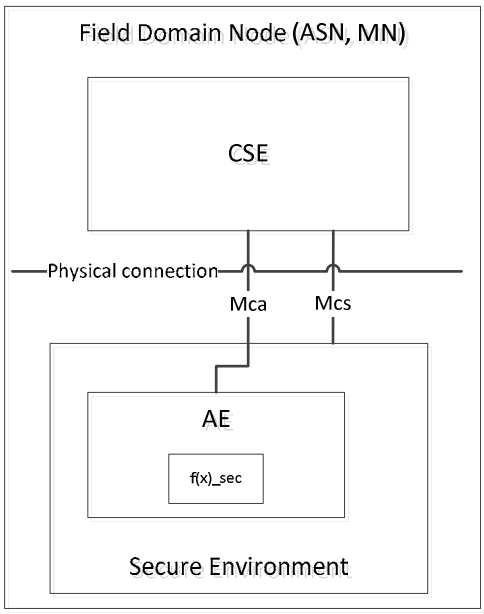

Figure 5.3-1: Secure Environment architecture in a Field Domain Node

The SE is integrated into the node as follows:

- Logically: the SE offers sensitive functions f(x)_sec and security resources res_sec to CSEs residing within the field domain node via the Mcs reference point. The AE is integrated within the SE such that it uses SE internal interfaces and methods.
- Physically: in case the SE is a dedicated hardware component, it has to be integrated into the field domain node physically including low level drivers that enable logical access to the SE. The physical connection is superfluous in case the SE is implemented in software.

6 Secure Environments

6.1 Secure Environments capabilities

A Secure Environment is an abstraction of a secure area within a computing system that provides a defined level of protection for code and data at rest, i.e. in storage, and in use, i.e. during process execution or data manipulation. A Secure Environment shall provide an authenticated entity (e.g. M2M Service Provider, M2M application provider or end user) with exclusive access to manage an isolated area of process space and memory within the host node that provides confidentiality and integrity of the contained instructions code and data during storage and process execution.

Each M2M node containing a CSE shall support instantiation of at least one SE by pre-provisioning, and may preferably support instantiation of SEs remotely on the field by means specified below. An SE shall be uniquely identifiable within a node and shall provide indication of the security level associated with the implementation it relies on.

6.2 Secure Environments security levels

According to ETSI TS 118 103 [1], an SE can be implemented in different ways that can be associated with different security levels, according to the type of attacks they have been designed to provide protection against. For example, an SE can be implemented as an independent security engine, as an exclusive CPU/Memory mode on a general purpose chip, or as an enclave providing memory encryption and code/data execution isolation. Within the scope of the present document the following security levels and associated categories of implementation are distinguished:

- Security Level 3 (highest), able to provide tamper resistance against attackers that have physical access to the supporting hardware, e.g. having the ability to dismantle a device and implement sophisticated attacks such as playing with out-of-boundary operating conditions or perpetrating power analysis. This security level shall rely on a tamper resistant hardware SE implementation dedicated to security storage and processing (e.g. a GlobalPlatform eSE) and should be associated with application specific security assessment or certification process.

- Security Level 2 (medium), intended to provide strong protection against all kind of remote attacks but not targeting protection against attacks requiring physical control of the hardware. This security level shall rely at least on a hardware isolated SE implementation which may be integrated within the general purpose processing environment running the device software (e.g. a GlobalPlatform TEE [4]).

- Security Level 1 (low) which can be supported by pure software based SE implementations, providing confidence that the software design process followed best practice cybersecurity recommendations to provide reasonable resistance against software based attacks such as trojans or viruses.

When none of the above security levels can reasonably be claimed, Security Level 0 (no particular security attention) shall be indicated.

6.3 Tamper resistant hardware SE implementation

The following tamper resistant hardware SE implementations are considered within the current release of the present document:

- Implementation as Secure Elements in different form factors including:

  - UICC according to ETSI TS 102 221 [3]. In this case, multiple SE may be supported by means of multiple UICC ADFs (Application Directory File) and remotely managed accordingly, see ETSI TS 102 221 [3].

  - Other variants of it such as eSE according to GlobalPlatform Card Specification [5]. In this case, multiple SE may be implemented as multiple GlobalPlatform Security Domains and remotely managed accordingly.

As outlined in ETSI TR 118 508 [i.2], such implementations are recommended as countermeasures against key discovery and device cloning for devices that are physically exposed to potential attackers.

6.4 Hardware isolated SE implementation

The following hardware isolated SE implementations are considered within the current release of the present document:

- Trusted Execution Environment according to GlobalPlatform [4]. In this case, multiple SE may be implemented as multiple GlobalPlatform Security Domains and remotely managed accordingly.

According to the security analysis in ETSI TR 118 508 [i.2], such implementations are appropriate for devices likely to be the target of remote attacks, without being physically accessible to attackers.

6.5 Software based SE implementation

An SE as defined in ETSI TS 118 103 [1] provides security services to applications and guarantee process isolation. In addition to hardware based solutions this can also be provided by dedicated software implementations such as White Box Cryptography. Such implementations are only appropriate when the value or lifetime of the protected asset is limited and the risks of compromising are otherwise mitigated.

7 Logical Abstraction - McsReference point

7.1 Overview

The Mcs reference point shall enable applications and service layer entities which are outside of Secure Environments to make use of sensitive functions, sensitive data and applications residing within the Secure Environment, independently of the technical implementation of the Secure Environment, via a logical abstract interface. The logical abstraction interface is between the physical or logical Secure Environments and any external service layer entities and applications. The logical abstraction interface shall provide access to the sensitive functions, sensitive data and applications residing within the Secure Environments regardless of their number and Secure Environment architecture scenarios.

7.2 Mcs reference point

7.2.1 Secure Environment Identifier (M2M-SE-ID)

M2M nodes may contain multiple Secure Environments each associated with a corresponding Secure Environment Identifier (M2M-SE-ID). Each SE contains several M2M Security Services, i.e. sensitive functions execution environment and associated sensitive data storage area for code and data. An M2M-SE-ID is assigned to each Secure Environment.

Table 7.2.1-1: Secure Environment Identifier

Identifier	Assigned by	Assigned to	Assigned during	Lifetime	Uniqueness	Used during
M2M Secure Environment Identifier	M2M SE issuer or delegated stakeholder	Secure Environment	Pre- or remote Provisioning or during manufacturing	Lifetime of the contract with the stakeholder to whom the SE is assigned	Global per SE and per type of SE	communication establishment with and selection of SE

M2M-SE-ID is structured as follows:

- Type of SE followed by unique ID, where type of SE is defined as given in table 7.2.1-2 and the unique ID is defined as described in table 7.2.1-1.

Table 7.2.1-2: Type of Secure Environment

Class of SE	Type of Secure Environment	Coding
Independent hardware	UICC as per ETSI	1
Independent hardware	GlobalPlatform Secure Element	2
Integrated hardware	TEE as per GlobalPlatform	3
Software	Security Library	4
NOTE: Other values are RFU.		

7.2.2 Differences between Mcs and Mcc/Mca reference points

The Mcs reference point is a simple variant of the Mcc/Mca reference points specifying the interaction of CSEs and AEs with secure environments.

An <*SE*> resource shall represent information about a Secure Environment available in a node. There could be multiple <*SE*> resources in one node. Secure Environments are represented in <*CSEbase*> resources and <*AE*> resources as <*SE*> child resources.

The present document has no further impact on the specification ETSI TS 118 101 [2] and has no impact on the specification ETSI TS 118 104 [14]. However, the Mcs reference point uses much of the specification in ETSI TS 118 104 [14] and in particular allows use of the WebSocket binding in ETSI TS 118 120 [18]. Though the other bindings, i.e. the HTTP binding in ETSI TS 118 108 [15], the CoAP binding in ETSI TS 118 109 [16] and the MQTT binding in ETSI TS 118 110 [17], remain applicable, they are not so relevant in the context of a node implementation.

The Mcs reference point incorporates the following concepts from the Mcc/Mca reference points:

- The concept of operations acting on resources.
- The resource addressing from Mcc/Mca is used.
- The universal attributes and some common attributes of resources.

The Mcs reference point differs from Mcc/Mca in the following ways:

- The CSE/AE can only communicate directly with the secure environment – there are no transited CSEs. Only Blocking Mode communication method is supported.
- The <subscription> resource and NOTIFY operations are not supported.
- The registration is conducted by the creation of the <*SE*> child resource in the corresponding <*CSEbase*> resource or <*AE*> resource, respectively. An AE needs to be registered at the CSE to be able to access the SE.
- The Mcs interface involves AE or CSE located on the same node as the SE abstraction layer, hence Security Association Establishment does not apply as such and can be superseded by implementation dependent mechanisms.
- There are no announced resources.

Common data types are inherited from clause 6.3 of [14]. The present document does not mention optional common attributes that are not used over Mcs.

7.2.3 Namespaces used for resource and data types

Representations of resources applicable to the Mcs Interface employ the namespace identifier "senv:" for global XML elements associated with a resource type. Data types of the attributes and complex-type elements of these resource types may use any of the name space identifiers listed in table 7.2.3-1.

Any data types of XML elements defined for use in present document shall be one of name spaces in table 7.2.3-1.

