ONEM2M
TECHNICAL SPECIFICATION

Document Number	oneM2M-TS-0003-Security_Solutions-V-2014-08
Document Name:	oneM2M Security Solutions
Date:	2014-08-01
Abstract:	Specification of oneM2M security solutions.

About oneM2M

The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide.

More information about oneM2M may be found at: http//www.oneM2M.org

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. NO oneM2M PARTNER TYPE 1 SHALL BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. oneM2M EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

Contents

1 Scope

The present document defines security solutions applicable within the M2M system.

2 References

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

2.1 Normative references

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

[1] oneM2M TS-0001 "oneM2M Architecture"

[2] Open Mobile API specification V2.0.2

[3] GP TEE Client API

[4] oneM2M TS-0004 “oneM2M Core Protocol”

[5] IETF RFC 5246 "The Transport Layer Security (TLS) Protocol Version 1.2".

[6] IETF RFC 6347 "Datagram Transport Layer Security Version 1.2"

[7] ETSI TS 102 225 (V11.0.0) "Smart Cards; Secured packet structure for UICC based applications (Release 11)" URL:http://www.etsi.org/

[8] ETSI TS 102 226 (V11.0.0) "Smart Cards; Remote APDU structure for UICC based applications (Release 11)" URL:http://www.etsi.org/

[9] 3GPP TS 31.115 (V10.1.0) "Remote APDU Structure for (U)SIM Toolkit applications (Release 10)"

[10] 3GPP TS 31.116 (V10.2.0) "Remote APDU Structure for (Universal) Subscriber Identity Module (U)SIM Toolkit applications (Release 10)"

[11] 3GPP2 C.S0078-0 (V1.0) "Secured packet structure for CDMA Card Application Toolkit (CCAT) applications"

[12] 3GPP2 C.S0079-0 (V1.0) "Remote APDU Structure for CDMA Card Application Toolkit (CCAT) applications"

[13] 3GPP TS 33.220 "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) (3GPP TS 33.220)"

[14] 3GPP2 S.S0109-A "Generic Bootstrapping Architecture (GBA) Framework"

[15] IETF RFC 4279 "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS) "

[16] IETF RFC 5246 "The Transport Layer Security (TLS) Protocol, Version 1.2"

[17] IETF RFC 6347 "Datagram Transport Layer Security Version 1.2"

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

[19] IETF RFC 3629 "UTF-8, a transformation format of ISO 10646".

[20] "Unicode Standard Annex #15; Unicode Normalization Forms", Unicode 5.1.0, March 2008. http://www.unicode.org

[21] GlobalPlatform Device Technology TEE Administration framework, DRAFT

[22] GlobalPlatform Device Technology TEE System Architecture, Version 1.0

[23] ETSI TS 102 671 "Smart Cards; Machine-to-Machine UICC; Physical and logical characteristics" URL:http://www.etsi.org/

[24] ETSI TS 102 221 "Smart Cards; UICC-Terminal Interface; Physical and logical characteristics" URL:http://www.etsi.org/

[25] ETSI TS 102 484 "Smart Cards; Secure channel between a UICC and an end-point terminal" URL:http://www.etsi.org/

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

[27] ETSI TS 101 220 "Smart Cards; ETSI numbering system for telecommunication application providers" URL:http://www.etsi.org/

[28] 3GPP TS 33.222, " Generic Authentication Architecture (GAA), Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)", Rel-12

[29] 3GPP TS 24.109, "Bootstrapping interface (Ub) and network application function interface (Ua)", Rel-12

[30] 3GPP TS 29.109, "Protocols details Generic Authentication Architecture (GAA); Zh and Zn Interfaces based on Diameter protocol; Stage 3", Rel-12

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

[32] IETF RFC 5289 "TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM) "

[33] IETF RFC 2014 "HMAC: Keyed-Hashing for Message Authentication"

[34] IETF RFC 5280 "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”

[35] IETF RFC 6960 "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP"

[36] IETF RFC 6961 "The Transport Layer Security (TLS) Multiple Certificate Status Request Extension"

[37] IETF RFC 7250 “Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)”

[38] IETF RFC 7252 “The Constrained Application Protocol (CoAP)”

[39] RECOMMENDED ELLIPTIC CURVES FOR FEDERAL GOVERNMENT USE, National Institute of Standards and Technology, July 1999. http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf

[40] IETF RFC 6920 “Naming Things with Hashes”

[41] IETF RFC 3548 "The Base16, Base32, and Base64 Data Encodings".

[42] IETF RFC 5487 “Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode”

[43] IETF RFC 4492 “Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)”

[44] IETF RFC 6066 "Transport Layer Security (TLS) Extensions: Extension Definitions”

[45] IETF RFC 7251 "AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)"

[46] IETF RFC 5480 "Elliptic Curve Cryptography Subject Public Key Information",

2.2 Informative references

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

[i.1] oneM2M Drafting Rules

NOTE: Available at http://member.onem2m.org/Static_pages/Others/Rules_Pages/oneM2M-Drafting-Rules-V1_0.doc.

[i.2] oneM2M-TR-0004 Definitions and abbreviations

[i.3] 3GPP TR 33.868 V0.13.0; Security aspects of Machine-Type and other Mobile Data Applications Communications Enhancements; (Release 12)

[i.4] oneM2M TR-0008, Security Analysis Technical Report

[i.5] eXtensible Access Control Markup Language (XACML) Version 3.0. 22 January 2013. OASIS Standard.

[i.6] Handbook of Applied Cryptography, A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, CRC Press, 1996

[i.7] Recommendation ITU T X.509, Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks, 10/2012.

[i.10] 3GPP TR 33.868 "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on security aspects of Machine-Type Communications (MTC) and other mobile data applications communications enhancements".

[i.11] OMA-TS-REST-NetAPI_TerminalLocation-V1_0-20130924-A: "RESTful Network API for Terminal Location", Version 1.0.

[i.12] ISO 3166-1:2013, "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes".

[i.13] ISO/IEC 7816-5: "Identification cards - Integrated circuit cards - Part 5: Registration of Application Providers".

[i.14] Guide to Attribute Based Access Control (ABAC) Definition and Considerations, NIST Special Publication 800-162 (http://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.sp.800-162.pdf)

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in [i.2] and the following apply:

AE-ID Certificate: A certificate with a certificate chain to a trust anchor certificate and containing an AE-ID in the subjectAltName extension. An AE_ID certificate can be used to verify that an entity has been assigned the AE-ID in the certificate.

association configuration: phase of a Security Association Establishment Framework in which the entity establishing the Security Association (and the Central Key Distribution Server, in the case of Centralized Security Frameworks), are

provided with identities (and any other relevant credentials) to ensure that the security association is established between the intended entities

association security handshake: phase of a Security Association Framework in which the security association endpoints perform mutual authentication

bootstrap credential: pre-provisioned credential enabling mutual authentication of the Enrolee and the M2M Enrolment function

bootstrap credential configuration: phase of a Security Bootstrap Framework in which the Bootstrap Credentials are pre-provisioned to the Enrolee and the M2M Enrolment function

bootstrap enrolment handshake: phase of a Security Bootstrap Framework in which the Enrolee and M2M Enrolment Function perform mutual authentication

bootstrap instruction configuration: phase of a Security Bootstrap Framework in which the Enrolee and M2M Enrolment Function are provided with identities (and any other relevant credentials) to enable the M2M Enrolment function to establish a Master Credential between the intended Enrolee and M2M Authentication Function

bootstrap server function [13]: BSF is hosted in a network element under the control of a Mobile Network Operator. BSF, HSS, and UEs participate in GBA in which a shared secret is established between the network and a UE by running the bootstrapping procedure

NOTE: The shared secret can be used between NAFs and UEs, for example, for authentication purposes.

bootstrapping transaction identifier [13]: bootstrapping transaction identifier (B-TID) is used to bind the subscriber identity to the keying material in GBA reference points Ua, Ub and Zn

CA-Certificate [i.8]: certificate created by one certification authority (CA) certifying the public key of another CA

central key distribution server: server which can perform mutual authentication with a set of entities, and which can use this ability to securely distribute keys to sets of two or more of these entities

central key distribution server handshake: Phase of a Centralized Security Association Establishment Framework in which an entity and the Centralized Key Distribution Server perform mutual authentication and generate a Symmetric Key which can then be used in the Association Security Handshake for mutual authentication between that entity and other entities

centralized security framework: Security Framework in which the entities authenticate each other with the assistance of a Central Key Distribution Server

certificate: See Public Key Certificate.

certificate chain: sequence of one or more CA-certificates, where: the Public Verification Key in each CA-certificate is certified in the previous CA-certificate; and the public key of the first CA-Certificate is trusted *a priori*

NOTE: Trust in the public key in each CA-certificate can be based on trust in the previous CA-Certificate.

certificate name: unique identifier in a name field of a Certificate (e.g. in the X.509 "Subject" or "Subject Alternative Name" attribute)

certificate verification: process necessary to trust an entity's Certificate

certification authority [i.8]: responsible for establishing and vouching for the authenticity of public keys

NOTE: [This] includes binding public keys to distinguished names through signed certificates, managing certificate serial numbers, and certificate revocation.

credential configuration: phase of a Security Association Establishment Framework in which the Credentials necessary for the Security Association Establishment Framework are configured to the relevant entities and functions

CSE-ID certificate: A certificate with a certificate chain to a root of trust and containing a CSE-ID in the subjectAltName extension. A CSE_ID certificate can be used to verify that an entity has been assigned the CSE-ID in the certificate.

device certificate: A certificate with a certificate chain to a root of trust and containing at least one globally unique hardware instance identifier in the subjectAltName extension. A device certificate can be used to verify that an entity is executing on the identified hardware instance.

digital signature [i.9]: information is signed by appending to it an enciphered summary of the information

NOTE: The summary is produced by means of a one-way hash function, while the enciphering is carried out using the private key of the signer.

direct security framework: Security Framework in which the entities authenticate each other directly, without assistance from a Central Key Distribution Server

enrolee: AE or CSE that requires remote provisioning of a symmetric key to be shared with an enrolment target

enrolment key: symmetric key established between an Enrolee and M2M Enrolment Function following successful mutual authentication

NOTE: A symmetric key to be shared by the Enrolee and an Enrolment Target may be derived (at the Enrolee and M2M Enrolment Function) from the currently valid Enrolment Key, and the M2M Enrolment Function subsequently securely delivers the symmetric key to the Enrolment Target.

enrolment key generation: phase of remote security provisioning Framework in which the Enrolee and M2M Enrolment function establish an Enrolment Key and Enrolment Key identifier

enrolment phase: The step in the lifecycle of an M2M equipment where it becomes provisioned for operation with a specific M2M Service Provider.

enrolment target: A M2M Authentication Function, CSE, or AE with whom an Enrolee wishes to establish a symmetric key (master credential or pre-provisioned secure connection key) using remote security provisioning

entity identifier: CSE-ID (or AE-ID respectively) of a CSE (or AE respectively)

FQDN certificate: A certificate with a certificate chain to a root of trust and containing an FQDN. These are the certificates that are commonly used to authentication web servers. This name has been used to distinguish this flavour of certificate from other flavour of certificates.

generic bootstrap architecture: set of 3GPP and 3GPP2 specifications providing security features and a mechanism to bootstrap authentication and key agreement for application security from the 3GPP and 3GPP2 underlying network authentication mechanisms

message integrity code: tag computed from a message and a symmetric key, and attached to a message

NOTE 1: The purpose of a messages integrity code is to facilitate, without the use of any additional mechanisms, assurances regarding both the source of a message and its integrity.

NOTE 2: A Message Integrity Code is sometimes called a "Message Authentication Code" - we have used "Message Integrity Code" since the abbreviation of "Message Authentication Code" (MAC) might be misunderstood to refer to "Media Access Control". The definition is based on text from [i.8] (p323).

M2M secure connection key: key shared between two CSEs of M2M Nodes (e.g. ASN/MN-CSE and IN-CSE) in order to secure the communication between those two entities

NOTE: This M2M Secure Connection Key results from a successful M2M Security Association Establishment procedure.

master credentials: Credentials used to mutually authenticate between an ASN/MN-CSE and the MAF. This is done to secure access to the infrastructure of an M2M Service Provider

NOTE: The Master Credentials are either pre-provisioned or remotely provisioned (without relying on those credentials).