Table 7.2.3-1: Namespaces applicable to resource types defined in the present document

Name space	Prefix	Name space definition	Types defined in
Secure Environment	senv:	http://www.onem2m.org/xml/secureEnvironment	the present document and ETSI TS 118 103 [1]
oneM2M protocol CDT	m2m:	http://www.onem2m.org/xml/protocol	ETSI TS 118 104 [14]
Device Configuration	dcfg:	http://www.onem2m.org/xml/deviceConfig	ETSI TS 118 122 [19]

7.2.4 Mcs Resource type definitions

The files defining the resource types of Mcs specific resources are given in table 7.2.4-1.

Table 7.2.4-1: Resource type definitions

Resource Type	XSD File Name
algorithmSpecificParameter	SENV-algorithmSpecificParameter-v3_0_0.xsd
cipher	SENV-cipher-v3_0_0.xsd
connectionInstance	SENV-connectionInstance-v3_0_0.xsd
hash	SENV-hash-v3_0_0.xsd
identity	SENV-identity-v3_0_0.xsd
Rand	SENV-rand-v3_0_0.xsd
secureConnection	SENV-secureConnection-v3_0_0.xsd
sensitiveDataObject	SENV-sensitiveDataObject-v3_0_0.xsd
SEReboot	SENV-SEReboot-v3_0_0.xsd
SE	SENV-SE-v3_0_0.xsd
signature	SENV-signature-v3_0_0.xsd

7.3 Resource SE

7.3.0 Overview

An <*SE*> resource shall represent information about a Secure Environment available in a node. There could be multiple <*SE*> resources in one node.

One Secure Environment may be represented in the <*CSEbase*> resource and multiple <*AE*> resources of that node. Concurrent accesses to the Secure Environment are resolved in the SE abstraction layer.

Common data types applicable to the Mcs interface are inherited from ETSI TS 118 104 [14].

The data types for the specific resource attributes specified in this clause are listed in the following subclauses and defined in the following file:

SENV-commonTypes-v3_0_0.xsd

Applicable values for resource attributes and for enumerating Mcs resources are detailed in clause 9. Short names for attributes and resource types are provided in clause 10.

The <*SE*> resource shall contain the child resources specified in table 7.3.0-1.

Table 7.3.0-1: Child resources of <SE> resource

Child Resources of <*SE*>	Child Resource Type	Multiplicity	Description
memory	<*mgmtObj*> as defined in the specialization [*memory]*	0..1	This resource provides the non volatile memory information of the Secure Environment. See clause D.4 of ETSI TS 118 101 [2].
firmware	<*mgmtObj*> as defined in the specialization [*firmware]*	*0..n*	This resource describes the information about the firmware of the Secure Environment include name, version etc. See clause D.2 of ETSI TS 118 101 [2].
software	<*mgmtObj*> as defined in the specialization [*software]*	0..n	This resource describes the information about the software of the Secure Environment. See clause D.3 of ETSI TS 118 101 [2].
deviceInfo	<*mgmtObj*> as defined in the specialization [*deviceInfo]*	0..n	The resource contains information about the Secure Environment, like identity, manufacturer and model number, if applicable. See clause D.8 of ETSI TS 118 101 [2].
SEReboot	<*mgmtObj*> as defined in the specialization [SER*eboot]*	0..n	The resource is the place to reboot the Secure Environment, if it is a rebootable hardware. In the case of secure elements there would be two resources, one for a cold reset and one for a warm reset of the secure element, defined in ISO/IEC 7816-3 [7].
accessControlPolicy	<*accessControlPolicy*>	0..n	The Access Control Policies (ACPs) shall be used by the SE to control access to the resources.
sensitiveDataObject	<*sensitiveDataObject*>	0..n	See clause 7.4.1.
cipher	<*cipher*>	0..n	See clause 7.5.1.
rand	<*rand*>	0..n	See clause 7.5.2.
hash	<*hash*>	0..n	See clause 7.5.3.
signature	<*signature*>	0..n	See clause 7.5.4.
secureConnection	<*secureConnection*>	0..n	See clause 7.6.1.
identity	<*identity*>	0..n	See clause 7.7.1.

The <*SE*> resource shall contain the attributes specified in table 7.3.0-2.

Table 7.3.0-2: Attributes of <SE> resource

Attributes of <*SE*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	Defines the resource type.
resourceID	1	RO	Defines an identifier for the resource. This attribute shall be provided by the creator. The creator shall assign a resourceID which is unique within its context.
resourceName	1	WO	This attribute is the name for the resource that is used for 'hierarchical addressing method' to represent the parent-child relationships of resources.
parentID	1	RO	This attribute is the *resourceID* of the parent of this resource.
creationTime	1	RO	Time/date of creation of the resource.
lastModifiedTime	1	RO	Last modification time/date of the resource.
accessControlPolicyIDs	0..1 (L)	RW	Is used to control access to the resource.
SEType	0..1	RO	See table 7.2.1-2.
m2mSeID	1	WO	See tables 7.2.1-1 and 7.2.1-2.
securityLevel	1	WO	See clause 6.2.
supportedResourceType	1 (L)	RW	List of the resource types which are supported in the SE.
e2eSecInfo	0..1 (L)	RW	Indicates the end-to-end security capabilities.
hostedCSELink	0..1	RW	This attribute allows to find the <CSEBase> resource representing the CSE that is residing on the Secure Environment that is represented by this <se> resource. The attribute contains the resource ID of that <CSEBase> resource.

Attributes of <*SE*>	Multiplicity	RW/ RO/ WO	Description
hostedAELinks	0..1 (L)	RW	This attribute allows to find the AEs hosted by this Secure Environment. The attribute contains a list of resource identifiers of <*AE*> resources representing the AEs residing on the specific Secure Environment that is represented by the current <*se*> resource.

Table 7.3.0-3: Data types of <SE> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
SEType	O	O	senv:SEType
securityLevel	M	NP	senv:securityLevel
m2mSeID	M	NP	m2m:ID
supportedResourceType	O	O	m2m:resourceType
e2eSecInfo	O	O	m2m:e2eSecInfo
hostedCSELink	O	O	m2m:ID
hostedAELinks	O	O	m2m:ID

7.3.1 Resource *SEReboot*

The *[SEReboot]* resource shall be used to reboot a Secure Environment. The *[SEReboot]* resource is a specialization of the <*mgmtObj*> resource.

The *[SEReboot]* resource shall contain the child resources specified in table 7.3.1-1.

Table 7.3.1-1: Child resources of [SEReboot] resource

Child Resources of *[SEReboot]*	Child Resource Type	Multiplicity	Description
[variable]	<*subscription*>	0..n	See clause 9.6.8 of ETSI TS 118 101 [2] where the type of this resource is described.
[variable]	<*semanticDescriptor*>	0..n	See clause 9.6.30 of ETSI TS 118 101 [2].

The *[SEReboot]* resource shall contain the attributes specified in table 7.3.1-2.

Table 7.3.1-2: Attributes of [SEReboot] resource

Attributes of *[seReboot]*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	See clause 9.6.1.3 of ETSI TS 118 101 [2].
resourceID	1	RO	See clause 9.6.1.3 of ETSI TS 118 101 [2].
resourceName	1	WO	See clause 9.6.1.3 of ETSI TS 118 101 [2].
parentID	1	RO	See clause 9.6.1.3 of ETSI TS 118 101 [2].
expirationTime	1	RW	See clause 9.6.1.3 of ETSI TS 118 101 [2].
accessControlPolicyIDs	0..1 (L)	RW	See clause 9.6.1.3 of ETSI TS 118 101 [2].
creationTime	1	RO	See clause 9.6.1.3 of ETSI TS 118 101 [2].
lastModifiedTime	1	RO	See clause 9.6.1.3 of ETSI TS 118 101 [2].
Labels	0..1(L)	RW	See clause 9.6.1.3 of ETSI TS 118 101 [2].
mgmtDefinition	1	WO	See clause 9.6.15 of ETSI TS 118 101 [2]. This attribute shall have the fixed value "seReboot".
objectIDs	0..1 (L)	WO	See clause 9.6.15 of ETSI TS 118 101 [2].
objectPaths	0..1 (L)	WO	See clause 9.6.15 of ETSI TS 118 101 [2].
Description	0..1	RW	See clause 9.6.15 of ETSI TS 118 101 [2].
rebootType	1	RO	The type of reboot supported by the Secure Environment. This attribute is a specialization of *[objectAttribute]* attribute. Type of reboots could be such as Cold Reset or Warm Reset as defined in ISO/IEC 7816-3 [7].
SEReboot	1	RW	The action that allows rebooting the device. The action is triggered by assigning value "TRUE" to this attribute. This attribute is a specialization of *[objectAttribute]* attribute.

Table 7.3.1-3: Data types of <SEReboot> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
RebootType	M	NP	senv:RebootType
SEReboot	O	O	xs:boolean

7.4 Sensitive Data Storage

7.4.1 <sensitiveDataObject> resource

Secure Environments shall provide a service to store and protect sensitive data. Sensitive data objects are represented as SE-resources and are created and managed within the Secure Environment. Requests to SE-resources are using absolute addressing. A <*sensitiveDataObject*> resource shall represent sensitive data and related information owned by a creator.

Attributes in <*sensitiveDataObject*> are shown in table 7.4.1-1.

Table 7.4.1-1: Attributes of <sensitiveDataObject> resource

Attributes of <sensitiveDataObject>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	Defines the resource type.
resourceID	1	RO	Defines an identifier for the resource. This attribute shall be provided by the creator. The creator shall assign a resourceID which is unique within its context.
creationTime	1	RO	Time/date of creation of the resource. The *creationTime* is set by the CSE hosting the SE when the resource is created.
lastModifiedTime	1	RO	Last modification time/date of the resource. This attribute is mandatory. The *lastModifiedTime* value is set by the Hosting CSE when the resource is created, and the *lastModifiedTime* value is updated when the resource is updated.
Creator	1	RO	The AE-ID or CSE-ID of the AE or CSE creating the resource.
currentByteSize	1	RO	Current size in bytes of sensitive data.
sensitiveData	1	RW	Contains sensitive data and required information to access and manage sensitive data owned by a dedicated creator.
accessControlPolicyID	0..1 (L)	RW	Is used to control access to the resource. If no *accessControlPolicyIDs* value is configured, the *accessControlPolicyIDs* of the parent resource shall be applied for privilege checking.

Table 7.4.1-2: Data types of <sensitiveDataObject> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
currentByteSize	M	NP	xs:nonNegativeInteger
sensitiveData	O	O	xs:byte
creator	M	NP	m2m:ID

7.4.2 <sensitiveDataObject> Resource Procedures

7.4.2.1 CREATE <sensitiveDataObject>

This procedure shall be used for creating a <*sensitiveDataObject*> resource.

Table 7.4.2.1-1: <*sensitiveDataObject*> CREATE

<*sensitiveDataObject*> CREATE request message parameters	
Associated Reference Point	Mcs
Information in Request message	Following parameters shall exist within the Create request: Operation *To:* contains M2M-SE-ID From Registree AE or CSE Request Identifier *Content:* <sensitiveData> Name: name of resource
Processing at Originator before sending Request	Establish security association between creator and SE Requests from an AE or CSE includes their AE-ID or CSE-ID
Processing at Receiver	Check seAccessPrivileges and validate request
Information in Response message	Response status codes: ack; successful operation = CREATE; Unsuccessful Operation = C; Request Identifier
Processing at Originator after receiving Response	n/a
Exceptions	According to ETSI TS 118 101 [2]

7.4.2.2 RETRIEVE <sensitiveDataObject>

This procedure shall be used for retrieving a <*sensitiveDataObject*> resource.

Table 7.4.2.2-1: <*sensitiveDataObject*> RETRIEVE

<*sensitiveDataObject*> RETRIEVE request message parameters	
Associated Reference Point	Mcs
Information in Request message	Following parameters shall exist within the RETRIEVE request: Operation *To:* contains M2M-SE-ID From Registree AE or CSE Request Identifier
Processing at Originator before sending Request	Establish security association between creator and SE Requests from an AE or CSE includes their AE-ID or CSE-ID
Processing at Receiver	Check seAccessPrivileges and validate request
Information in Response message	Response status codes: ack; successful operation = RETRIEVE; Unsuccessful Operation = R; Request Identifier Content = Sensitive Data
Processing at Originator after receiving Response	As defined in ETSI TS 118 101 [2]
Exceptions	As defined in ETSI TS 118 101 [2]

7.4.2.3 UPDATE <sensitiveDataObject>

This procedure shall be used for updating the attributes and actual data of a *<sensitiveDataObject>* resource.

Table 7.4.2.3-1: *<sensitiveDataObject>* UPDATE

***<sensitiveDataObject>* UPDATE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	Following parameters shall exist within the UPDATE request: Operation *To:* contains M2M-SE-ID From Registree AE or CSE Request Identifier Name *Content:* sensitive data and/or attributes
Processing at Originator before sending Request	Establish security association between creator and SE Requests from an AE or CSE includes their AE-ID or CSE-ID
Processing at Receiver	Check seAccessPrivileges and validate request
Information in Response message	Response status codes: ack; successful operation = UPDATE; Unsuccessful Operation = R; Request Identifier
Processing at Originator after receiving Response	As defined in ETSI TS 118 101 [2]
Exceptions	As defined in ETSI TS 118 101 [2]

7.4.2.4 DELETE <sensitiveDataObject>

This procedure shall be used for deleting a *<sensitiveDataObject>* resource.

Table 7.4.2.4-1: *<sensitiveDataObject>* DELETE

***<sensitiveDataObject>* DELETE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	Following parameters shall exist within the DELETE request: Operation *To:* contains M2M-SE-ID From Registree AE or CSE Request Identifier Name
Processing at Originator before sending Request	Establish security association between creator and SE Requests from an AE or CSE includes their AE-ID or CSE-ID
Processing at Receiver	Check seAccessPrivileges and validate request
Information in Response message	Response status codes: ack; successful operation = DELETE Unsuccessful Operation = D Request Identifier
Processing at Originator after receiving Response	As defined in ETSI TS 118 101 [2]
Exceptions	As defined in ETSI TS 118 101 [2]

7.5 Sensitive Cryptographic Functions

7.5.1 <cipher> resource

7.5.1.0 Introduction

Secure Environments shall provide a service for cryptographic operations. A <cipher> resource shall represent sensitive data and related information owned by a creator.

The <*cipher*> resource shall contain the child resources specified in table 7.5.1.0-1.

Table 7.5.1.0-1: Child resources of <*cipher*> resource

Child Resources of <*cipher*>	Child Resource Type	Multiplicity	Description
encrypt	*<encrypt>*	1	See clause 7.5.1.2
decrypt	*<decrypt>*	1	See clause 7.5.1.3
generateKey	*<generateKey>*	0..1	See clause 7.5.1.4
algorithmSpecificParameter	*<algorithmSpecificParameter>*	0..1	See clause 7.5.1.5

Attributes in <*cipher*> are shown in table 7.5.1.0-2.

Table 7.5.1.0-2: Attributes of <*cipher*> resource

Attributes of <*signature*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	Defines the resource type.
resourceID	1	RO	Defines an identifier for the resource. This attribute shall be provided by the creator. The creator shall assign a resourceID which is unique within its context.
resourceName	1	WO	This attribute is the name for the resource that is used for 'hierarchical addressing method' to represent the parent-child relationships of resources.
parentID	1	RO	This attribute is the *resourceID* of the parent of this resource.
expirationTime	1	RW	Time/date after which the resource will be deleted by the Hosting CSE.
accessControlPolicyIDs	0..1 (L)	RW	Is used to control access to the resource. If no *accessControlPolicyIDs* are provided at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute.
creationTime	1	RO	Time/date of creation of the resource. The *creationTime* is set by the CSE hosting the SE when the resource is created.
lastModifiedTime	1	RO	Last modification time/date of the resource. This attribute is mandatory. The *lastModifiedTime* value is set by the Hosting CSE when the resource is created, and the *lastModifiedTime* value is updated when the resource is updated.
sensitiveData	0..1	RW	Message to be either encrypted or decrypted.
currentByteSize	1	RW	Current size in bytes of sensitive data.
maxByteSize	1	RO	Maximum size in bytes of sensitive data.
algorithm	1	WO	Contains the algorithm type of the resource instance.
keyData	0..1	WO	Contains the value of the key.
calculatedData	0..1	RO	Contains the result of a cipher operation.

Table 7.5.1.0-3: Data types of <cipher> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
sensitiveData	O	O	xs:byte
algorithm	M	NP	senv:cipherAlgorithm
keyData	O	NP	xs:byte
currentByteSize	O	O	xs:nonNegativeInteger
maxByteSize	M	NP	xs:nonNegativeInteger
calculatedData	NP	NP	xs:byte

The following types are defined for the algorithm attribute:

- ALG_AEAD_AES_128_GCM: The AEAD_AES_128_GCM authenticated encryption algorithm works as specified in IETF RFC 5116 [8], using AES-128 as the block cipher, by providing the key, nonce, and plaintext, and associated data to that mode of operation.

- ALG_AEAD_AES_256_GCM: This algorithm is identical to AEAD_AES_128_GCM, but with the following differences: K_LEN is 32 octets, instead of 16 octets, and AES-256 GCM is used instead of AES-128 GCM.
- ALG_AEAD_AES_128_CCM: The AEAD_AES_128_CCM authenticated encryption algorithm works as specified in IETF RFC 5116 [8], using AES-128 as the block cipher, by providing the key, nonce, associated data, and plaintext to that mode of operation.
- ALG_AEAD_AES_256_CCM: This algorithm is identical to AEAD_AES_128_CCM, but with the following differences: K_LEN is 32 octets, instead of 16, and AES-256 CCM is used instead of AES-128 CCM.
- ALG_AEAD_AES_128_CCM_8: The AEAD_AES_128_CCM_8 authenticated encryption algorithm is identical to the AEAD_AES_128_CCM algorithm (see section 5.3 of IETF RFC 5116 [8]), except that it uses 8 octets for authentication, instead of the full 16 octets used by AEAD_AES_128_CCM (see section 6.1 of IETF RFC 6655 [9]).
- ALG_AEAD_AES_256_CCM_8: The AEAD_AES_256_CCM_8 authenticated encryption algorithm is identical to the AEAD_AES_256_CCM algorithm (see section 5.4 of IETF RFC 5116 [8]), except that it uses 8 octets for authentication, instead of the full 16 octets used by AEAD_AES_256_CCM (see section 6.2 of IETF RFC 6655 [9]).
- ALG_AES_BLOCK_128_CBC_NOPAD : Cipher algorithm ALG_AES_BLOCK_128_CBC_NOPAD provides a cipher using AES with block size 128 in CBC mode and does not pad input data.
- ALG_AES_CBC_ISO9797_M1: Cipher algorithm ALG_AES_CBC_ISO9797_M1 provides a cipher using AES with block size 128 in CBC mode, and pads input data according to the ISO 9797 [10] method 1 scheme.
- ALG_AES_CBC_ISO9797_M2: Cipher algorithm ALG_AES_CBC_ISO9797_M2 provides a cipher using AES with block size 128 in CBC mode, and pads input data according to the ISO 9797 [10] method 2 (ISO 7816-4 [i.3], EMV'96) scheme.
- ALG_AES_CBC_PKCS5: Cipher algorithm ALG_AES_CBC_PKCS5 provides a cipher using AES with block size 128 in CBC mode, and pads input data according to the PKCS#5 scheme.