Online Certificate Status Protocol: A protocol for requesting a report on the status of one or more X.509 certificates [RFC6960]

operational phase: The period in the lifecycle of an M2M equipment where it is actually used for providing M2M services.

policy decision point [i.7]: system entity that evaluates applicable policy and renders an authorization decision

policy enforcement point [i.7]: system entity that performs access control, by making decision requests and enforcing authorization decisions

policy information point [i.7]: system entity that acts as a source of attribute values

policy retrieval point: system entity that retrieves applicable policy or policy set

pre-provisioned secure connection key: Symmetric Key that is pre-provisioned to two entities (which may be AEs or CSEs) to be used for mutual authentication of those entities in Security Association Establishment

pre-provisioned secure connection key identifier: Identifier for a Pre-Provisioned Secure Connection Key

pre-provisioned symmetric enrolee key: Symmetric Key that is pre-provisioned to the Enrolee and M2M Enrolment Function

pre-provisioned symmetric enrolee key identifier: Identifier for a Pre-Provisioned Symmetric Enrolee Key

private signing key: secret key that can generate signatures that can be verified using a corresponding Public Verification Key

public key certificate: electronic document that uses a digital signature to bind a public key with an identity

NOTE: **[i.8]** A *public-key certificate* is a data structure consisting of a data part and a signature part. The data part contains cleartext data including, as a minimum, a public [verification] key and a string identifying the part (subject entity) to be associated therewith. The signature part consists of the digital signature of a certification authority over the data part, thereby binding the subject entity's identity to the specified public key.

public key certificate flavour: A name describing the usage of a public key certificate within the scope of oneM2M.

public key infrastructure: set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke Public Key Certificates. For more details, see [i.8].

public verification key: Credential that can verify digital signatures generated by a corresponding Private Signing Key, but which cannot be used to generate digital signatures

raw public key certificate: A certificate comprising only the SubjectPublicKeyInfo structure of an X.509 certificate that carries the parameters necessary to describe the public key [37]

relative enrolment key identifier: part of the enrolment key identifier that is unique within the context of a M2M Enrolment Function.

security association establishment: sequential processing of credential configuration, association configuration and association security handshake between two entities. Credential configuration and/or association configuration can not be performed if those steps have already been executed before.

security association establishment framework: Security Framework for Security Association Establishment

security bootstrap framework: Security Framework for Remote security provisioning: a mechanism for remotely provisioning a Master Credential and Master Credential Identifier to a Enrolee and an M2M Authentication Function

security framework: set of procedures providing Security Association Establishment or Remote security provisioning

self-signed certificate: Public Key Certificate that is signed by the same entity whose identity it certifies

symmetric key: secret key that is shared between two entities

trust anchor certificate: a certificate that is trusted a priori

X.509: ITU-T recommendation for a Public Key Infrastructure

3.2 Symbols

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

||	Concatenation

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in [i.2] and the following abbreviations apply:

ACL	Access Control List
ACP	AccessControlPolicy Instance
API	Application Programming Interface
ASN	Application Service Node
Authn	Authentication
BSF	Bootstrapping Server Function
B-TID	Bootstrapping Transaction Identifier
CA	Certification Authority
DTLS	Datagram Transport Layer Security (Protocol)
Enrolee-ID	Enrollee Identity
FQDN	Fully Qualified Domain Name
GBA	Generic Bootstrapping Architecture
GBA_ME	ME-based GBA
GBA_U	GBA with UICC-based enhancements
GUSS	GBA User Security Settings
HLR	Home Location Register
HSS	Home Subscriber System
HW	Hardware
IdA	Identifier for entity A
IdB	Identifier for entity B
Kc	M2M Secure Connection Key
KcId	M2M Secure Connection Key identifier
Ke	Enrolment Key
KeId	Enrolment Key Identifier
Km	Master Credential
KmId	Master Credential Identifier
Kpm	pre-provisioned credential for Master Credential provisioning
KpmId	pre-provisioned credential for Master Credential provisioning Identifier
Kpsa	pre-provisioned credential for M2M Security Association Establishment
KpsaId	pre-provisioned credential for M2M Security Association Establishment Identifier
Ks	temporary Key material referred to in GBA
Ks_(ext/int)_NAF	Derived key in GBA_ME or Derived key in GBA_U which remains on UICC
Ks..NAF	Abbreviation of Ks_(int/ext)_NAF
Ks_NAF	Derived key in the ME
Ks_ext_NAF	Derived key in GBA_U sent to the ME
Ks_int_NAF	Derived key in GBA_U which remains on UICC
M-TID	MAF Transaction Identifier
MAF	M2M Authentication Function
MAF-ID	M2M Authentication Function Identifier
MEF	M2M Enrolment Function
MIC	Message Integrity Code
MN	Middle Node
NAF	Network Application Function
OCSP	Online Certificate Status Protocol
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PIP	Policy Information Point
PKI	Public Key Infrastructure
PMSCK	Provisioned M2M Secure Connection Key
PRP	Policy Retrieval Point
RBAC	Role Based Access Control
RSPF	Remote Security Provisioning Framework

SAEF	Security Association Establishment Framework
SE	Secure Environment
SW	Software
TEE	Trusted Execution Environment
TLS	Transport Layer Security (Protocol)
(D)TLS-PSK	(D)TLS Pre-Shared Key (ciphersuites)
UE	(3GPP) User Equipment
USS	User Security Settings
URI	Uniform Resource Identifier

4 Conventions

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

5 Security Architecture

5.1 Overview

Figure 5.1-1 provides a high level overview of the Security architecture.

The architecture consists of following layers:

- Security Functions layer:
  - This layer contains a set of security functions that are exposed at reference point Mca and Mcc. These security functions can be classified into six categories; they are Identification, Authentication, Authorization, Security Association, Sensitive Data Handling and Security Administration.
- Security Environment Abstraction Layer:
  - This layer implements various security capabilities such as key derivation, data encryption/decryption, signature generation/verification, security credential read/write from/to the Secure Environments, and so on. The security functions in the Security Functions Layer invoke these functions in order to do the operations related to the Secure Environments. In addition this layer also provides physical access to the Secure Environments. Implementation of this is out of scope of the present document. This layer is not specified in the initial release but is expected to be considered in future releases.
- Secure Environment layer:
  - This layer contains one or multiple secure environments that provide various security services related to sensitive data storage and sensitive function execution. The sensitive data includes SE capability, security keys, local credentials, security policies, identity information, subscription information, and so on. The sensitive functions include data encryption, data decryption, and so on. Implementation of secure environments is out of scope of the present document.

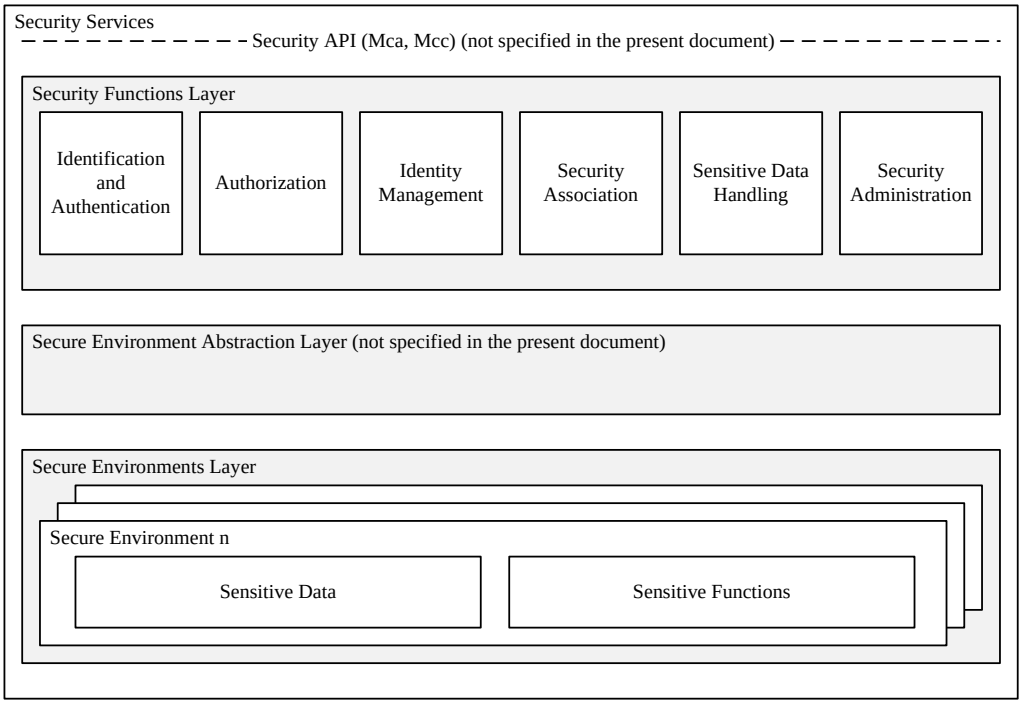


Figure 5.1-1: High level overview of the Security architecture

Design principles:

- Security Services are modular and configurable according to the needs of the hosting CSE, its supported reference points and its purpose.
- The architecture is split into several components and sub-components providing a modular design. With this design, mapping of the architecture to different nodes and entities is enabled.
- Depending on the requirements of each entity, Security should consist of components relevant to fulfil the requirements of the respective node or entity and the intended use case.
- The architecture may need to be adapted to be suitable for implementation in different entities. For example, the architecture can be mapped to different device classes.
- The security administration component shall enable administration of all sensitive resources (data and functions) and shall also allow configuration and extension of Security services itself.
- The Secure Environment within the CSE is accessed via the Secure Environment Abstraction layer and shall hold all sensitive resources.

5.1.1 Identification and Authentication

The Identification and Authentication function is in charge of identification and mutual authentication of CSEs and AEs.

Identification is the process of checking if the identity provided for authentication is valid. How to perform an identification process will depend on the purpose of authentication. For example, in the case of resource access, the authentication function may require the identification to check if the AE or CSE has registered with the local CSE; in the case of AE or CSE registration, the authentication function may require the identification to check if the identity provided by an AE or CSE fits a certificate. Once passing this checking process, the AE or CSE is identified, and the identified identity will be supplied to authentication process.

Authentication is the process of validating if the identity supplied in the identification step is associated with a trustworthy credential. How to perform an authentication process will depend on using which mutual authentication mechanism. For example, in the case of using certificate based authentication mechanism, the authentication function may require the authentication to verify a digital signature; in the case of using symmetric key based authentication mechanism, the authentication function may require the authentication to verify a Message Authentication Code (MAC). When this validating process has been completed, the AE or CSE is authenticated.

5.1.2 Authorization

The Authorization function is responsible for authorizing services and data access to authenticated entities according to provisioned access control policies and assigned roles.

Access control policy is defined as sets of conditions that define whether entities should be permitted access to a protected resource. The authorization function may support different authorization mechanisms, such as Access Control List (ACL), Role Based Access Control (RBAC), etc. The Authorization function may need to evaluate multiple access control policies in an authorization process in order to get a finial access control decision. This process is further described in clause 7 “Authorization”.

Authorization evaluation process is based on the Service Subscription resource which specifies what M2M Services and M2M Service roles the authenticated entity has subscribed to and the access control policies associated with the protected resource. The authorization evaluation process may also need to consider contextual attributes such as time or geographic location.

Prior to authorization mutual authentication between the originator CSE or AE and hosting CSE shall be performed.

5.1.3 Identity Management

The Identity Management function provides oneM2M identities/identifiers to the requesting entity in case those identities are stored within the secure environment. oneM2M identifiers as defined in the oneM2M Architecture [1] may also be treated as sensitive data that are accessible to AEs or CSEs and used independently of Authentication or Authorization functions.

5.2 Security Layers

5.2.1 Security Service Layer

The security service layer provides the following services:

- Access Management
  - Authorization
  - Authentication
  - Access Control
- Sensitive Data Handling
  - Sensitive Functions protection
  - Secure Storage
- Security Association Establishment
  - Secure Connection via secure session establishment

- Secure Connection via object security

- Security Administration (including remote security provisioning)
- Identity Protection

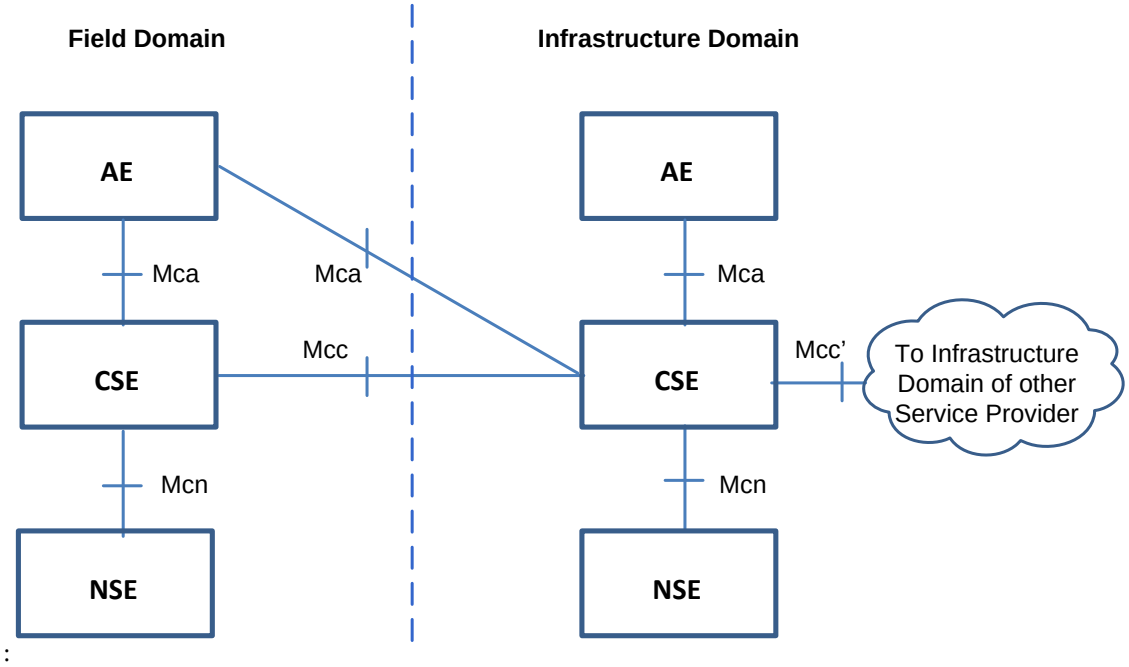
Each of these services provides functions and resources on the Security Service and Administration API.

5.2.2 Secure Environment Abstraction Layer

The Secure Environment Abstraction Layer (not specified in the present document) provides access to the Secure Environment via a general Security Transport API. A Plug-in associated to the type of Secure Environment shall provide physical/logical connectivity to the secure environment. The Secure Environment Abstraction Layer shall also be accessible on the Service Layer.

5.3 Integration within overall oneM2M architecture

Security services may be provided within the following architectural components and interacts on the different reference points as described in TS-0001[1].



:

Figure 5.2.1-1: oneM2M Functional Architecture

The present document addresses the security over Mcc and Mca in hop-by hop scenario

6 Security Services and Interactions

6.1 Security Integration in oneM2M flow of events

This clause specifies the integration of security process and procedures during deplotyment and operation of a oneM2M solution.

6.1.1 Interactions between layers

Before any M2M Common Services layer procedure can take place, connectivity has to be established in the underlying Network Services Layer, which may involve independent provisioning and service registration procedures specified by the underlying network.

The Service Layer Security provisioning (security pre-provisioning or security bootstrapping) and Security Association Establishment procedures specified in the present document can take place independently (and generally consecutively) from any required Network Service Layer connectivity establishment procedures.

Finally, the security provisioning and security association establishment requirements imposed by M2M Application Service Providers have to be accounted for. The security association establishment results in a TLS or DTLS session which protects messages being exchanged between adjacent AE/CSE.

6.1.2 High level sequence of events

6.1.2.1 Enrolment phase

M2M equipments typically require provisioning and configuration phases before being put in actual operation. This may be performed by a pre-provisioning that can be integrated in the manufacturing or product deployment phase, or by means of a security bootstrapping procedure (i.e. remote security provisioning) that takes place before the equipment starts actual operation.

At the service layer level, such provisioning and configuration requires selection of the stakeholder that will provide services through the equipment, especially the M2M Service Provider. This Enrolment phase requires contractual agreements between the stakeholders.