7.5.1.1 <cipher> Resource Procedures

7.5.1.1.1 CREATE <cipher>

This procedure shall be used for creating a <cipher> resource.

Table 7.5.1.1.1-1: <*cipher*> CREATE

<*cipher*> CREATE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID/CSE-ID *Content:* The resource content shall provide the information as defined in clause 7.5.1
Processing at Originator before sending Request	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.1.1.of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with the specific details for: *Content:* Address of the created <cipher> resource, according to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.1.1 of ETSI TS 118 101 [2]

If <*generateKey*> is created, the key to be used will be generated and stored in keyData.

7.5.1.1.2 RETRIEVE <cipher>

This procedure shall be used for retrieving the generated output from all/last input data of the <*Cipher*> resource.

Table 7.5.1.1.2-1: <*cipher*> RETRIEVE

<*cipher*> RETRIEVE request message parameters	
Associated Reference Point	Mcs
Information in Request message	According to clause 10.1.2 of ETSI TS 118 101 [2] with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID/CSE-ID
Processing at Originator before sending Request	According to clause 10.1.2 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.2 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with specific details for: *Content:* Attributes of the <*cipher*> resources as defined in clause 7.5.1
Processing at Originator after receiving Response	According to clause 10.1.2 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.2 of ETSI TS 118 101 [2]

7.5.1.1.3 UPDATE <cipher>

This procedure shall be used for updating the <cipher> resource with data to encrypt or decrypt. It may be necessary to use this procedure several times until all data is transmitted.

Table 7.5.1.1.3-1: <*cipher*> UPDATE

<*cipher*> UPDATE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE-hosted AE-ID/CSE-ID *Content:* attributes of the <cipher> resource as defined in clause 7.5.1 which need be updated
Processing at Originator before sending Request	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.3 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.3 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.3 of ETSI TS 118 101 [2]

7.5.1.1.4 DELETE <cipher>

This procedure shall be used for deleting a <*cipher*> resource.

Table 7.5.1.1.4-1: <*cipher*> DELETE

<*cipher*> DELETE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID/CSE-ID
Processing at Originator before sending Request	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.4.1 of ETSI TS 118 101 [2]

7.5.1.2 <encrypt> Resource

The <encrypt> resource is a virtual resource because it does not have a representation. It is the child resource of a <cipher> resource. When a RETRIEVE request addresses the <encrypt> resource, the *sensitiveData* of the <cipher> resource shall be encrypted and the result shall be stored in *calculatedData.*

The <encrypt> resource inherits access control policies that apply to the parent <cipher> resource.

7.5.1.3 <decrypt> Resource

The <decrypt> resource is a virtual resource because it does not have a representation. It is the child resource of a <cipher> resource. When a RETRIEVE request addresses the <decrypt> resource, the *sensitiveData* of the <cipher> resource shall be decrypted and the result shall be stored in *calculatedData*.

The <decrypt> resource inherits access control policies that apply to the parent <cipher> resource.

7.5.1.4 <generateKey> Resource

The <generateKey> resource is a virtual resource because it does not have a representation. When a RETRIEVE request addresses the <generateKey> resource, the *keyData* attribute shall be filled with a key generated according to the *algorithm* attribute.

The <generateKey> resource inherits access control policies that apply to the parent resource.

7.5.1.5 <algorithmSpecificParameter> Resource

The <*algorithmSpecificParameter*> contains parameter required for the different algorithm.

The <*algorithmSpecificParameter*> resource shall inherit the same access control policies of the parent <*cipher*> resource, and shall not have its own *accessControlPolicyIDs* attribute.

Attributes in <*algorithmSpecificParameter*> are shown in table 7.5.1.5-1.

Table 7.5.1.5-1: Attributes of <*algorithmSpecificParameter*> resource

Attributes of <*algorithmSpecificParameter*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	Defines the resource type.
resourceID	1	RO	Defines an identifier for the resource. This attribute shall be provided by the creator. The creator shall assign a resourceID which is unique within its context.
resourceName	1	WO	This attribute is the name for the resource that is used for 'hierarchical addressing method' to represent the parent-child relationships of resources.
parentID	1	RO	This attribute is the *resourceID* of the parent of this resource.
expirationTime	1	RW	Time/date after which the resource will be deleted by the Hosting CSE.
accessControlPolicyIDs	0..1 (L)	RW	Is used to control access to the resource. If no *accessControlPolicyIDs* are provided at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute
creationTime	1	RO	Time/date of creation of the resource. The *creationTime* is set by the CSE hosting the SE when the resource is created.
lastModifiedTime	1	RO	Last modification time/date of the resource. This attribute is mandatory. The *lastModifiedTime* value is set by the Hosting CSE when the resource is created, and the *lastModifiedTime* value is updated when the resource is updated.
initialVector	0..1	RW	AES algorithms, except for ALG_AES_CMAC_128, in CBC mode expect a 16-byte parameter value for the initialization vector (IV).
Nonce	0..1	RW	a value for the nonce as expected by AEAD algorithm
associatedData	0..1	RW	The associated data for AEAD algorithm, which contains the data to be authenticated, but not encrypted.
Label	0..1	RW	"EXPORTER-oneM2M-Bootstrap" for TLS Key Export for Enrolment Key. "EXPORTER-oneM2M-Connection" for TLS Key Export for M2M Secure Connection Key.

Table 7.5.1.5-2: Data types of <algorithmSpecificParameter> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
initialVector	O	O	xs:byte
nonce	O	O	xs:byte
associatedData	O	O	xs:byte
label	O	O	senv:cipherLabel

7.5.2 <rand> resource

7.5.2.0 Introduction

A <rand> resource shall represent random data owned by a creator.

The <*rand*> resource shall contain the child resources specified in table 7.5.2.0-1.

Table 7.5.2.0-1: Child resources of <*rand*> resource

Child Resources of <*rand*>	Child Resource Type	Multiplicity	Description	<*randAnnc*> Child Resource Types
generateRand	<*generateRand*>	1	See clause 7.5.2.2	*None*

Attributes in <*rand*> are shown in table 7.5.2.0-2.

Table 7.5.2.0-2: Attributes of <*rand*> resource

Attributes of <*rand*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	Defines the resource type.
resourceID	1	RO	Defines an identifier for the resource. This attribute shall be provided by the creator. The creator shall assign a resourceID which is unique within its context.
resourceName	1	WO	This attribute is the name for the resource that is used for 'hierarchical addressing method' to represent the parent-child relationships of resources.
parentID	1	RO	This attribute is the *resourceID* of the parent of this resource.
expirationTime	1	RW	Time/date after which the resource will be deleted by the Hosting CSE.
accessControlPolicyIDs	0..1 (L)	RW	Is used to control access to the resource. If no *accessControlPolicyIDs* are provided at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute.
creationTime	1	RO	Time/date of creation of the resource. The *creationTime* is set by the CSE hosting the SE when the resource is created.
lastModifiedTime	1	RO	Last modification time/date of the resource. This attribute is mandatory. The *lastModifiedTime* value is set by the Hosting CSE when the resource is created, and the *lastModifiedTime* value is updated when the resource is updated.
randomData	0..1	RO	Contains random data which can be retrieved by the creator.
rngType	1	WO	The following types of RNGs can be requested according to ISO/IEC 18031 [13]: pseudo RNG or true (physical) RNG.
requestedDataSize	0..1	RW	Requested amount of randomData in Bytes.

Table 7.5.2.0-3: Data types of <rand> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
randomData	NP	NP	xs:byte
rngType	M	NP	senv:rngType
requestedDataSize	O	O	xs:nonNegativeInteger

7.5.2.1 <rand> Resource Procedures

7.5.2.1.1 CREATE <rand>

This procedure shall be used for creating a <rand> resource.

Table 7.5.2.1.1-1: *<rand>* CREATE

***<rand>* CREATE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID *Content:* The resource content shall provide the information as defined in clause 7.5.2
Processing at Originator before sending Request	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with the specific details for: *Content:* Address of the created <cipher> resource, according to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.1.1 of ETSI TS 118 101 [2]

7.5.2.1.2 RETRIEVE <rand>

This procedure shall be used for retrieving the random numbers.

Table 7.5.2.1.2-1: *<rand>* RETRIEVE

***<rand>* RETRIEVE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	According to clause 10.1.2 of ETSI TS 118 101 [2] with the specific details for *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.2 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.2 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with specific details for: *Content:* Attributes of the *<rand>* resources as defined in clause 7.5.2
Processing at Originator after receiving Response	According to clause 10.1.2 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.2 of ETSI TS 118 101 [2]

7.5.2.1.3 UPDATE <rand>

This procedure shall be used for setting the amount of random data which is requested.

Table 7.5.2.1.3-1: *<rand>* UPDATE

***<rand>* UPDATE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID *Content:* attributes of the <rand> resource as defined in clause 7.5.2 which need be updated
Processing at Originator before sending Request	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.3 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.3 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.3 of ETSI TS 118 101 [2]

7.5.2.1.4 DELETE <rand>

This procedure shall be used for deleting a <*rand*> resource.

Table 7.5.2.1.4-1: <rand> DELETE

***<rand>* DELETE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.4.1 of ETSI TS 118 101 [2]

7.5.2.2 <generateRand> Resource

The <generateRand> resource is a virtual resource because it does not have a representation. When a RETRIEVE request addresses the <generateRand> resource, the *randomData* attribute shall be filled with random numbers from a random number generator according to the *rngType* attribute.

The <generateRand> resource inherits access control policies that apply to the parent resource.

7.5.3 <hash> resource

7.5.3.0 Introduction

Secure Environments shall provide a service for calculating hashes. A <*hash*> resource shall represent sensitive data and related information owned by a creator.

The <*hash*> resource shall contain the child resources specified in table 7.5.3.0-1.

Table 7.5.3.0-1: Child resources of <*hash*> resource

Child Resources of <*hash*>	Child Resource Type	Multiplicity	Description	<*hashAnnc*> Child Resource Types
calculateHash	*<calculateHash>*	1	See clause 7.5.3.2	*None*

Attributes in <*hash*> are shown in table 7.5.3.0-2.

Table 7.5.3.0-2: Attributes of <hash> resource

Attributes of <*hash*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	Defines the resource type.
resourceID	1	RO	Defines an identifier for the resource. This attribute shall be provided by the creator. The creator shall assign a resourceID which is unique within its context.
resourceName	1	WO	This attribute is the name for the resource that is used for 'hierarchical addressing method' to represent the parent-child relationships of resources.
parentID	1	RO	This attribute is the *resourceID* of the parent of this resource.
expirationTime	1	RW	Time/date after which the resource will be deleted by the Hosting CSE.
accessControlPolicyIDs	0..1 (L)	RW	Is used to control access to the resource. If no *accessControlPolicyIDs* are provided at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute.
creationTime	1	RO	Time/date of creation of the resource. The *creationTime* is set by the CSE hosting the SE when the resource is created.
lastModifiedTime	1	RO	Last modification time/date of the resource. This attribute is mandatory. The *lastModifiedTime* value is set by the Hosting CSE when the resource is created, and the *lastModifiedTime* value is updated when the resource is updated.
algorithm	1	WO	Specifies the algorithm for the hash.
message	0..1	RW	The message which is to be hashed.
hashValue	0..1	RO	Is the calculated Hash.

Table 7.5.3.0-3: Data types of <hash> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
algorithm	M	NP	senv:hashAlgorithm
message	O	O	xs:byte
hashValue	NP	NP	xs:byte

The following types are defined as algorithm types:

- SHA256;
- SHA384;
- SHA512.

7.5.3.1 <hash> Resource Procedures

7.5.3.1.1 CREATE <hash>

This procedure shall be used for creating a <hash> resource.

Table 7.5.3.1.1-1: *<hash>* CREATE

***<hash>* CREATE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID *Content:* The resource content shall provide the information as defined in clause 7.5.3
Processing at Originator before sending Request	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with the specific details for: *Content:* Address of the created <hash> resource, according to clause 10.1.1.1.of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.1.1 of ETSI TS 118 101 [2]

7.5.3.1.2 RETRIEVE <hash>

This procedure shall be used for retrieving the calculated hash.

Table 7.5.3.1.2-1: *<hash>* RETRIEVE

***<hash>* RETRIEVE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	According to clause 10.1.2 of ETSI TS 118 101 [2] with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.2 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.2 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with specific details for: *Content:* Attributes of the *<hash>* resources as defined in clause 7.5.3
Processing at Originator after receiving Response	According to clause 10.1.2 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.2 of ETSI TS 118 101 [2]

7.5.3.1.3 UPDATE <hash>

This procedure shall be used for updating the <hash> resource with data to be hashed.

Table 7.5.3.1.3-1: *<hash>* UPDATE

***<hash>* UPDATE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID Content = attributes of the <hash> resource as defined in clause 7.5.3 which need be updated
Processing at Originator before sending Request	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.3 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.3 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.3 of ETSI TS 118 101 [2]

7.5.3.1.4 DELETE <hash>

This procedure shall be used for deleting a <*hash*> resource.

Table 7.5.3.1.4-1: *<hash>* DELETE

***<hash>* DELETE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.4.1 of ETSI TS 118 101 [2]

7.5.3.2 <calculateHash> Resource

The <calculateHash> resource is a virtual resource because it does not have a representation. When a RETRIEVE request addresses the <calculateHash> resource, the *hashValue* attribute shall be filled with the hash calculated over the data in the *message* attribute according to the *algorithm* attribute.

The <calculateHash> resource inherits access control policies that apply to the parent resource.

7.5.4 <signature> resource

7.5.4.0 Introduction

Secure Environments shall provide a service for signing messages and verifying signatures. A <signature> resource shall represent sensitive data and related information owned by a creator.

The <*signature*> resource shall contain the child resources specified in table 7.5.4.0-1.

Table 7.5.4.0-1: Child resources of <*signature*> resource

Child Resources of <*signature*>	Child Resource Type	Multiplicity	Description	<*signatureAnnc*> Child Resource Types
calculateSignature	<*calculateSignature*>	1	See clause 7.5.4.2	*None*
verifySignature	<*verifySignature*>	1	See clause 7.5.4.3	*None*
generateKey	<*generateKey*>	0..1	See clause 7.5.4.4	*None*

Attributes in <*Signature*> are shown in table 7.5.4.0-2.

Table 7.5.4.0-2: Attributes of <*signature*> resource

Attributes of <*signature*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	Defines the resource type.
resourceID	1	RO	Defines an identifier for the resource. This attribute shall be provided by the creator. The creator shall assign a resourceID which is unique within its context.
resourceName	1	WO	This attribute is the name for the resource that is used for 'hierarchical addressing method' to represent the parent-child relationships of resources.
parentID	1	RO	This attribute is the *resourceID* of the parent of this resource.
expirationTime	1	RW	Time/date after which the resource will be deleted by the Hosting CSE.
accessControlPolicyIDs	0..1 (L)	RW	Is used to control access to the resource. If no *accessControlPolicyIDs* are provided at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute.
creationTime	1	RO	Time/date of creation of the resource. The *creationTime* is set by the CSE hosting the SE when the resource is created.
lastModifiedTime	1	RO	Last modification time/date of the resource. This attribute is mandatory. The *lastModifiedTime* value is set by the Hosting CSE when the resource is created, and the *lastModifiedTime* value is updated when the resource is updated.
message	0..1	RW	Message either to be signed or to be used to verify a signature, this could be alternatively a hash value.
algorithm	1	WO	Contains the algorithm type of the resource instance.
keyData	0..1	WO	Contains the value of the key.
keyInformation	0..1	RW	Contains information about the key like a certificate.
signature	0..1	RW	Contains the signature either to be calculated or to be verified.
verificationResult	0..1	RO	Contains the result of a signature verification operation.

Table 7.5.4.0-3: Data types of <signature> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
message	O	O	xs:byte
algorithm	M	NP	senv:signatureAlgorithm
keyData	O	NP	xs:byte
keyInformation	O	O	xs:anyType
signature	O	O	xs:byte
verificationResult	NP	NP	xs:boolean

The following types are defined for the algorithm attribute:

- ALG_AES_CMAC_128 : Signature algorithm ALG_AES_CMAC_128 generates a 16-byte Cipher-based MAC (CMAC) using AES with blocksize 128 in CBC mode with ISO9797_M2 padding scheme.
- ALG_AES_MAC_128_NOPAD :Signature algorithm ALG_AES_MAC_128_NOPAD generates a 16-byte MAC using AES with blocksize 128 in CBC mode and does not pad input data.
- ALG_ECDSA_SHA_256 :Signature algorithm ALG_ECDSA_SHA_256 generates a 32-byte SHA-256 digest and signs/verifies the digest using ECDSA with the curve defined in the ECKey parameters - such as the P-256 curve specified in the Digital Signature Standards specification NIST FIPS PUB 186-4 [11].
- ALG_ECDSA_SHA_384 :Signature algorithm ALG_ECDSA_SHA_384 generates a 48-byte SHA-384 digest and signs/verifies the digest using ECDSA with the curve defined in the ECKey parameters - such as the P-384 curve specified in the Digital Signature Standards specification NIST FIPS PUB 186-4 [11].
- ALG_ECDSA_SHA_512 :Signature algorithm ALG_ECDSA_SHA_512 generates a 64-byte SHA-512 digest and signs/verifies the digest using ECDSA with the curve defined in the ECKey parameters - such as the P-521 curve specified in the Digital Signature Standards specification NIST FIPS PUB 186-4 [11].
- ALG_HMAC_SHA_256 :HMAC message authentication algorithm ALG_HMAC_SHA_256 This algorithm generates an HMAC following the steps found in IETF RFC 2104 [12] using SHA-256 as the hashing algorithm.
- ALG_HMAC_SHA_384 :HMAC message authentication algorithm ALG_HMAC_SHA_384 This algorithm generates an HMAC following the steps found in IETF RFC 2104 [12] using SHA-384 as the hashing algorithm.
- ALG_HMAC_SHA_512 :HMAC message authentication algorithm ALG_HMAC_SHA_512 This algorithm generates an HMAC following the steps found in IETF RFC 2104 [12] using SHA-512 as the hashing algorithm.