Enrolment phase may occur several times during the lifecycle of an M2M equipment, but is only repeated when a change in the Service Provider affects the provisioning or configuration of the equipment.

The security provisioning phase for the different layers can be combined using a common method of security pre-provisioning.

Remote Security Provisioning Frameworks (RSPF) provide post-provisioning of the essential information to establish a security association between a Field Domain entity and the M2M Authentication Function of a chosen M2M Service Provider. The essential security information includes the security credentials and identifiers. Remote Security Provisioning procedures rely on an M2M Enrolment Function which can be external to the M2M Service Provider to establish appropriate credentials.

- **Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework:** A symmetric key is pre-provisioned to the Enrolee and M2M Enrolment Function for the mutual authentication of those entities. For more details, see clause 8.3.2.1.
- **Certificate-Based Remote Security Provisioning Framework:** The Enrolee and M2M Enrolment Function are each issued and authenticate themselves with private signing keys and Certificates containing the corresponding Public Verification Key. For more details see clause 8.3.2.2.
- **GBA-based Remote Security Provisioning Framework**. In this case, the M2M Enrolment Function includes the functionality of a GBA Bootstrap Server Function. This framework uses 3GPP or 3GPP2 symmetric keys to authenticate the Enrolee and the M2M Enrolment Function (which is also a GBA BSF). The details are specified by 3GPP TS 33.220 [13] and 3GPP2 S.S0109-A [14]. For more details see clause 8.3.2.3.

Figure 6.1.2.1-1 illustrates the different Remote Security Provisioning Frameworks. Note there is no communication between M2M Entities A and B in the Remote Security Provioning procedure. After successful completion of the Remote Security Provisioning procedure, a Security Association Establishment procedure is applied.

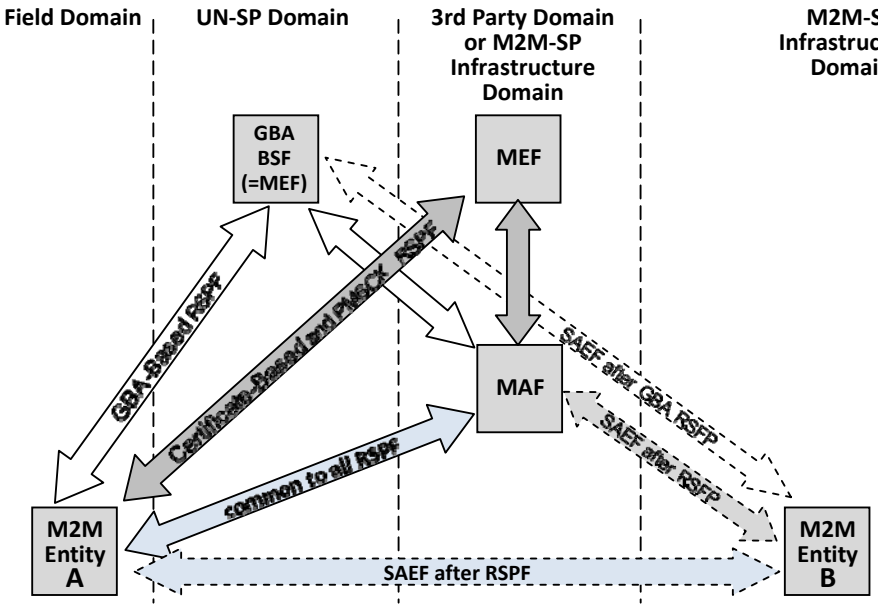


Figure 6.1.2.1-1: Entities involved in Remote Security Provisioning

6.1.2.2 Operational phase

6.1.2.2.1 M2M Service Access

AEs and CSEs seeking access to M2M services need to be mutually identified and authenticated with the M2M Service Infrastructure, in order to provide protection from unauthorized access and Denial of Service attacks. This mutual authentication enables to additionally provide encryption and integrity protection for the exchange of messages across a single Mca, Mcc or Mcc' reference point.

This is the purpose of the Security Association Establishment procedure, which shall take place before execution of the service related procedures specified in TS-0001 [1] for the corresponding reference point.

On the Mca and Mcc reference points, security association establishment between a field domain AE or CSE, respectively, and an IN-CSE is mandatory.

On the Mcc' reference point, security association establishment between IN-CSE and IN-CSE is mandatory.

On the Mca reference point, security association establishment between AE and the CSE in the field domain is strongly recommended.

The security association establishment phase of the M2M Service Layer and M2M Application Layer are generally independent from similar procedures that may be required by the Network Layer, though they may rely on the security services provided by the Network Layer.

The oneM2M system supports the following authentication mechanisms for Security Association Establishment, described in more detail in clause 8.2.1 "Overview on Security Association Establishment Frameworks":

- **Provisioned M2M Secure Connection Key Security Association Establishment Framework**: A symmetric key is pre-provisioned to the Security Association end-points. For more details see clause 8.2.2.1.
- **Certificate-Based Security Association Establishment Framework**: Security Association end-points authenticate themselves using private signing keys and Certificates containing the corresponding Public Verification Key. For more details see clause 8.2.2.2.
- **M2M Authentication Function (MAF) Security Association Establishment Framework**. For MAF-based SAEF, the centralized key distribution server is a MAF hosted either by a 3rd party service provider which has a service relationship with the M2M Service Provider (M2M-SP), or hosted by the M2M-SP itself. The MAF

authenticates a Field Domain entitiy on behalf of an IN-CSE using a symmetric key. For more details see clause 8.2.3.1.

- **GBA-Based Security Association Establishment Framework**, where the centralized key distribution server is a Bootstrap Server Function (BSF) of the 3GPP Generic Bootstrap Architecture (GBA) hosted by the Underlying Network Service Provider. 3GPP or 3GPP2 symmetric keys are used to authenticate Field Domain entities. The 3GPP-defined communication protocols as specified in 3GPP TS 33.222 [28], 3GPP TS 24.109 [29] and 3GPP TS 29.109 [30] are employed. The details are specified by 3GPP TS 33.220 [13] and 3GPP2 S.S0109-A [14], see clause 8.2.3.2.

Figure 6.1.2.2-1 illustrates the different use cases and entities involved in the various Security Association Establishment Frameworks (SAEF) considered in this specification.

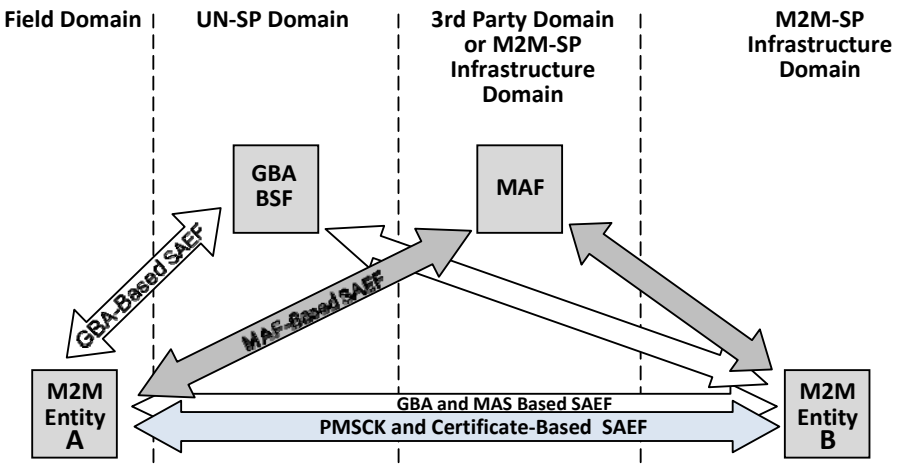


Figure 6.1.2.2-1: Entities involved in Security Association Establishment

6.1.2.2.2 Authorization to access M2M resources

Once an AE or CSE has been granted access to M2M services, the Access Control procedure specified in Clause 7 of the present document shall be executed before accessing an M2M resource, as specified in oneM2M TS-0001 [1].

6.2 Security Service Layer

6.2.1 Access Management

6.2.1.1 Authentication

This component provides authentication services to the Application Layer. Annex B provides a general description of Authentication mechanisms.

6.2.2 Authorization Architecture

Figure 6.1.2-1 provides a high level overview of the authorization architecture. This architecture comprises four subcomponents that are described as follows:

- Policy Enforcement Point (PEP)
  - PEP intercepts resource access requests, makes access control decision requests, and enforces access control decisions. The PEP coexists with the entity that need authorization services.

- Policy Decision Point (PDP)
    - PDP interacts with the PRP and PIP to get applicable authorization polices and attributes needed for evaluating authorization policies respectively, and then evaluates access request using authorization policies for rendering an access control decision. The PDP is located in the Authorization service.
- Policy Retrieval Point (PRP)
    - PRP obtains applicable authorization policies according to an access control decision request. These applicable policies should be combined in order to get a finial access control decision. The PRP is located in the Authorization service.
- Policy Information Point (PIP)
    - PIP provides attributes that are needed for evaluating authorization policies, for example the IP address of the requester, creation time of the resource, current time or location information of the requester. The PIP is located in the Authorization service.

The Authorization service may comprise any of the subcomponents: PDP, PRP and/or PIP. This means that the subcomponents PEP, PRP, PDP and PIP could be distributed across different nodes. For example the PEP is located in an ASN/MN and the PDP is located in the IN.

NOTE: Release 1 does not support separation of PRP and PIP on different CSE from PDP.

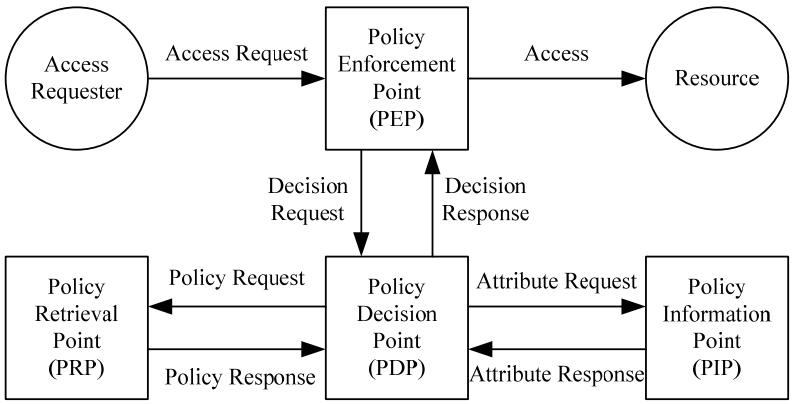


Figure 6.2.2-1: Overview of the authorization architecture

The authorization procedure is shown in figure 6.3.2-2.

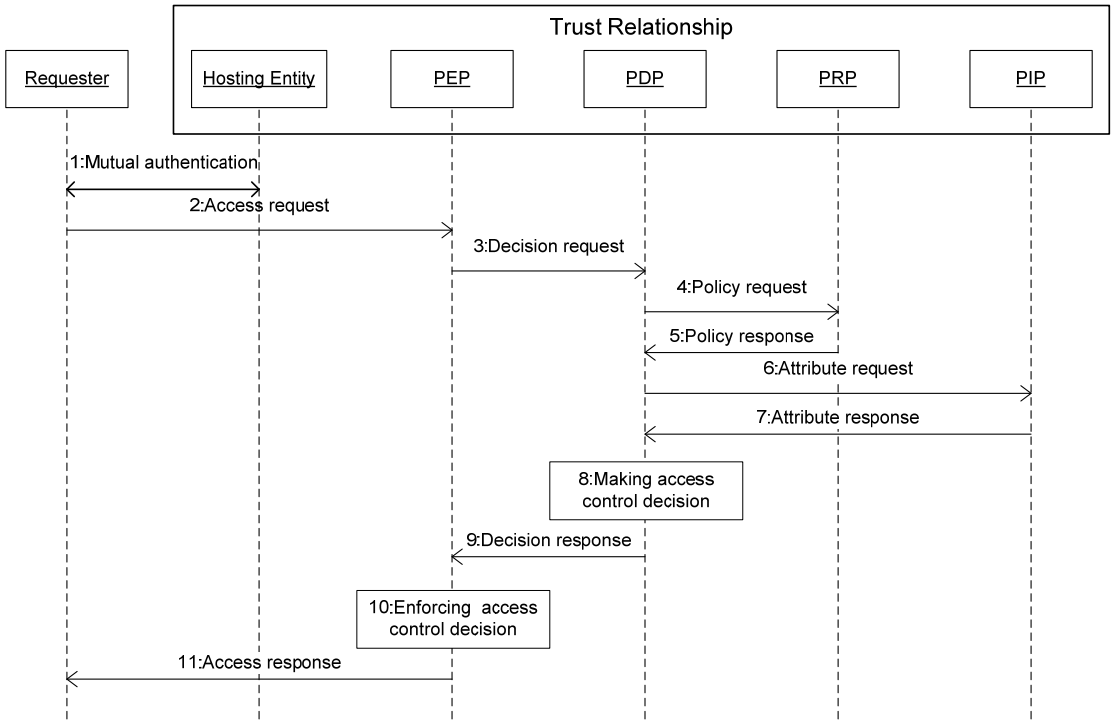


Figure 6.2.2-2: Authorization Procedure

Step 001: Mutual authentication (Pre-requisite).

Step 002: Access Requester sends an Access Request to the PEP.

Step 003: PEP makes an Access Control Decision Request according to the requester's Access Request, and sends the Access Control Decision Request to the PDP.

Step 004: PDP sends an Access Control Policy Request that is generated based on the Access Control Decision Request to the PRP.

Step 005: PRP finds all applicable access control policies to the access request and sends them back to the PDP. When multiple access control polices are involved, the PRP also provides a policy combination algorithm for combining multiple evaluation results into one finial result.

Step 006 PDP sends Attribute Request to the PIP if any attributes are required for evaluating these access control policies.

Step 007: PIP gets required attributes and sends them back to the PDP.

Step 008: PDP evaluates Access Request using access control policies. When there are multiple applicable access control policies, the PEP needs to calculate a final Access Control Decision using the policy combination algorithm.

Step 009: PDP returns the Access Control Decision back to the PEP.

Step 010: PEP enforces the access control decision, i.e. either forwards the Access Request to the resource or denies this access.

Step 011: PEP returns access result back to the Access Requester.

6.2.3 Security Administration

The Security Administration service shall provide functions to manage the Security functions, resources and attributes. This shall include management of resources provided via the secure environment. In addition it should provide functions to manage sensitive data with their associated identifiers and subscriptions on behalf of other entities. Security administration is therefore dependent upon the type of secure environment being used (independent hardware module, integrated trusted execution environment or software protection). Depending on the type of Secure Environment, distinct existing standards may be used for remote administration of those SEs.

6.2.3.1 Security Pre-Provisioning

Several sensitive data and associated objects are often configured by pre-provisioning the secure environment prior to deploying the M2M device it is associated with.

UICCs specified in ETSI TS 102 671 [23] are comonly used for such purpose because their use is required to access some underlying networks, they provide a high security level, and they offer an interoperable transport interface specified in ETSI TS 102 221 [24]. An interoperable oneM2M provisioning framework relying on this interface is specified in Annex D.

6.2.3.2 Remote security administration of SE

The Security Administration service may provide mechanisms for remote security administration of sensitive data and security functions of M2M field domain nodes, reusing mechanisms provided by existing standards where appropriate. However, since remote security administration requires the target information to be remotely modifiable, hardware based protection is required to protect such security information from remote software hacking of the device: This is the purpose of the Secure Environment. Remote security administration differs from standard device management by the requirement that the secure channel established with the administration server shall have its endpoint in the Secure Environment of the M2M Node.

The choice of a Secure Environments is guided by a risk analysis considering all layers of an M2M application, though it should leverage where possible on capabilities provided by the M2M Service Layer or the Underlying Network, e.g. UICC in 3GPP and 3GPP2 networks. Applicable remote security administration protocols are therefore dependent on the risk level of each M2M application and not just on the underlying network technologies. Widespread technologies that enable remote security administration for the different security levels distinguished in oneM2M TR-0008 [i.6] are considered in Annex C.

In case the Secure Environment relies on software protection only, remote security administration of the following data should be possible only when remote access by potential attackers can be controlled:

- Private key and associated identifiers
- Long-term shared symmetric key (compared to expected lifetime of the M2M node) and associated identifiers
- Any process and parameters thereof that manipulates to the above information, i.e. security functions.

6.2.4 Identity Protection

Identity Protection provides services to the Application Layer such as pseudonyms and protecting the anonymity of transactions.

6.2.5 Sensitive Data Handling

The Sensitive Data Handling service provides certain Sensitive Functions to the Application Layer.

Sensitive Functions shall include following functions:

- Secure Storage.

In addition the service should support the following sensitive functions

- Cryptographic operations.

- Methods for bootstrapping initial secrets (e.g. GBA).

6.2.5.1 Sensitive Functions

This service shall provide AEs and CSEs with access to Sensitive Functions of the SE.

6.2.5.2 Secure Storage

This service shall provide AEs and CSEs with access to the secure storage capability of the SE. Data securely stored by the AE or CSE shall only be accessible through the Security API and by authorized entities. Secure Storage should be managed by the Secure Environment. Stored data shall be associated with the entity owning the data, i.e. the entity that requested the data to be stored within the secure storage.

6.2.6 Trust Enabler security functions

OneM2M Trust Enabling Architecture requires the presence of two security functionalities within the Infrastructure Domain: the M2M Authentication Function (MAF) and the M2M Enrolment Function (MEF). They can be either under M2M Service Provider control or delegated to a M2M Trust Enabler.

- M2M Enrolment Function (MEF)
  - The MEF supports the security bootstrap procedure enabling the provisioning of the Master Credentials to be used to mutually authenticate entities accessing the infrastructure of an M2M Service Provider. The MEF relies on an initial credential pre-provisioned in the M2M node (e.g. during manufacturing). In case of MAF-based M2M Remote Security Provisioning procedure, the MEF provides the M2M Master Credential both to the MAF and the ASN/MN-CSE.
- M2M Authentication Function (MAF)
  - M2M Master Credentials, used to mutually authenticate CSEs/AEs before granting them access to M2M services, shall be securely stored in a specific infrastructure functionality named M2M Authentication Function (MAF).
  - The MAF securely contains the set of M2M Master Credentials that are used for authenticating CSEs/AEs that have been enrolled for M2M services. The MAF stores the M2M Master Credentials and possibly the identifiers of the associated CSE/AE. The MAF is identified by its MAF-ID.
  - When M2M Remote Security Provisioning procedure takes place to share a M2M Master Credential between an ASN/MN CSE and the M2M Authentication Function, the M2M Enrolment Function (MEF) communicates with the MAF through an appropriate secure interface, if not co-located.
  - The MAF is also in charge of all security operations involving the usage of the M2M Master Credentials.

6.3 Secure Environment AbstractionLayer Components

6.3.1 Secure Environment

The Secure Environment component is a logical entity that provides Sensitive Functions operating on Sensitive Data, Secure Storage and other resources/functions.

There is no assumption made on the particular implementation of the Secure Environment. A SE may be implemented as an independent HW Security Element or as an integrated SW function. Each Secure Environment shall be associated with one certain Security Level depending on the particular implementation of the SE. Different Secure Environments may provide different Security Levels and protection levels as indicated in table 6.2.1-1.

Protection Level	Description
0	No protection. The data are exposed even without active attacks.
1	Low protection, data are protected from passive observers but could be exposed by active attacks, be they local or remote. E.g. software solutions exist that rely on general purpose processing hardware of the supporting equipment.
2	Medium protection, protection of the data from remote attacks is addressed, but local attacks, especially physical attacks, remain possible, ie. Medium protection provides countermeasures against software attacks only E.g. Software solutions to protect data and sensitive functions rely on specific processing providing enforced isolation and enables sensitive code and data to be kept away from an unprotected operating environment, software and memory. The code running in the protected environment is cryptographically verified for integrity assurance.
3	High protection, addressing both remote and local attacks to access the data, including attacks involving physical access. This includes strong counter measures against software and hardware attacks, such as detection of abnormal operating conditions and scrambling plus hardware masking of the memory and side channel analysis of operations involving sensitive data.

Table 6.3.1-1: Classification of Protection levels

There shall be at least one Secure Environment, however there may be multiple.

6.3.2 SE Plug-in

The SE Plug-in enables physical access to the respective Secure Environment. Depending on the type of Secure Environment, the SE Plug-in may be implemented differently for each Secure Environment.

NOTE: Specification of the SE Plug-in is out of scope of the present document.

6.3.3 Secure Environment Abstraction

This component is not specified in the present document.

7 Authorization

7.1 Access Control Mechanism

7.1.1 General Description

The M2M authorization procedure controls access to resources and services hosted by CSEs and AEs. The authorization procedure requires that the originator of the resource access request message has been identified to the Authentication Function, and originator and receiver are mutually authenticated with each other.

The resource addressed in a request message has an associated accessControlPolicyID attribute (either included explicitly as an attribute of the resource addressed in the request message, implied from the parent of the resource, or set fixed by the system, see clause 9.6.1 of TS-0001 [1]). The accessControlPolicyID attribute contains a list of identifiers of <accessControlPolicy> resources applicable to the resource addressed in the request message.

The overall structure of <accessControlPolicy> resources is described in clause 9.6.2 "Resource Type *accessControlPolicy"* of TS-0001 [1]).

Each of these <accessControlPolicy> resources include *privileges* and *selfPrivileges* attributes, which comprise the information, denoted as *access control rules* in this specification, that is evaluated against the parameters associated with the request message to obtain the access decision.

Figure 7.1.1-1 illustrates the relation between <accessControlPolicy> resource instances (ACP) and the instances of the protected resources, denoted Resource_1 to Resource_N.

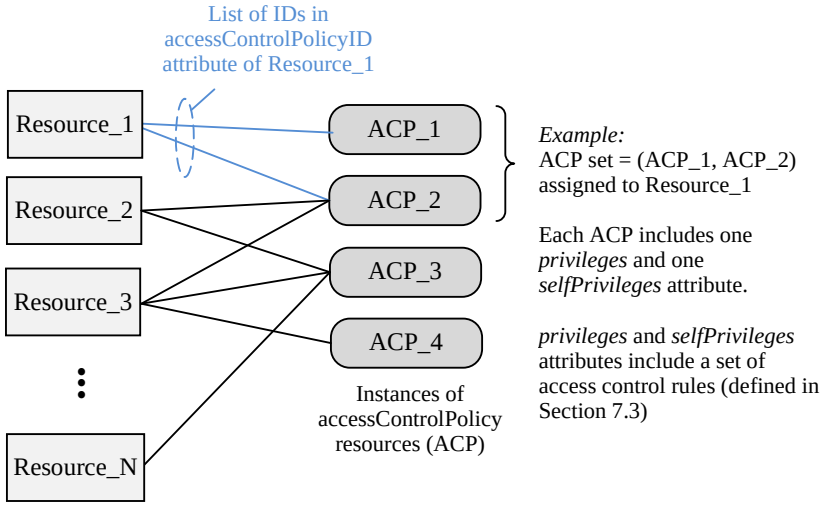


Figure 7.1.1-1: Relation between Resource Instances and Access Control Policies

Access requests to ACP's itself are evaluated against the *selfPrivileges* attribute of that ACP. Access requests to instances of all other resource types, are evaluated against the *privileges* attributes of the ACP set associated with the targeted resource.

For requests to <accessControlPolicy> resource type, authorization is granted if the request is evaluated to "Permit" for at least one *selfPrivileges* attribute. For other resource types, authorization is granted if the request is evaluated to "Permit" for at least one *privileges* attribute.

The *privileges* and *selfPrivileges* defined in the *accessControlPolicy* resource determine *which request originator* is allowed to access the resource containing this attribute, for *which specific operation* (i.e. Create, Retrieve, Update, Delete, etc.) and *for which specific context constraints* (i.e. constraints regarding access time, originator's IP address and originator's location).

The access control approach specified here conforms to the concept of Attribute Based Access Control (ABAC) as defined in [i.14].

The policies defined in the <accessControlPolicy> resources are enforced by an access control mechanism which employs the authorisation logical architecture outlined in Section 6.2.2.

The access control mechanism assembles the information needed to render the access decision which consists of

- information included in the resource access request message as defined in section 7.1.2 below (Table 7.1.2-1),
- contextual information as defined in section 7.1.2 below (Table 7.1.2-2),
- the policies governing the access as defined in section 7.1.3 below.

7.1.2. Parameters of the Request message

This section specifies the parameters of a request message which are evaluated by the access control mechanism.

The data types applicable to these parameters are defined in clause 6.4 of TS-0004 [4].

The parameters are listed in Table 7.1.2-1.

Table 7.1.2-1: Parameters indicated in the request message

Parameter	Description	Mandatory / Optional	Usage in access control mechanism
to	URI of target resource	M	Selection of accessControlPolicy associated with the target resource
fr	Identifier representing the originator of the request	M	Evaluated against accessControlOriginators in *privileges* and *selfPrivileges* attributes
role	Role of the originator	O	Evaluated against accessControlOriginators in *privileges* and *selfPrivileges* attributes NOTE: this parameter is for use in future Release(s).
op	Requested operation	M	Evaluated against accessControlOperations in *privileges* and *selfPrivileges* attributes
fc	Filter criteria	O	Differentiation between Retrieve and Discovery operations

Table 7.1.2-2 lists the context parameters associated with a request message which are evaluated by the access control mechanism. These parameters are not explicitly included in a request message but can be obtained at the receiver and validated against the context policy parameters as given in Table 7.1.2-2.

Table 7.1.2-2: Context parameters associated with a request message

Parameter	Description	Mandatory / Optional	Usage in access control mechanism
rq_time	Time stamp when the request message was received at the hosting CSE. Obtained by the hosting CSE's system time clock.	O	Validated against accessControlTimeWindows parameter in an access control rule, cf. section 7.3
rq_loc	Location information about the originator of the request. Obtained over the Mcn reference point.	O	Validated against accessControlLocationRegions parameter in an access control rule, cf. section 7.3
rq_ip	IP source address associated with the IP packets that carry the request message. Obtained over the Mcn reference point.	O	Validated against accessControlIpAddresses parameter in an access control rule, cf. section 7.3

7.1.3 Format of *privileges* and *selfprivileges* Attributes

The *privileges* and *selfPrivileges* attributes exhibit the same data type format which is specified as follows.

Each *privileges* or *selfPrivileges* attribute comprises a set (aka. list) of access control rules. We denote in the following the set of access control rules as *acrs* and an individual access control rule in this set as *acr*. The access control rules in *acrs* are indexed with the letter *k*. The number of access control rules in the set is denoted with the letter K:

$$acrs = \{ acr(1), acr(2), ..., acr(k), ..., acr(K) \}$$

Each access control rule *acr*(*k*) is comprised of three type of components, denoted accessControlOriginators, accessControlOperations and accessControlContexts. The accessControlContext component is an optional parameter.

Hence, an access control rule *acr*(*k*) is either represented as a pair,

$$acr(k) = \{acr(k)\_accessControlOriginators, acr(k)\_accessControlOperations\}$$

or as a 3-tuple

$$acr(k) = \{acr(k)\_accessControlOriginators, acr(k)\_accessControlOperations, acr(k)\_accessControlContexts\}$$

We use the generic term "access-control-rule-tuple" when referring to a rule *acr*(*k*).

A set *acrs* of access control rules may consist of a mix of pairs and 3-tuples. For pairs, any context parameters associated with a request message are admissible.

The three component parameters of an access-control-rule-tuple supported in this specification are shown in Table 7.3.1-1.

Table 7.1.3-1: Parameters of an access-control-rule-tuple

Parameter	Usage Description	Mandatory/ Optional	Format
accessControlOriginators	Set of Originators that can be authorized	M	List of CSE-IDs and/or AE-IDs, or keyword "all" to grant access to all originators
accessControlOperations	Set of Operations that can be authorized	M	Enumerated list of operations Create Retrieve, Update, Delete, Discover, Notify
accessControlContexts	See Table 7.3.1-2	O	See Table 7.3.1-2

The accessControlOriginators parameter comprises a list of CSE-IDs and/or AE-IDs of any format defined in TS-0001 [1]. It is allowed to include the wildcard characters, e.g. "*", into the URI string of CSE-ID and AE-ID at any level. Examples include the following: *.mym2msp.org/mycseID, /mycseID/*, mym2msp.org/mycseID, /mycseID/myAE*. If access for all originators should be allowed, the reserved keyword 'all' can be included into the value space of accessControlOriginators. Granting access to all CSE originators of the same M2M SP domain could be represented as /*, all AE-IDs of all CSEs in the same domain as /*/*.