7.5.4.1 <signature> Resource Procedures

7.5.4.1.1 CREATE <signature>

This procedure shall be used for creating a <signature> resource.

Table 7.5.4.1.1-1: <*signature*> CREATE

<*Signature*> CREATE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID *Content:* The resource content shall provide the information as defined in clause 7.5.1
Processing at Originator before sending Request	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with the specific details for: *Content:* Address of the created <signature> resource, according to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.1.1 of ETSI TS 118 101 [2]

If <*generateKey*> is created, the key to be used shall be generated and stored in keyData, in such a case keyInformation shall be filled with the public part of the generated key.

7.5.4.1.2 RETRIEVE <signature>

This procedure shall be used for retrieving either the calculated signature of the <*Signature*> resource or the result of the verification of a signature.

Table 7.5.4.1.2-1: <*signature*> RETRIEVE

<*signature*> RETRIEVE request message parameters	
Associated Reference Point	Mcs
Information in Request message	According to clause 10.1.2 of ETSI TS 118 101 [2] with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.2 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.2 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with specific details for: *Content:* Attributes of the <*signature*> resources as defined in clause 7.5.4
Processing at Originator after receiving Response	According to clause 10.1.2 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.2 of ETSI TS 118 101 [2]

7.5.4.1.3 UPDATE <signature>

This procedure shall be used for updating the <signature> resource with the *message* to be signed or to be verified and in the case of verification the *signature* to be verified.

Table 7.5.4.1.3-1: <*signature*> UPDATE

<*signature*> UPDATE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID *Content:* attributes of the <signature> resource as defined in clause 7.5.4 which need be updated
Processing at Originator before sending Request	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.3 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.3 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.3 of ETSI TS 118 101 [2]

7.5.4.1.4 DELETE <signature>

This procedure shall be used for deleting a <*signature*> resource.

Table 7.5.4.1.4-1: <*signature*> DELETE

<*signature*> DELETE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.4.1 of ETSI TS 118 101 [2]

7.5.4.2 <calculateSignature> Resource

The <calculateSignature> resource is a virtual resource because it does not have a representation. It is the child resource of a <signature> resource. When a RETRIEVE request addresses the <calculateSignature> resource, the signature shall be calculated and written in the *signature* attribute of the <signature> resource.

The <calculateSignature> resource inherits access control policies that apply to the parent <signature> resource.

7.5.4.3 <verifySignature> Resource

The <verifySignature> resource is a virtual resource because it does not have a representation. It is the child resource of a <signature> resource. When a RETRIEVE request addresses the <verifySignature> resource, *signature* attribute of the <signature> resource shall be verified and the result shall be stored in *verificationResult*.

The <verifySignature> resource inherits access control policies that apply to the parent <cipher> resource.

7.5.4.4 <generateKey> Resource

The <generateKey> resource is a virtual resource because it does not have a representation. When a RETRIEVE request addresses the <generateKey> resource, the *keyData* attribute shall be filled with a key generated according the *algorithm* attribute.

The <generateKey> resource inherits access control policies that apply to the parent resource.

7.6 Secure Connection Establishment

7.6.1 <secureConnection> resource

Secure Environments shall provide a service to AEs or CSEs to establish a secure connection to a dedicated communication partner. The *<secureConnection>* resource shall represent the services offered by the Secure Environment to enable the establishment of a secure connection to a communication partner. The services include the following:

- generation of key material within the secure environment that can be used for the establishment of a secure connection by the requesting entity (creator) outside of the secure environment;
- acting as secure connection endpoint and sending the data provided by the requesting entity (creator) from within the secure environment with the key material generated and kept inside the secure environment.

The *<secureConnection>* resource shall contain the child resources specified in table 7.6.1-1.

Table 7.6.1-1: Child resources of *<secureConnection>* resource

Child Resources of *<secureConnection>*	Child Resource Type	Multiplicity	Description
connectionInstance	*<connectionInstance>*	0..n	See clause 7.6.3
generateKey	*<generateKey>*	0..1	See clause 7.6.7

Attributes in *<secureConnection>* are shown in table 7.6.1-2.

Table 7.6.1-2: Attributes of *<secureConnection>* resource

Attributes of *<secureConnection>*	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	Defines the resource type.
resourceID	1	RO	Defines an identifier for the resource. This attribute shall be provided by the creator. The creator shall assign a resourceID which is unique within its context.
resourceName	1	WO	This attribute is the name for the resource that is used for 'hierarchical addressing method' to represent the parent-child relationships of resources.
parentID	1	RO	This attribute is the *resourceID* of the parent of this resource.
expirationTime	1	RW	Time/date after which the resource will be deleted by the Hosting CSE.
accessControlPolicyIDs	0..1 (L)	RW	Is used to control access to the resource. If no *accessControlPolicyIDs* are provided at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute.
creationTime	1	RO	Time/date of creation of the resource. The *creationTime* is set by the CSE hosting the SE when the resource is created.
lastModifiedTime	1	RO	Last modification time/date of the resource. This attribute is mandatory. The *lastModifiedTime* value is set by the Hosting CSE when the resource is created, and the *lastModifiedTime* value is updated when the resource is updated.
maxNrOfInstances	0..1	RO	Maximum number of direct child *<connectionInstance>* resources in the <secureConnection> resource.

Attributes of <*secureConnection*>	Multiplicity	RW/ RO/ WO	Description
currentNrOfInstances	0..1	RW	Current number of direct child <*connectionInstance*> resource in the <*secureConnection*> resource. It is limited by the *maxNrOfInstances*.
connectionType	1	RW	Contains the type of connection that has to be supported. Supported types are: • TLS; • DTLS; • SMS; • E2EKey.
keyData	0..1	WO	Contains the key material.
keyInformation	0..1	RW	Specifies the additional information required for the key and the ciphersuite, e.g. Certificates, rootkeys, the public part of keyData.
cipherSuite	0..1	RW	Specifies the ciphersuites that are supported. Supported cipher suites are given in ETSI TS 118 103 [1].

Table 7.6.1-3: Data types of <secureConnection> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
maxNrOfInstances	M	NP	xs:nonNegativeInteger
currentNrOfInstances	O	O	xs:nonNegativeInteger
connectionType	M	NP	senv:connectionTypeID
keyData	O	NP	xs:byte
keyInformation	O	O	xs:anyType
cipherSuite	O	O	dcfg:TLSCiphersuites

7.6.2 <secureConnection> Resource Procedures

7.6.2.1 CREATE <secureConnection>

This procedure shall be used for creating a <secureConnection> resource.

Table 7.6.2.1-1: <*secureConnection*> CREATE

<*secure Connection*> CREATE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID *Content:* The resource content shall provide the information as defined in clause 7.6.1
Processing at Originator before sending Request	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with the specific details for: *Content:* Address of the created <cipher> resource, according to clause 10.1.1.1.of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.1.1 of ETSI TS 118 101 [2]

If <*generateKey*> is created, the key to be used shall be generated and stored in *keyData*, in such a case *keyInformation* shall be filled with the public part of the generated key.

7.6.2.2 RETRIEVE <secureConnection>

This procedure shall be used for retrieving information about the <secureConnection> resource.

Table 7.6.2.2-1: <*secureConnection*> RETRIEVE

<*secureConnection*> RETRIEVE request message parameters	
Associated Reference Point	Mcs
Information in Request message	According to clause 10.1.2 of ETSI TS 118 101 [2] with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.2 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.2 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with specific details for: *Content:* Attributes of the <*secureConnection*> resources as defined in clause 7.6.1
Processing at Originator after receiving Response	According to clause 10.1.2 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.2 of ETSI TS 118 101 [2]

7.6.2.3 UPDATE <secureConnection>

This procedure shall be used for sending payload data via an established secure connection.

Table 7.6.2.3-1: <*secureConnection*> UPDATE

<*secureConnection*> UPDATE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID *Content:* attributes of the <cipher> resource which is to be updated as defined in clause 7.5.1
Processing at Originator before sending Request	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.3 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.3 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.3 of ETSI TS 118 101 [2]

7.6.2.4 DELETE <secureConnection>

This procedure shall be used for deleting a <*secureConnection*> resource. Deleting a <*secureConnection*> resource shall close an established secure connection between the originator (creator) and the destination.

Table 7.6.2.4-1: <*secureConnection*> DELETE

<*secureConnection*> DELETE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.4.1 of ETSI TS 118 101 [2]

7.6.3 <connectionInstance> resource

The <*connectionInstance*> resource represents a data instance in the <*secureConnection*> resource.

The <*connectionInstance*> resource inherits the same access control policies of the parent <*secureConnection*> resource, and does not have its own *accessControlPolicyIDs* attribute.

The services shall include the following:

- generation of key material within the secure environment that can be used for the establishment of a secure connection by the requesting entity (creator) outside of the secure environment;
- acting as secure connection endpoint and sending the data provided by the requesting entity (creator) from within the secure environment with the key material generated and kept inside the secure environment.

The <*connectionInstance*> resource shall contain the child resources specified in table 7.6.3-1.