The data type applicable to accessControlOriginators is defined in TS-0004 [4].

The accessControlOperations parameter comprises a list of admissible operations which can be any subset of the following elements: Create, Request, Update, Delete, Discover, Notify. While Create, Request, Update, Delete, and Notify operation are explicitly indicated in the *op* parameter of a request message, the Discovery operation is indicated by ***op*** = retrieve in combination with the provisioning of ***fc*** and *Disrestype* parameters in the request message.

The data type applicable to accessControlOperations is defined in TS-0004 [4].

The accessControlContexts parameters are listed in Table 7.1.3-2.

Table 7.1.3-2: Parameters of accessControlContexts

Parameter	Usage Description	Mandatory/ Optional	Formats
accessControlTimeWindows	Set of Time Windows that can be authorized	O	List of time intervals where access can be granted in extended crontab format
accessControlLocationRegions	Set of Location Regions that can be authorized	O	1) Latitude/longitude coordinates, and a radius defining a circular region around the coordinates 2) Country code
accessControlIpAddresses	Set of IPv4 and IPv6 addresses that can be authorized	O	IPv4: dotted-decimal notation with CIDR suffix IPv6: colon separated groups of hexadecimal digits with CIDR suffix

The accessControlTimeWindows parameter represents a list of elements that comply to the extended crontab syntax as defined in clause 7.3.8 of TS-0004 [4]. It allows definition of periodically recurring time intervals at which access shall be granted, when the ***rq_time*** parameter associated with the access request message falls into such interval.

For the elements of accessControlLocationRegions there are two representation choices. These can be represented by a 2-character country code or a circle with radius *R* centred at a point defined in terms of longitude and latitude parameters. Refer to Annex E for detailed information. Each element of accessControlLocationRegions defines an admissible location region, which is compared with the ***rq_loc*** parameter associated with the access request message.

The data types applicable to accessControlLocationRegions and ***rq_loc*** are defined in TS-0004 [4].

The accessControlIpAddresses parameter represents a list of IPv4 and IPv6 addresses in dotted-decimal notation with CIDR suffix or colon separated groups of hexadecimal digits with CIDR suffix, respectively. If the ***rq_loc*** parameter associated with the access request message matches one of these addresses, access max be granted with regard to this criterion.
The data types applicable toaccessControlIpAddresses and ***rq_ip*** are defined in TS-0004 [4].

7.1.4 Access Control Decision

The access decision is derived by comparing the parameters associated with a resource access request message as described in clause 7.1.2 with the access control rules included in the *privileges* or *selfPrivileges* attributes of all ACP sets assigned to the protected resource by means of the accessControlPolicyID, cf. Figure 7.1.1-1.

The result of the access control algorithm, i.e. the access decision, is the overall result of evaluating the applicable set of access control rules, *acrs*, against the parameters associated with the access request message. This access decision can be represented by a value of binary data type. The overall result of the access decision algorithm is denoted here with the variable name *res_acrs*:

$$res\_acrs = \begin{cases} \text{TRUE or 1} & \text{if the request matches the access control rules} \\ \text{FALSE or 0} & \text{else} \end{cases}$$

The access control algorithm is specified in clause 7.1.5.

If the access control algorithm yields the result res_acrs = TRUE, the access decision for the requested resource is "Permit".

If the result is *res_acrs* = FALSE, or the access control algorithm is not capable to derive a final result (e.g. due to indeterminate parameters), the access decision for the requested resource is "Deny".

Access decisions that result in indeterminate access control rules should be logged for diagnostics purposes.

7.1.5 Description of the Access Decision Algorithm

The access control algorithm specified in this section combines partial access control results obtained for each of the individual access control rules contained in a *privileges* or *selfPrivileges* attribute. Further, if multiple ACP instances

are assigned to the protected resource, the access control algorithm combines the partial access control results obtained for the individual ACPs of an ACP set.

The algorithm specified in this section adopts a "Permit-overrides" combining algorithm with respect to access control rules and ACPs as defined in XACML [i.7]. This algorithm has the following behaviour:

(1)If a decision is "Permit" for only a single access control rule included in the *privileges* (or *selfPrivileges*) attribute of a single ACP, the result is "Permit",

(2)otherwise, the result is "Deny".

The logic for evaluating a request against a privilege can be described mathematically as follows. A *privileges* or *selfPrivileges* attribute included in an <accessControlPolicy> resource represents a set of access control rules, *acrs*, which is built as in the graph below:

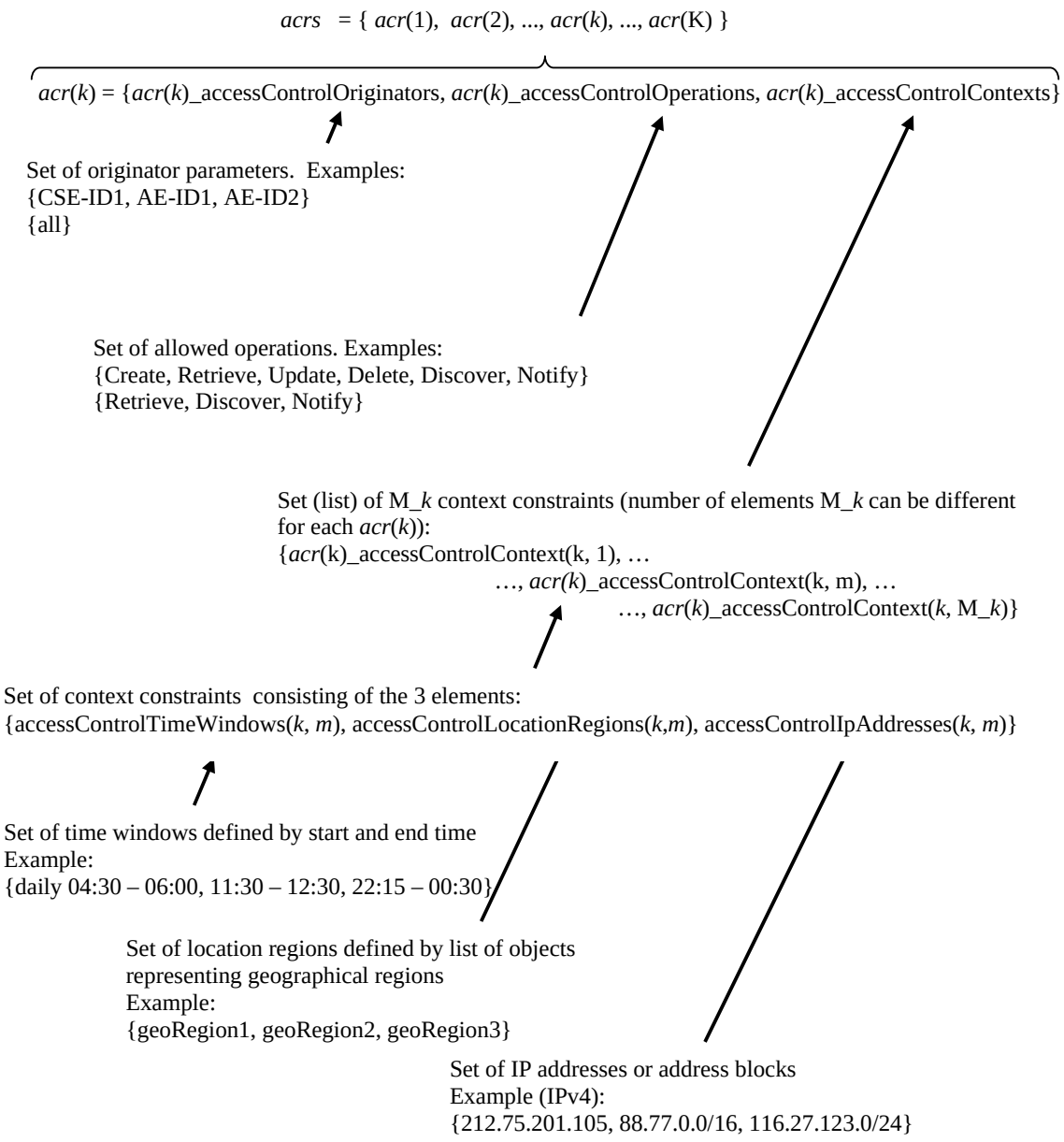


Figure 7.1.5-1 Logic to evaluate privileges

The parameters associated with a request, which are evaluated against the parameters contained in the access control rules are specified in clause 7.1.3 above:

The access decision *res_acrs* defined in section 7.1.4 is derived by evaluating whether or not the parameters associated with the request message listed in Tables 7.1.2-1 and 7.1.2-2 match any of the access control rules contained in the access control rule set defined in section 7.1.3 as follows:

$$res\_acrs = res\_acr(1)\ \text{OR}\ res\_acr(2)\ \ldots\ \text{OR}\ res\_acr(\text{k})\ \ldots\ \text{OR}\ res\_acr(\text{K}),$$

where *res_acr*(k) represents the logical evaluation result (i.e. TRUE/FALSE or 1/0) of the request parameters against the k^{th} access control rule in the set *acrs*, which can be expressed as follows:

$$res\_acr(k) = res\_origs(k)\ \text{AND}\ res\_ops(k)\ \text{AND}\ res\_ctxts(k),\ k = 1\ldots\text{K}.$$

The 3 partial logical result variables on the right side of above equation can be defined by using the following set function:

$$\text{ismember}(x, \text{setX}) = \begin{cases} \text{TRUE or 1} & \text{if } x \in \text{setX} \\ \text{FALSE or 0} & \text{else} \end{cases}$$

With this definition:

$$res\_origs(k) = \text{ismember}(rq\_orig,\ acr(k)\_\text{accessControlOriginators})$$

$$res\_ops(k) = \text{ismember}(rq\_op,\ \text{acr}(k)\_\ \text{accessControlOperations})$$

The third partial logical result *res_ctxts*(*k*) is derived as follows

$$res\_ctxts(\text{k}) = res\_context(k, 1)\ \ldots\ \text{OR}\ res\_context(k, m)\ \ldots\ \text{OR}\ res\_context\ (k, \text{M}\_k),$$

where,

$$res\_context(k, m) = res\_time(k, m)\ \text{AND}\ res\_ip(k, m)\ \text{AND}\ res\_loc\ (k, m),\ \text{k} = 1\ldots\text{K},\ m = 1\ldots\text{M}\_k$$

and

$$res\_time(k, m) = \text{ismember}(rq\_time, acr(k)\_\text{accessControlTimeWindows}(m))$$

$$res\_ip(k, m) = \text{ismember}(rq\_ip, acr(k)\_\text{accessControlIpAddresses}(m))$$

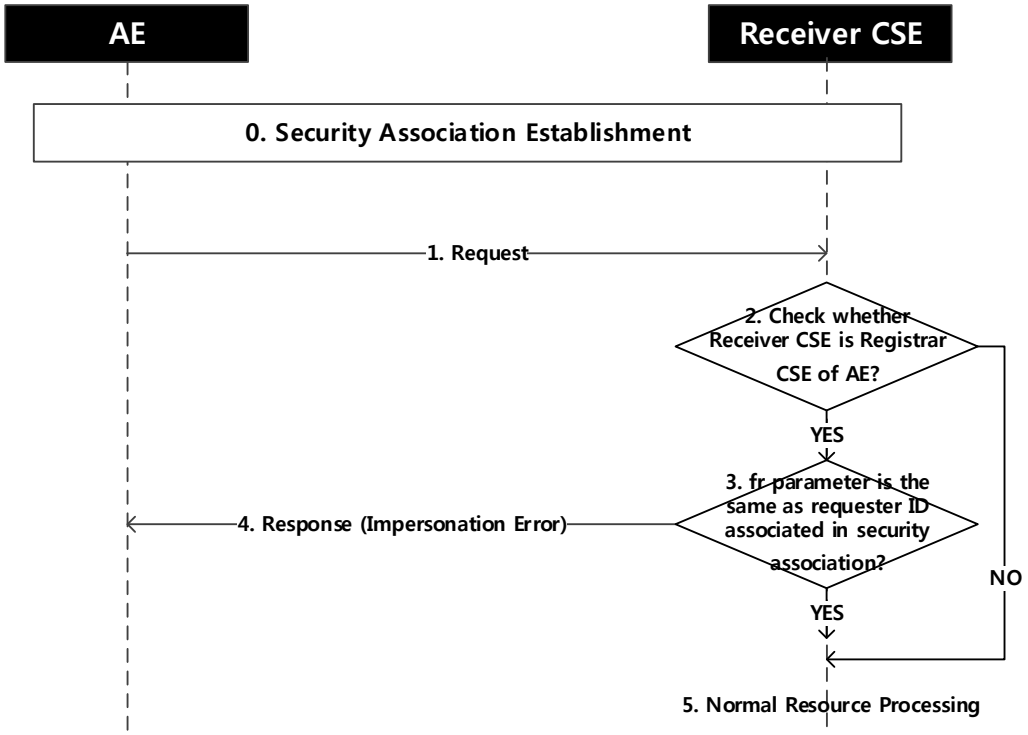
$$res\_loc\ (k, m) = \text{ismember}(rq\_loc, acr(k)\_\text{accessControlLocationRegions}(m))$$

Thanks to the "Permit-overrides" combining approach, if the access control decision for one access control rule results in *res_acr* = TRUE, the access control algorithm can stop without evaluating any other applicable access control rules of the current ACP or any other ACPs in the ACP set, and the final access decision is "Permit".

7.2 AE Impersonation Prevention

Since several AEs can behave maliciously and pretend to be another AE with their ID changed, Receiver CSE needs prevention mechanism for AE impersonation. This mechanism works at Registrar CSE since Registrar CSE is an entry point of M2M system.

When Receiver CSE receives a request, Receiver CSE shall perform as follows.



0. Security association establishment is performed

1. AE sends a request to Registrar CSE.

2. Receiver CSE checks whether Receiver CSE is Registrar CSE (Registrar CSE finds ID associated in security association and find any <AE> resource which contains that ID in *AE-ID* attribute).

3. If Receiver CSE is Registrar CSE, Receiver CSE checks the value in *fr* parameter is the same as the found ID in step 2. If the request is AE registration and fr parameter is empty string, this procedure is terminated.

4. If the value is not the same, Receiver CSE sends response with impersonation error response code.

5. Receiver CSE performs procedures specified in clause 8.2 [1].

8 Security Frameworks

8.1 General Introductions to the Security Frameworks

To accommodate the variety of deployment scenarios that can be encountered in M2M applications, the present specification supports a diversity of methods to provision and establish security in M2M systems.

The security frameworks can be categorized into two main types

- *Direct Security Frameworks:* where the entities authenticate each other directly, without assistance from a Central Key Distribution Server. To authenticate each other, those entities shall be able to support either a symmetric-based security framework where the two entities share a symmetric key to perform a mutual authentication, or a certificate-based security framework to perform certificate-based mutual authentication.

- *Centralized Security Frameworks:* where the entities authenticate each other with the assistance of a Central Key Distribution Server. The M2M applications have to trust the operator of the Central Key Distribution Server (which may be an M2M Service Provider, an Underlying Network Service Provider or an M2M Trust Enabler. The Central Key Distribution enables the establishment of symmetric key that is shared between the two entities. Then, this symmetric key can be used by the two entities to perform mutual authentication.

8.1.1 General Introduction to the Direct Security Frameworks

8.1.1.1 General Introduction to the Symmetric Key Direct Security Framework

In the Direct Security Framework using Symmetric Key, each pair of entities that need to authenticate each other has to be provisioned with its own shared symmetric key. This may be performed through pre-provisioning, e.g. during device manufacturing or deployment, or in the case of security association, a remote security provisioning framework may be used to perform the provisioning.

8.1.1.2 General Introduction to the Certificate-Based Direct Security Frameworks

This clause describes the Credential Configuration and Certificate Verification used in the Certificate-Based Security Association Establishment Framework and Certificate-Based Remote Security Provisioning Framework.

8.1.1.2.1 Public Key Certificate Flavours

This document defines procedures using the following Public Key Certificate flavours:

- Raw Public Key Certificates:
    - **Description:** A raw public key certificate (RFC7250 [37]) contains only the raw public key, without other information normally provided in a certificate. The raw public key certificate is exchanged in the TLS handshake in the place of a traditional certificate (see RFC7250 [37]).
    - **Use:** A raw public key certificate may be used for authenticating a CSE or AE either during the Association Security Handshake phase of the Certificate-Based Security Association Establishment or during the Bootstrap phase of the Certificate-Based Remote Security Provisioning Framework.
- Device certificates:
    - **Description:** These certificates have a certificate chain to a trust anchor and include one or more globally unique hardware instance identifier (such as the Object Identifier Based M2M Device identifiers discussed in Annex H “Object Identifier Based M2M Device Identifier” TS-0001 [1]) in the subjectAltName extension of the certificate. A device certificate can be used to verify the identity of the hardware instance on which the entity is being executed.
    - **Use:** Device certificates may be used to authenticate a CSE or AE executing on a specific M2M Device. If the M2M device is an ASN or MN (which supports a CSE), then the device certificate is implicitly associated with the CSE that executes on the device. If the device is an ADN (which does not support a CSE) then the device certificate is not implicitly associated with a specific AE executing on the hardware. A device certificate may be used for authenticating a Field Domain CSE either during the Association Security Handshake phase in the Certificate-Based Security Association Establishment Framework or during the Bootstrap phase of the Certificate-Based Remote Security Provisioning Framework.
- CSE-ID certificates:
    - **Description:** These certificates have a certificate chain to a trust anchor and include the full URI representation of a CSE-ID in the subjectAltName extension of the certificate. A CSE-ID certificate verifies that the entity presenting the certificate has been assigned a particular CSE-ID.
    - **Use:** A CSE-ID certificate may be used to authenticate a CSE only.
- AE-ID certificates:
    - **Description:** These certificates have a certificate chain to a trust anchor and include the full URI representation of an AE-ID in the subjectAltName extension of the certificate. An AE-ID certificate verifies that the entity presenting the certificate has been assigned a particular AE-ID.
    - **Use:** An AE-ID certificate may be used to authenticate an AE only.
- FQDN certificates:

- **Description:** These certificates have a certificate chain to a trust anchor and include the FQDN of an M2M Enrolment Function or M2M Authentication Function in the subjectAltName extension of the certificate. An FQDN certificate verifies that the entity presenting the certificate has been assigned a particular FQDN.
- **Use:** A FQDN certificate shall be used to authenticate an M2M Enrolment Function to an Enrolee during a Bootstrap phase in a Certificate-Based Remote Security Provisioning Framework. FQDN certificates shall be used to for mutual authentication of M2M Enrolment Function (or GBA BSF) with M2M Authentication Functions in the Enrolment Phase of all Remote Security Provisioning Frameworks.

NOTE: The flavours, and the details specific for these flavours, are specified to support a range of deployment models while ensuring that oneM2M entities have clear procedures for authenticating other oneM2M entities using certificates.

The profiles for these certificates are found in clause 10.1.1 “Certificate Profiles”.

8.1.1.2.2 Path Validation and Certificate Status Verification

If an entity is to authenticate another entity using a device certificate, CSE-ID certificate, AE-ID certificate or FQDN certificate, then the entity shall perform basic path validation (Section 6.1of RFC 5280 [34]) as part of verifying the other entity’s certificate (see clause 8.1.1.2.4 “Certificate Verification”).

Certificate authority certificates shall include the name constraint extensions (clause 4.2.1.10 “Name Constraints” of RFC 5280 [34]) and shall constrain the names (object identifier M2M Device IDs from Annex H “Object Identifier Based M2M Device Identifier” TS-0001 [1], CSE-ID URIs, AE-ID URIs or FQDNs respectively) which may be in the subsequent certificate used to authenticate the entity (device certificate, CSE-ID certificate, AE-ID certificate or FQDN certificate respectively).

- Clause 4.2.1.10 “Name Constraints” in RFC5280 [34] describes how the name constraint extension is used for constraining URIs and FQDNs.
- Clause 10.4.1.4.2 “Profile for Certificate Authority Certificates for Device Certificates” describes how the name constraint extension is used for constraining object identifier M2M Device IDs.

The trust anchor certificate containing the trust anchor information (Section 6.1.1 of RFC5280 [34]) is provided to the entity during Credential Configuration, Association Configuration, Bootstrap Credential Configuration or Bootstrap Instruction Configuration.

NOTE: Section 6.1.1 of RFC 5280 [34] states “The trust anchor information is trusted because it was delivered to the path processing procedure by some trustworthy out-of-band procedure”. Credential Configuration, Association Configuration, Bootstrap Credential Configuration and Bootstrap Instruction Configuration satisfy the requirements of being trustworthy out-of-band procedures.

Certificate status verification: In the case of an Infrastructure Domain entity receiving an MEF certificate, the entity shall verify the status of the certificate using a Certificate Revocation List as described in RFC 5280 [34]. oneM2M support for certificate status checking in Field Domain entities requires further study. A mapping of the Online Certificate Status Protocol (OCSP) onto HTTP may be used, as described in Appendix A of RFC 6960 [35], however a mapping of OCSP onto CoAP is not currently defined. Furthermore, OCSP may also not be easily applicable in all environments. An alternative approach may be using the TLS Certificate Status Request extension (Section 8 of [44]; also known as "OCSP stapling") or preferably the Multiple Certificate Status Extension ([36]), if available.

NOTE: Most of the above paragraph is based on almost identical text in the CoAP specification RFC 7252 [38], a protocol with similar (if not identical) considerations to oneM2M deployments.

8.1.1.2.3 Credential Configuration for Certificate-Based Security Frameworks

If an entity is to authenticate itself using a Certificate-Based Security Framework, then the entity shall be pre-provisioned with the following information

- The entity's Private Signing Key.

NOTE 1: An entity authenticates itself to other entities by proving that it knows the Private Signing Key corresponding to a particular Public Verification Key.

- The entity's Certificate (and if applicable, Certificate Chain) as described in clause 10.1.1 “Certificate Profiles”.
- In the case of a CSE-ID certificate the entity shall be configured with the entity’s CSE-ID.
- In the case of an AE-ID certificate the entity shall be configured with the entity’s AE-ID.

8.1.1.2.4 Information Needed for Certificate Authentication of another Entity

An entity must trust the following information in order to authenticate another entity using certificates:

- An indication of the public key certificate flavour of other entity’s Certificate (that is, raw public key certificate, device certificate, CSE-ID certificate, AE-ID certificate, or an MEF certificate).
- In the case where other entity’s certificate is a raw public key certificate:
    - A public key identifier for the raw public key in the certificate (see clause 10.1.2 “Public Key Identifiers”). The public key identifier can be available
- In the case where other entity’s certificate is an device certificate, CSE-ID certificate, AE-ID certificate or FQDN certificate:
    - **A Globally unique identifier:** The globally unique identifier for the entity which is also present in the subjectAltName extension of the other entity’s certificate
        - Device Certificate: A globally unique hardware instance identifier (such as the object identifier M2M Device ID in Annex H “Object Identifier Based M2M Device Identifier” TS-0001 [1]) that is present in the device certificate.
        - CSE-ID Certificate: The full URI representation of the CSE-ID
        - AE-ID Certificate: The full URI representation of the AE-ID
        - FQDN Certificate: The FQDN of the MEF or MAF
    - **Trust Anchor Certificates:** One or more trust anchor certificates for the other entity’s certificate chain (see clause 8.1.1.2.2 “Path Validation and Certificate Status Verification”)

8.1.1.2.5 Certificate Verification

This clause describes how an entity authenticates the other entity in the Security Handshake of a Certificate-Based Security Framework.

The other entity's Certificate is received during the Security Handshake.

The other entity's Certificate is verified as follows:

- If the certificate information configured during the Association Configuration or Bootstrap Instruction Configuration indicates that the other entity's Certificate is a raw public key certificate, then the entity verifies that the public key identifier (received during Association Configuration or Bootstrap Instruction Configuration) corresponds matches the raw public key certificate (received during the Security Handshake) using the process described in clause 10.1.2 “Public Key Identifiers”.
- If the certificate information configured during the Association Configuration or Bootstrap Instruction Configuration indicates that the other entity's Certificate is a device certificate, CSE-ID certificate, AE-ID certificate or FQDN certificate, then the entity shall perform the following verifications:
    - The entity shall look for a match between the globally unique identifier described in clause 8.1.1.2.4 “Information Needed for Certificate Authentication of another Entity” (received during Association Configuration or Bootstrap Instruction Configuration) and the values in the subjectAltName extension of the other entity's Certificate (received during the Security Handshake). If there is not an exact match, then the entity shall abort the (D)TLS handshake.
        - In the case of device certificate, the globally unique identifier is a globally unique hardware instance identifier (such as the object identifier M2M Device ID in Annex H “Object Identifier Based M2M

Device Identifier" TS-0001 [1]). In this case, the notion of a "match" depends on how the globally unique hardware instance identifier can be represented in the subjectAltName extension.

- In the case of a CSE-ID certificate, the globally unique identifier is the CSE-ID, and a match is a URI that is an exact match for the CSE-ID.
- In the case of an AE-ID certificate, the globally unique identifier is the AE-ID, and a match is a URI that is an exact match for the AE-ID.
- In the case of an FQDN certificate, the globally unique identifier is the FDQN of the M2M Authentication Function or M2M Enrolment Function, and a match is a URI, FQDN or dNSName that is an exact match for the FDQN of the M2M Authentication Function or M2M Enrolment Function.

- The entity shall perform path validation and certificate status verification using the trust anchor certificate as described in clause 8.1.1.2.2 "Path Validation and Certificate Status Verification"). If this verification fails, then the entity shall abort the (D)TLS handshake.

NOTE: After a successful Security Handshake in which the other entity provides a Certificate Chain, the other entity's identity (received during Association Configuration or Bootstrap Instruction Configuration) can be associated with additional information extracted from the other entity's Certificate Chain (e.g. the other entity Manufacturer, other entity owner, or conformance criteria). These details are not described in the present document.

8.1.2 General Introduction to the Centralized Security Frameworks

A Centralized Security Framework relies on a Central Key Distribution Server in charge of establishing a symmetric key that will be shared between two entities. This framework could be used either in the scope of Security Association Establishment or in the scope of Remote security provisioning. The type of the server playing the role of the Central Key Distribution Server depends on the scope. In case of Remote security provisioning, the Centralized Key Distribution Server is the MEF. In case of Security Association Establishment, the Central Key Distribution Server can be the M2M Authentication Function (MAF) or the Bootstrapping Server Function (BSF). The BSF is an element of GBA framework.

Generic Bootstrapping Architecture (GBA), described below, is a framework that could be used as Centralized Security Frameworks either for Security Association Establishment or Remote security provisioning.

8.1.2.1 General Introduction to the GBA (Generic Bootstrapping Architecture) Framework

In case of scenario where the M2M Service Provider and the operator of the underlying network have an agreement to use the underlying network credentials as the basis for security between a M2M Application Service/Middle Node and Infrastructure Node (including the case that the M2M Service Provider and the operator of an underlying network are actually the same entity), GBA procedure could be used.

It is important that this feature is used only within the scope of an appropriate agreement between the M2M Service Provider and the operator of the underlying network. The normative text for the GBA-Based Security Association Establishment Framework (clause 8.2.2.2) and the GBA-Based Security Bootstrap Framework (clause 8.3.2.2) implicitly assumes that such an agreement is already in place. Since the present document is a technical specification, it does not address the details of such an agreement.

A general introduction to GBA is included in TR-0008 [i.6].

After a successful GBA bootstrapping, the M2M Application Service/Middle Node and the BSF share a security association which consists of a bootstrapping transaction identifier (B-TID) and key material (GBA bootstrap Ks).

This security association may be used by the M2M Application Service/Middle Node to derive NAF keys (Ks_(ext/int)_NAF) shared between a M2M Application Service/Middle Node and a M2M Infrastructure Node or an M2M Authentication Function.

There are two modes of GBA: ME-based GBA (GBA_ME) and UICC-based GBA (GBA_U). In case of GBA_ME, one NAF-specific key is derived: the key Ks_NAF. In case of GBA_U, two NAF-specific keys are derived: Ks_ext_NAF (available in the ME) and Ks_int_NAF (which remains inside the UICC).

GBA_U requires that the UICC is GBA aware.

The BSF determines which mode to run based on the UICC capability indicated in the GBA User Security Settings (GUSS).