Table 7.6.3-1: Child resources of <*connectionInstance*> resource

Child Resources of <*connectionInstance*>	**Child Resource Type**	**Multiplicity**	**Description**
algorithmSpecificParameter	*<algorithmSpecificParameter>*	0..1	See clause 7.5.1.5
connect	*<connect>*	1	See clause 7.6.5
send	*<send>*	1	See clause 7.6.6

Attributes in <*connectionInstance*> are shown in table 7.6.3-2.

Table 7.6.3-2: Attributes of <*connectionInstance*> resource

Attributes of <*connectionInstance*>	**Multiplicity**	**RW/ RO/ WO**	**Description**
resourceType	1	RO	Defines the resource type.
resourceID	1	RO	Defines an identifier for the resource. This attribute shall be provided by the creator. The creator shall assign a resourceID which is unique within its context.
resourceName	1	WO	This attribute is the name for the resource that is used for 'hierarchical addressing method' to represent the parent-child relationships of resources.
parentID	1	RO	This attribute is the *resourceID* of the parent of this resource.
expirationTime	1	RW	Time/date after which the resource will be deleted by the Hosting CSE.

Attributes of <*connectionInstance*>	Multiplicity	RW/ RO/ WO	Description
accessControlPolicyIDs	0..1 (L)	RW	Is used to control access to the resource. If no *accessControlPolicyIDs* are provided at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute.
creationTime	1	RO	Time/date of creation of the resource. The *creationTime* is set by the CSE hosting the SE when the resource is created.
lastModifiedTime	1	RO	Last modification time/date of the resource. This attribute is mandatory. The *lastModifiedTime* value is set by the Hosting CSE when the resource is created, and the *lastModifiedTime* value is updated when the resource is updated.
destinationURI	1	RW	Specifies the end point to which the secure connection shall be established.
outgoingPayloadData	0..1	RW	Contains the data that has to be sent via the established secure connection.
incomingPayloadData	0..1	RO	Contains the data received via the established secure connection.
negotiatedKey	0..1	RO	Contains the negotiated key e.g. the pairwiseE2EKey using TLS Exporter specification (IETF RFC 5705 [6]).
negotiatedCipherSuite	0..1	RO	Is the cipher suite negotiated between the Secure Environment and the remote entity.

Table 7.6.3-3: Data types of <connectionInstance> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
destinationURI	M	O	xs:anyURI
outgoingPayloadData	O	O	xs:byte
incomingPayloadData	NP	NP	xs:byte
negotiatedKey	O	NP	xs:byte
negotiatedCipherSuite	O	NP	dcfg:TLSCiphersuites

7.6.4 <connectionInstance> Resource Procedures

7.6.4.1 CREATE <connectionInstance>

This procedure shall be used for creating a <connectionInstance> resource.

Table 7.6.4.1-1: <*connectionInstance*> CREATE

<*connectionInstance*> CREATE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID *Content:* The resource content shall provide the information as defined in clause 7.6.3
Processing at Originator before sending Request	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with the specific details for: *Content:* Address of the created <connectionInstance> resource, according to clause 10.1.1.1.of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.1.1 of ETSI TS 118 101 [2]

7.6.4.2 RETRIEVE <connectionInstance>

This procedure shall be used for retrieving payload data from a communication partner or to retrieve the negotiated key.

Table 7.6.4.2-1: <*connectionInstance*> RETRIEVE

<*connectionInstance*> RETRIEVE request message parameters	
Associated Reference Point	Mcs
Information in Request message	According to clause 10.1.2 of ETSI TS 118 101 [2] with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.2 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.2 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with specific details for: *Content:* Attributes of the <*connectionInstance*> resources as defined in clause 7.6.3
Processing at Originator after receiving Response	According to clause 10.1.2 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.2 of ETSI TS 118 101 [2]

7.6.4.3 UPDATE <connectionInstance>

This procedure shall be used for sending payload data via an established secure connection.

Table 7.6.4.3-1: <*connectionInstance*> UPDATE

<*connectionInstance*> UPDATE request message parameters	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted M2M-AE-ID or CSE-ID *Content:* attributes of the <connectionInstance> resource which is to be updated as defined in clause 7.6.3
Processing at Originator before sending Request	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.3 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.3 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.3 of ETSI TS 118 101 [2]

7.6.4.4 DELETE <connectionInstance>

This procedure shall be used for deleting a <connectionInstance> resource. Deleting a <connectionInstance> resource closes an established secure connection between the originator (creator) and the destination.

Table 7.6.4.4-1: *<connectionInstance>* DELETE

***<connectionInstance>* DELETE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.4.1 of ETSI TS 118 101 [2]

7.6.5 <connect> Resource

The <connect> resource is a virtual resource because it does not have a representation. When a RETRIEVE request addresses the <connect> resource, a connection shall be established to the destination URI. If <negotiatedKey> exists the negotiated key shall be stored in this attribute.

The <connect> resource inherits access control policies that apply to the parent resource.

7.6.6 <send> Resource

The <send> resource is a virtual resource because it does not have a representation. When a RETRIEVE request addresses the <send> resource, the value of *outgoingPayloadData* shall be sent to the destination URI.

The <send> resource inherits access control policies that apply to the parent resource.

7.6.7 <generateKey> Resource

The <generateKey> resource is a virtual resource because it does not have a representation. When a RETRIEVE request addresses the <generateKey> resource, the *keyData* attribute shall be filled with a key generated according the *algorithm* attribute.

The <generateKey> resource inherits access control policies that apply to the parent resource.

7.7 Authentication and Identification

7.7.1 <identity> resource

Secure Environments shall provide a service to AEs or CSEs to establish an Identity and provide authentication of this Identity. The <*identity*> resource shall represent the services offered by the Secure Environment to enable the establishment of a secure Identity. The services include the following:

- generation of an Identity and associated key material within the secure environment;
- using the associated key material for authenticating the generated Identity.

The <*identity*> resource shall contain the child resources specified in table 7.7.1-1.

Table 7.7.1-1: Child resources of <*identity*> resource

Child Resources of <*identity*>	Child Resource Type	Multiplicity	Description
authenticate	<*authenticate*>	0..1	See clause 7.7.3
generateKey	<*generateKey*>	0..1	See clause 7.7.4

Attributes in <*Identity*> are shown in table 7.7.1-2.

Table 7.7.1-2: Attributes of <*identity*> resource

Attributes of <*identity*>	Multiplicity	RW/ RO/ WO	Description
resourceType	1	RO	Defines the resource type.
resourceID	1	RO	Defines an identifier for the resource. This attribute shall be provided by the creator. The creator shall assign a resourceID which is unique within its context.
resourceName	1	WO	This attribute is the name for the resource that is used for 'hierarchical addressing method' to represent the parent-child relationships of resources.
parentID	1	RO	This attribute is the *resourceID* of the parent of this resource.
expirationTime	1	RW	Time/date after which the resource will be deleted by the Hosting CSE.
accessControlPolicyIDs	0..1 (L)	RW	Is used to control access to the resource. If no *accessControlPolicyIDs* are provided at the time of creation, the *accessControlPolicyIDs* of the parent resource is linked to this attribute.
creationTime	1	RO	Time/date of creation of the resource. The *creationTime* is set by the CSE hosting the SE when the resource is created.
lastModifiedTime	1	RO	Last modification time/date of the resource. This attribute is mandatory. The *lastModifiedTime* value is set by the Hosting CSE when the resource is created,and the *lastModifiedTime* value is updated when the resource is updated.
idName	1	WO	Contains the name of the identity.
keyData	0..1	WO	Contains the value of a key.
idData	0..1	RW	Contains information associated to the identity and which is necessary for the authentication protocol. The detailed structure depends on the authentication protocol and could comprise among others public key material, protocol identifier, certificates.
originatorAuthenticationData	0..1	RW	Contains information provided by the Originator and which is necessary for the authentication protocol. The detailed structure depends on the authentication protocol and could comprise among others nonces, certificates, signatures.
receiverAuthenticationData	0..1	RO	Contains information provided by the Receiver and which is necessary for the authentication protocol. The detailed structure depends on the authentication protocol and could comprise among others nonces, certificates, signatures.

Table 7.7.1-3: Data types of <identity> resource specific attributes

Name	Request Optionality		Data type
	Create	Update	
idName	M	NP	xs:string
keyData	O	NP	xs:byte
idData	O	O	xs:anyType
originatorAuthenticationData	O	O	xs:anyType
receiverAuthenticationData	NP	NP	xs:anyType

7.7.2 <identity> Resource Procedures

7.7.2.1 CREATE <identity>

This procedure shall be used for creating a <identity> resource.

Table 7.7.2.1-1: *<identity>* CREATE

***<identity>* CREATE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID *Content:* The resource content shall provide the information as defined in clause 7.7.1
Processing at Originator before sending Request	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with the specific details for: *Content:* Address of the created <identity> resource, according to clause 10.1.1.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.1.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.1.1 of ETSI TS 118 101 [2]

If <*generateKey*> is created, the key to be used shall be generated and stored in keyData, in such a case keyInformation shall be filled with the public part of the generated key.

7.7.2.2 RETRIEVE <identity>

This procedure shall be used for retrieving the identity and retrieve authentication data according the used authentication protocol.