The usage of GBA_U is recommended since it provides a higher level of security than GBA_ME. The implication of this recommendation is that the entity, AE or CSE, using the GBA_U-based NAF keys should be resident in the UICC.

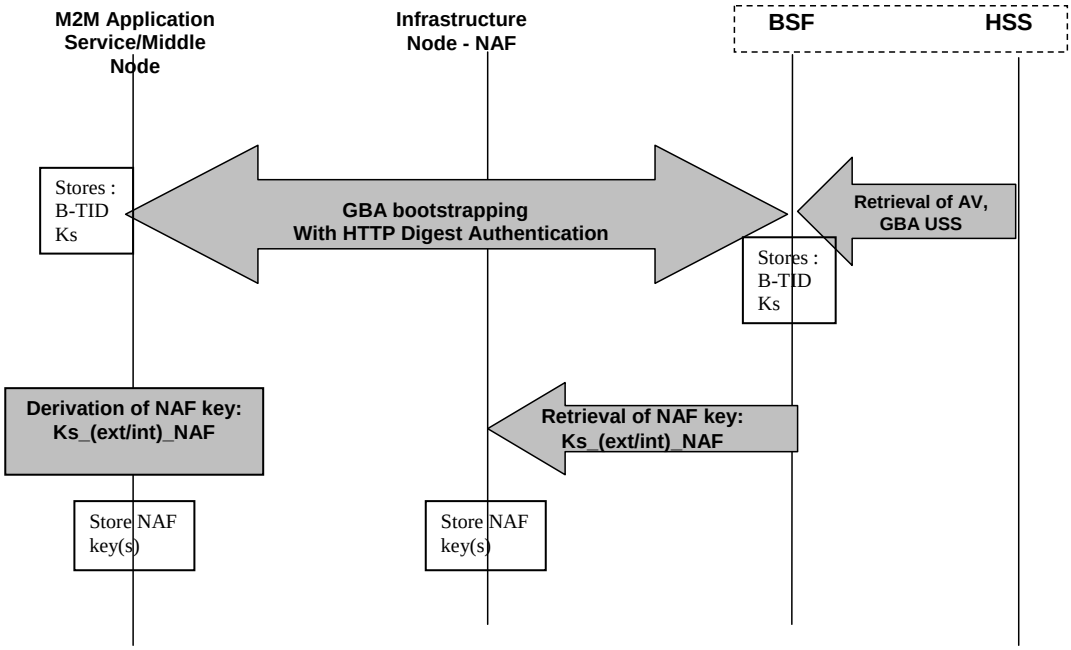


Figure 8.1.2.2-1: GBA framework. Note that the Network application Function (NAF) may be an Infrastructure Node or an M2M Authentication Function

8.2 Security Association Establishment Frameworks

8.2.1 Overview on Security Association Establishment Frameworks

In the present document, security associations are restricted to single hop on Mcc, Mcc' or Mca reference point.

The oneM2M system supports the following Security Association Establishment Frameworks:

- **Direct Security Association Establishment Frameworks:**
  - **Provisioned M2M Secure Connection Key** Security Association Establishment. A symmetric key is pre-provisioned to the entities: this is called the Provisioned M2M Secure Connection Key, and denoted Kpsa. The entities authenticate each other by verifying Message Integrity Codes (MIC) in the Security Handshake which were generated using the symmetric key. For more details see clause 8.2.2.1.
  - **Certificate-Based** Security Association Establishment: The entities are each issued with
    - a Private Signing Key that is known only to that entity,
    - a Certificate containing the corresponding Public Verification Key, and

- (Optionally) a Certificate Chain from the entity's Certificate to a Root Certificate.

  The entities must validate each other's Certificate before trusting the Public Verification Keys in the Certificate. Within the Security Handshake, entity A creates a digital signature of the session parameters using its private signing key and entity B verifies the digital signature using entity A's public verification key. Then the roles are reversed: entity B creates a digital signature and entity A verifies it. For more details see clause 8.2.2.2.

- **Centralized Security Association Establishment Frameworks:** In such schemes, entity A and a Central Key Distribution Server authenticate each other and derive a M2M Secure Connection key (Kc) that the Central Key Distribution Server delivers to entity B. The entities then authenticate each other using the M2M Secure Connection key (Kc). The oneM2M authentication Frameworks using centralized key distribution are:

  - **GBA based Security Association Establishment**. This Security Association Establishment Framework uses 3GPP or 3GPP2 symmetric keys to authenticate entity A and the Central Key Distribution Server. The details are specified by 3GPP [13] and 3GPP2 [14]. For more details see clause 8.2.3.2.

  - **M2M Authentication Function (MAF)-based Security Association Establishment**. This Security Association Establishment Framework uses symmetric keys to authenticate the entity A and the Central Key Distribution Server. For more details see clause 8.2.3.1.

For a more detailed description of the above Security Association Establishment Frameworks, it is useful to compare the following aspects of the Security Association Establishment Frameworks:

- **Credential Configuration:**

  - For Central Key Distribution Security Association Establishment Frameworks:

    - Entity A is configured with (or otherwise establishes) the Master Credential (Km) that the entity A will use to authenticate the entity A to the Central Key Distribution Server.

    - The Central Key Distribution Server is configured with the Master Credential (Km) that will be used to authenticate the Central Key Distribution Server to entity A.

    The details for the GBA-Based Security Association Establishment Framework are out of scope.

  - For the Provisioned M2M Secure Connection Key Security Association Establishment Framework, each entity is provisioned with the Pre-Provisioned M2M Secure Connection Key that entities will use to authenticate each other using pre-provisioning or remote provisioning.

  - For the Certificate-Based Security Association Establishment Frameworks, each entity is pre-provisioned with the Credential that the entity will use to authenticate itself to the other entity.

- **Association Configuration:** Configuration of entity identifiers (that is, CSE-ID or AE-ID) for the entities to be authenticated.

  Additionally, in the case of Certificate-Based Authentication Framework: each entity is configured with the Certificate Name and Root of Trust that the entity will use to verify the other entity.

- **Association Security Handshake:** Identification, authentication and security context establishment between the entities.

  - **Central Key Distribution Server Handshake**: When a Centralized Security Association Establishment Framework is used, entity A and the Central Key Distribution Server (MAF or GBA-BSF) perform mutual authentication and generate a M2M Secure Connection Key (Kc) which is then used in the Security Handshake for mutual authentication between entity A and entity B. This is not applicable to Direct Security Association Establishment Frameworks.

Figure 8.2.1-1 provides a summary of the above defined four Security Association Establishment Frameworks.

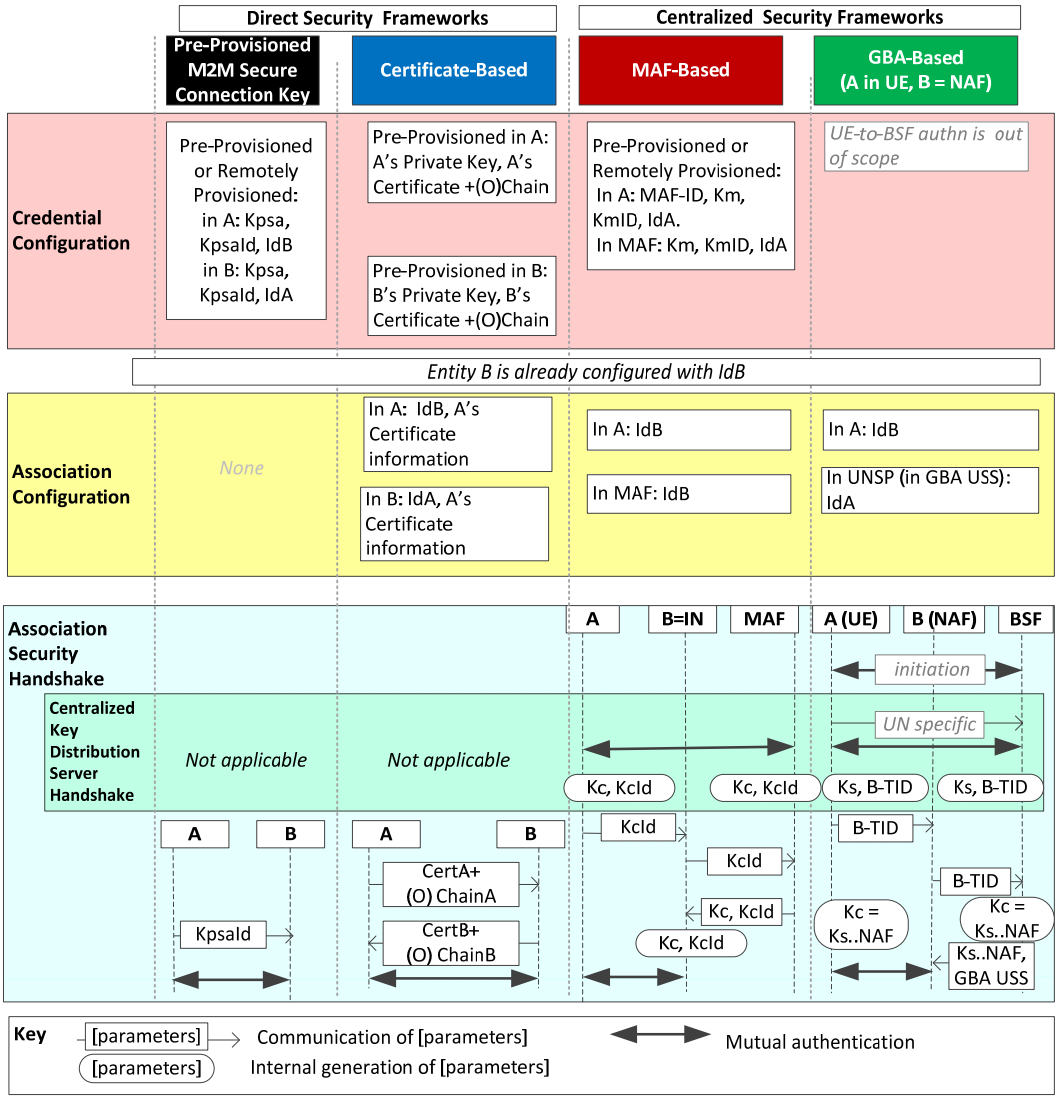


Figure 8.2.1-1: Overview of the Security Association Establishment Frameworks supported by oneM2M.

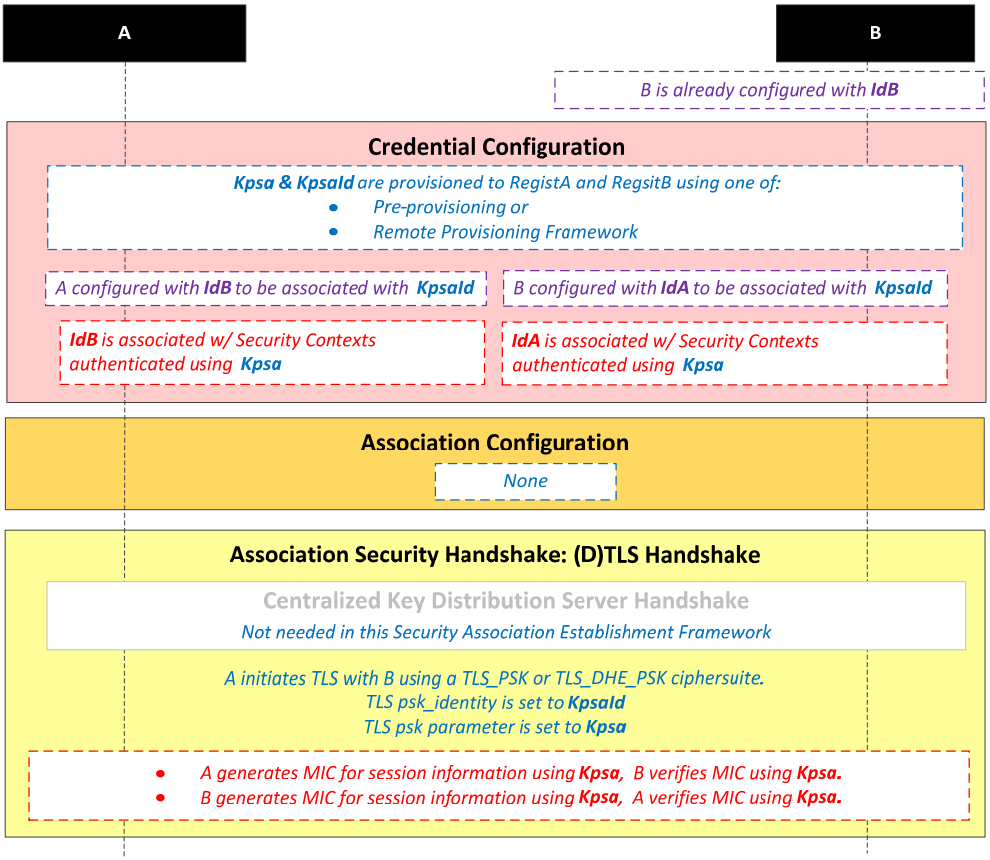
8.2.2 Direct Security Association Establishment Frameworks

8.2.2.1 Provisioned Symmetric Key Security Association Establishment Frameworks

This clause describes the Provisioned Secure Connection Key Security Association Establishment Framework. This framework enables mutual authentication of two entities corresponding to either two CSEs or a CSE and an AE. The Credential for this framework is a long-term symmetric key that has been provisioned into the entities to be authenticated. This key is called a Provisioned Secure Connection Key and is denoted Kpsa. The provisioning of Kpsa could be a pre-provisioning or a remote provisioning thanks to Remote Security Provisioning Frameworks, as described in clause 8.3. The entities authenticate each other by verifying message authentication codes in the Security Handshake which were generated using the Provisioned Secure Connection Key.

NOTE: Long term Provisioned Secure Connection Keys can pose a security risk if not adequately secured, and for this reason long term Provisioned Secure Connection Keys are recommended to be stored in Secure Environments.

Figure 8.2.2.1-1 illustrates the sequence of events when using the Provisioned Secure Connection Key Security Association Establishment Framework. In this description, "Entity A" and "Entity B" correspond to either two CSEs or a CSE and an AE or an AE and a CSE (respectively).