Table 7.7.2.2-1: *<identity>* RETRIEVE

***<identity>* RETRIEVE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	According to clause 10.1.2 of ETSI TS 118 101 [2] with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.2 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.2 of ETSI TS 118 101 [2]
Information in Response message	All parameters defined in table 8.1.3-1 of ETSI TS 118 101 [2] apply with specific details for: *Content:* Attributes of the <*identity*> resources as defined in clause 7.7.1
Processing at Originator after receiving Response	According to clause 10.1.2 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.2 of ETSI TS 118 101 [2]

7.7.2.3 UPDATE <identity>

This procedure shall be used for updating the <identity resource> and sending authentication data according the used authentication protocol.

Table 7.7.2.3-1: *<identity>* UPDATE

***<identity>* UPDATE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID *Content:* attributes of the <cipher> resource which is to be updated as defined in clause 7.5.1
Processing at Originator before sending Request	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.3 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.3 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.3 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.3 of ETSI TS 118 101 [2]

7.7.2.4 DELETE <identity>

This procedure shall be used for deleting an <*identity*> resource.

Table 7.7.2.4-1: *<identity>* DELETE

***<Identity>* DELETE request message parameters**	
Associated Reference Point	Mcs
Information in Request message	All parameters defined in table 8.1.2-3 of ETSI TS 118 101 [2] apply with the specific details for: *To:* contains M2M-SE-ID or SE hosted AE-ID or CSE-ID
Processing at Originator before sending Request	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Receiver	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Information in Response message	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Processing at Originator after receiving Response	According to clause 10.1.4.1 of ETSI TS 118 101 [2]
Exceptions	According to clause 10.1.4.1 of ETSI TS 118 101 [2]

7.7.3 <authenticate> Resource

The <authenticate> resource is a virtual resource because it does not have a representation. It is the child resource of an <identity> resource. When a RETRIEVE request addresses the <authenticate> resource, the *originatorAuthenticationData*, *keyData* and *idData* shall be used to calculate a value which shall be stored in *retrieverAuthenticationData*.

Depending on the authentication protocol it may be necessary to repeat the process of UPDATE *originatorAuthenticationData* , RETRIEVE <authenticate> and RETRIEVE *retrieverAuthenticationData* several times.

The <authenticate> resource inherits access control policies that apply to the parent <identity> resource.

7.7.4 <generateKey> Resource

The <generateKey> resource is a virtual resource because it does not have a representation. When a RETRIEVE request addresses the <generateKey> resource, the *keyData* attribute shall be filled with a key generated according the *algorithm* attribute.

The <generateKey> resource inherits access control policies that apply to the parent resource.

8 Physical Interface

The present release does not specify how the Mcs reference point can be bound to physical interfaces used by specific Secure Environments. Organizations specifying technologies applicable for Secure Environments, such as GlobalPlatform, specify bindings that can be applicable.

9 Resource type definitions for the Mcs reference point

9.1 Mcs specific enumeration values of m2m:resourceType

The following values are defined specifically for the Mcs interface, as an extension to applicable values inherited from ETSI TS 118 104 [14].

Table 9.1-1: Mcs enumeration values

Value	Resource type
20001	algorithmSpecificParameter
20002	cipher
20003	connectionInstance
20004	hash
20005	identity
20007	rand
20008	secureConnection
20009	sensitiveDataObject
20010	SEReboot
20011	SE
20012	signature

9.2 senv:SEType

The values in table 9.2-1 are defined as per table 7.2.1-2.

Table 9.2-1: SEType defined values

Value	Meaning
1	UICC as per ETSI
2	GlobalPlatform Secure Element
3	TEE as per GlobalPlatform
4	Software cryptographic library

9.3 senv:securityLevel

The values in table 9.3-1 are defined as per clause 6.2.

Table 9.3-1: securityLevel defined values

Value	Meaning
1	Low security
2	Medium security
3	High security

9.4 senv:rebootType

The values in table 9.4-1 are defined.

Table 9.4-1: rebootType defined values

Value	Meaning
1	Cold Reset as defined in ISO/IEC 7816-3 [7]
2	Warm Reset as defined in ISO/IEC 7816-3 [7]

9.5 senv:cipherLabel

The values in table 9.5-1 are defined.

Table 9.5-1: cipherLabel defined values

Value	Meaning
1	EXPORTER-oneM2M-Bootstrap
2	EXPORTER-oneM2M-Connection

9.6 senv:cipherAlgorithm

The values in table 9.6-1 are defined.

Table 9.6-1: cipherAlgorithm defined values

Value	Meaning
1001	ALG_AEAD_AES_128_GCM
1002	ALG_AEAD_AES_256_GCM
1003	ALG_AEAD_AES_128_CCM
1004	ALG_AEAD_AES_256_CCM
1018	ALG_AEAD_AES_128_CCM_8
1019	ALG_AEAD_AES_256_CCM_8
13	ALG_AES_BLOCK_128_CBC_NOPAD
22	ALG_AES_CBC_ISO9797_M1
23	ALG_AES_CBC_ISO9797_M2
24	ALG_AES_CBC_PKCS5

NOTE: Values for AEAD algorithms were taken from IANA with an offset of 1000. Values for other algorithms were taken from JavaCard API.

9.7 senv:rngType

The values in table 9.7-1 are defined.

Table 9.7-1: rngType defined values

Value	Meaning
1	Pseudo RNG
2	True (physical) RNG

9.8 senv:hashAlgorithm

The values in table 9.8-1 are defined.

Table 9.8-1: hashAlgorithm defined values

Value	Meaning
4	SHA256
5	SHA384
6	SHA512

NOTE: These values were taken from the JavaCard API.

9.9 senv:signatureAlgorithm

The values in table 9.9-1 are defined.

Table 9.9-1: signatureAlgorithm defined values

Value	Meaning
49	ALG_AES_CMAC_128
18	ALG_AES_MAC_128_NOPAD
33	ALG_ECDSA_SHA_256
34	ALG_ECDSA_SHA_384
38	ALG_ECDSA_SHA_512
25	ALG_HMAC_SHA_256
26	ALG_HMAC_SHA_384
27	ALG_HMAC_SHA_512

NOTE: These values were taken from the JavaCard API.

9.10 senv:connectionTypeID

The values in table 9.10-1 are defined.

Table 9.10-1: connectionTypeID defined values

Value	Meaning
1	TLS
2	DTLS
3	SMS
4	E2EKey

10 Short Name definitions for the Mcs reference point

10.1 Short Names for Mcs specific resource attributes

The mapping between the full names and their shortened form is given in the following clauses.

These names are case-sensitive. A oneM2M implementation shall use the letter casing given in these clauses.

In protocol bindings, resource attributes names shall be translated into short names shown in table 10.1-1. All attributes name not mentioned in this table that match defined attributes in ETSI TS 118 104 are assumed to reuse the short names defined in ETSI TS 118 104 [14].

Table 10.1-1: Mcs Resource attribute short names

Attribute Name	Occurs in	Short Name
SEType	SE	***seT***
securityLevel	SE	***seL***
rebootType	SEReboot	***rbT***
SEReboot	SEReboot	***rb***
label	algorithmSpecificParameter	***Clab***
algorithm	cipher	***Calg***
rngType	rand	***rgT***
randomData	rand	***rndD***
requestedDataSize	rand	***Dsz***
algorithm	hash	***Halg***
algorithm	signature	***Salg***
connectionType	secureConnection	***cnT***
idName	identity	***idN***
keyData	identity, secureConnection, signature, cipher	***kDt***
idData	identity	***idDt***
originatorAuthenticationData	identity	***oAD***
receiverAuthenticationData	identity	***rAD***
destinationURI	connectionInstance	***dst***
outgoingPayloadData	connectionInstance	***oD***
incomingPayloadData	connectionInstance	***iD***
negotiatedKey	connectionInstance	***ngK***
negotiatedCipherSuite	connectionInstance	***ngCS***
maxNrOfInstances	secureConnection	***mni***
currentNrOfInstances	secureConnection	***cni***
connectionType	secureConnection	***cnT***
keyInformation	secureConnection, signature	***kInf***
cipherSuite	secureConnection	***aCS***
message	signature, hash	***msg***
signature	signature	***Sgn***
verificationResult	signature	***vR***
hashValue	hash	***Hv***
initialVector	algorithmSpecificParameter	***iV***
nonce	algorithmSpecificParameter	***nc***
associatedData	algorithmSpecificParameter	***aD***
sensitiveData	cipher, sensitiveData	***msg***
currentByteSize	cipher, sensitiveData	***cbs***
maxByteSize	cipher	***mbs***
calculatedData	cipher	***cD***
m2mSeID	SE	***sID***
supportedResourceType	SE	***srt***
hostedCSELink	SE	***hcl***
hostedAELinks	SE	***hal***
e2eSecInfo	SE	***esi***

10.2 Short Names for Mcs specific resource types

In protocol bindings resource type names shall be translated into short names of table 10.2-1.

Table 10.2-1: Mcs Resource type short names

Resource Type Name	Short Name
SE	***Senv***
SEReboot	***Srbt***
sensitiveDataObject	***Sdo***
cipher	***Cph***
encrypt	***Enc***
decrypt	***Dec***
generateKey	***gnK***
algorithmSpecificParameter	***algP***
rand	***Rnd***
generateRand	***gnR***
hash	***Hsh***
calculateHash	***cHsh***
signature	***Sgn***
calculateSignature	***cSgn***
verifySignature	***vSgn***
secureConnection	***Ssc***
connectionInstance	***Isc***
connect	***cnt***
send	***snd***
identity	***Sidn***
authenticate	***Sauth***

History

Document history		
V4.0.1	November 2023	Publication