NOTE: The following font colours differentiate the general topic that the text relates to:
Blue italic text highlights details specific to this particular Security Association Establishment Framework.
Purple italic text highlights technical actions that may include steps not specified by oneM2M.
Red italic text highlights security-related properties.

Figure 8.2.2.1-1: The sequence of events when using the Pre-Provisioned Secure Connection Key Security Association Establishment Framework

Credential Configuration: The Provisioned Secure Connection Key (Kpsa) and the corresponding Provisioned Secure Connection Key Identifier, denoted KpsaId, are provisioned to both entities either with pre-provisioning or remote provisioning.

NOTE 1: The provisioning (by definition) uses mechanisms not specified by oneM2M. The remote provisioning is performed thanks to Security Bootstrap Frameworks described in clause 8.3.

Additionally:

Entity A is configured with Entity B identity (IdB). Entity A is to use this identity for Entity B authenticating using the above arguments. This identity is also used to route the (D)TLS exchange.

NOTE 2: Entity A will associate Entity B's identity with messages secured within Security Contexts established using the Provisioned Secure Connection Key Kpsa associated with the Provisioned Secure Connection Key Identifier KpsaId.

- Entity B is configured Entity A identity (IdA). Entity B is to use this identity for Entity A authenticating using the above arguments.

NOTE 3: Entity B will associate the configured Entity A identity with messages secured within Security Contexts established using the Provisioned Secure Connection Key Kpsa associated with the Provisioned Secure Connection Key Identifier KpsaId.

Association Configuration: Each entity is configured with the information needed for mutual authentication and identification:

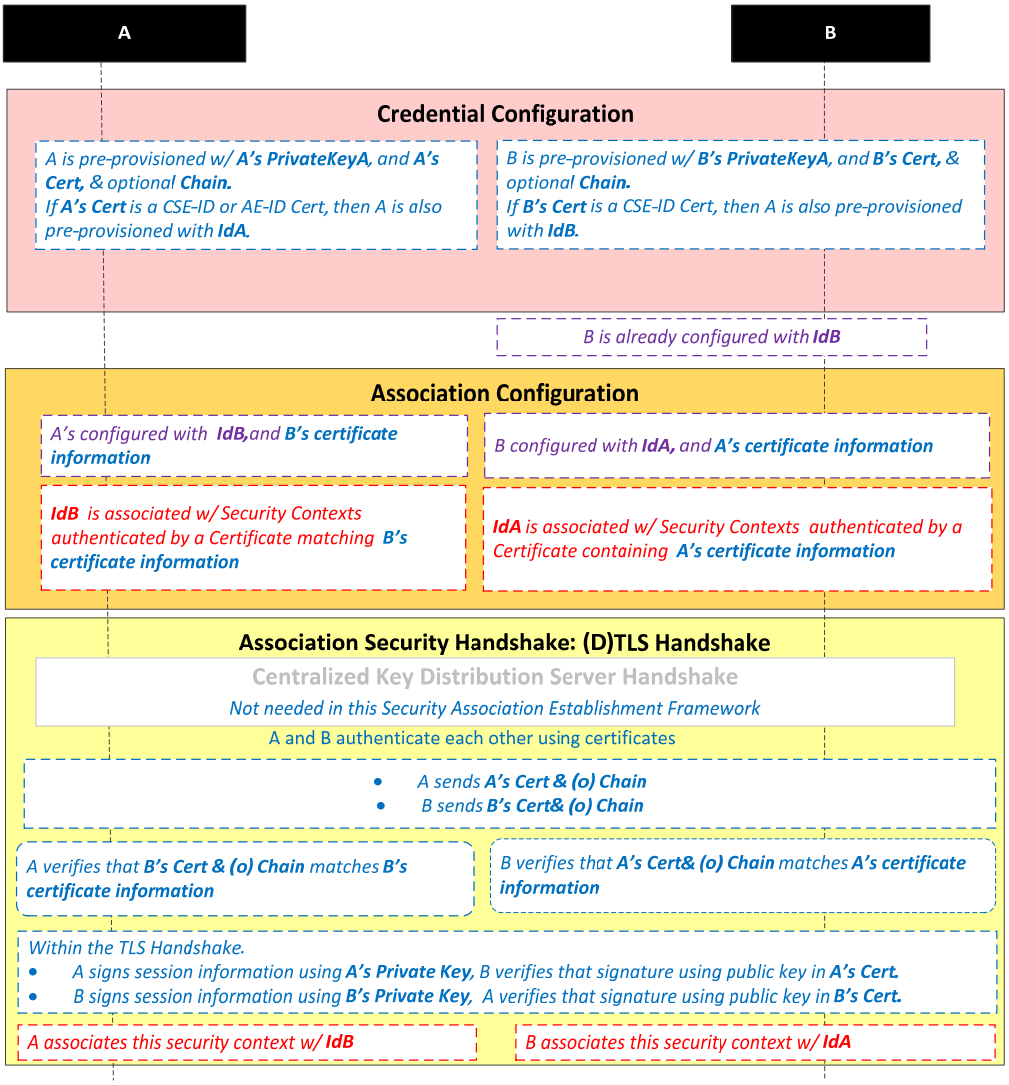
Association Security Handshake: The entities perform a (D)TLS-PSK handshake [15] to establish a secure session.

- Centralized Key Distribution Server Handshake: There is no Centralized Key Distribution Server Handshake applied in the Provisioned Secure Connection Key Security Association Establishment Framework.
- The "psk_identity" parameter [15] is set to the value of the Provisioned Secure Connection Key Identifier KpsaId.
- The "psk" parameter [15] is set to the value of the Provisioned Secure Connection Key Kpsa.
- The (D)TLS cipher suite profile for the Provisioned Secure Connection Key Security Association Establishment Framework is specified in clause 10.2.2.

8.2.2.2 Certificate-Based Security Association Establishment Frameworks

This clause describes the Certificate-Based Security Association Establishment Framework.

Figure 8.2.2.2-1 illustrates the sequence of events when using the Certificate-Based Security Association Establishment Framework. In this description, "Entity A" and "Entity B" correspond to either two CSEs or a CSE and an AE.



NOTE: The following font colours differentiate the general topic that the text relates to:
Blue italic text highlights details specific to this particular Security Association Establishment Framework.
Purple italic text highlights technical actions that may include steps not specified by oneM2M.
Red italic text highlights security-related properties.

Figure 8.2.2.2-1: The sequence of events when using the Certificate-Based Security Association Establishment Framework

Credential Configuration: The private keys and certificates for each entity are pre-provisioned as described in clause 8.1.1.2.1 "Credential Configuration for Certificate-Based Security Frameworks".

Association Configuration: Entity A and Entity B are configured with the information needed for the authentication and identification (during Security Handshake) of Entity B and Entity A respectively:

- Entity A is commanded to initiate a Security Handshake, and the command includes the following arguments:
  - Entity B's certificate information: as described in clause 8.1.1.2.4 "Information Needed for Certificate Authentication of another Entity"
  - Entity B's identity (IdB). Entity A is to use this identity for Entity B authenticating using the above arguments. This is used to route the (D)TLS exchange.

NOTE 1: The Entity A will associate Entity B's identity with messages secured within Security Contexts established in accordance with the configured Entity B‘s certificate information.

- The Entity B is configured with the following arguments describing Entity A authorized to perform Security Handshake with Entity B:
    - Entity A’s certificate information: as described in clause 8.1.1.2.4 “Information Needed for Certificate Authentication of another Entity”.
    - Entity A’s identity (IdA). Entity B is to use this entity identity for Entity A authenticating using the above arguments.

NOTE 2: Entity B will associate Entity A's identity with messages secured within Security Contexts established in accordance with the configured Entity A‘s certificate information.

Association Security Handshake:

- **Centralized Key Distribution Server Handshake:** There is no Centralized Key Distribution Server Handshake applied in the Certificate-Based Security Association Establishment Framework.
- Each entity verifies the other entity's certificate as described in clause 8.1.1.2.2 "Certificate Verification".
- The entities authenticate each other using the validated certificates as specified in TLS 1.2 RFC 5246 [16] and DTLS 1.2 RFC 6347 [17] specifications.
- The (D)TLS cipher suite profile for the Certificate-Based Security Association Establishment Framework is specified in clause 10.2.3.

8.2.3 Centralized Security Association Establishment Frameworks

8.2.3.1 MAF-Based Symmetric Key Security Association Establishment Frameworks

This clause describes the MAF-based Security Association Establishment Framework.

This release addresses the scenario where the Entity B is an Infrastructure Node.

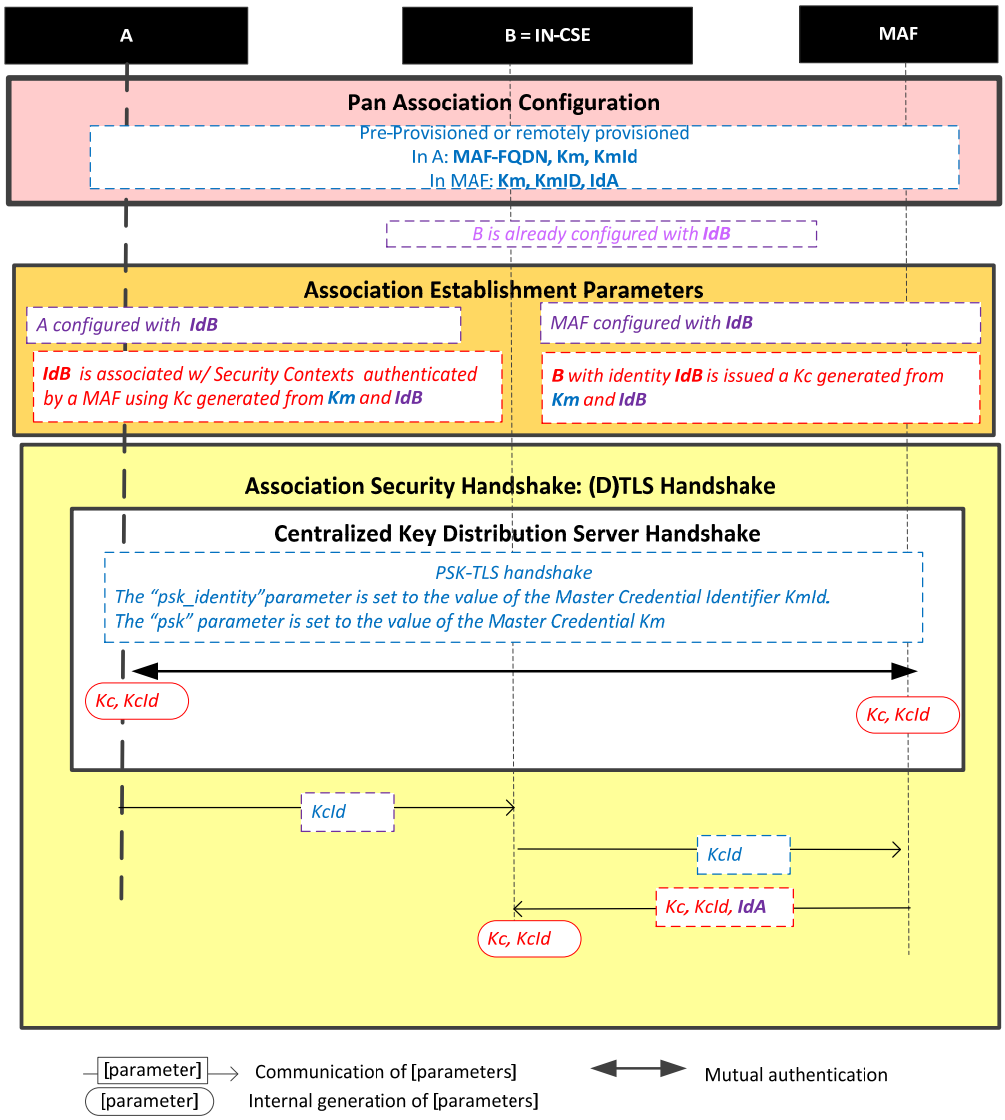


Figure 8.2.3.1-1: The sequence of events when using the MAF-Based Security Association Establishment Framework.

NOTE: The following font colours differentiate the general topic that the text relates to:
Blue italic text highlights details specific to this particular Security Association Establishment Framework.
Purple italic text highlights technical actions that may include steps not specified by oneM2M.
Red italic text highlights security-related properties.

Credential Configuration: The Master Credential (Km) and corresponding Master Credential Identifier (KmId) are either pre-provisioned for Entity A and the MAF or remotely provisioned thanks to Security Bootstrap Frameworks described in Clause 8.3.

Association Configuration: Entity A, Entity B, and the BSF shall be configured with the information needed for the authentication and identification during Centralized Key Distribution Server Handshake and Association Security Handshake:

- Entity A has to know Entity B Identity (IdB)

- Entity B has to know Entity A Identity (IdA)
- The MAF has to know Entity B Identity (IdB). In this case, IdB corresponds to IN Identity.

Association Security Handshake with Centralized Key Distribution Server Handshake:

- **Centralized Key Distribution Server Handshake**
    - The Centralized Key Distribution Server Handshake in MAF-based Security Association Establishment framework enables the establishment of a M2M Secure Connection Key (Kc) and associated M2M Secure Connection Key Identifier (KcId) shared between the Entity A and the MAF thanks to (D)TLS-PSK handshake [15]
        - The “psk_identity” parameter [15] is set to the value of the Master Credential Identifier KmId.
        - The “psk” parameter [15] is set to the value of the Master Credential Km.
- Entity A sends KcId to Entity B (Infrastructure Node).
- Entity B (Infrastructure Node) can retrieve the M2M Secure Connection Key (Kc) from the MAF.
- The (D)TLS cipher suite profile for the MAF-Based Security Association Establishment Framework is specified in clause 10.2.2.

8.2.3.2 GBA-Based Security Association Establishment Frameworks

This clause describes the GBA-based Security Association Establishment Framework.

To obtain a short term key (Kc) used for M2M Service Connection between a M2M Application Service/Middle Node and a M2M Infrastructure Node, the M2M Application Service/Middle Node shall perform a successful GBA bootstrapping and derive NAF keys (Ks_(ext/int)_NAF). This NAF key is the M2M Secure Connection Key (Kc) used for M2M Service Connection.

In case of GBA, ME, Kc = Ks_NAF.

In case of GBA_U, Kc = Ks_int_NAF, if the application resides in the UICC. Otherwise, Kc = Ks_ext_NAF.

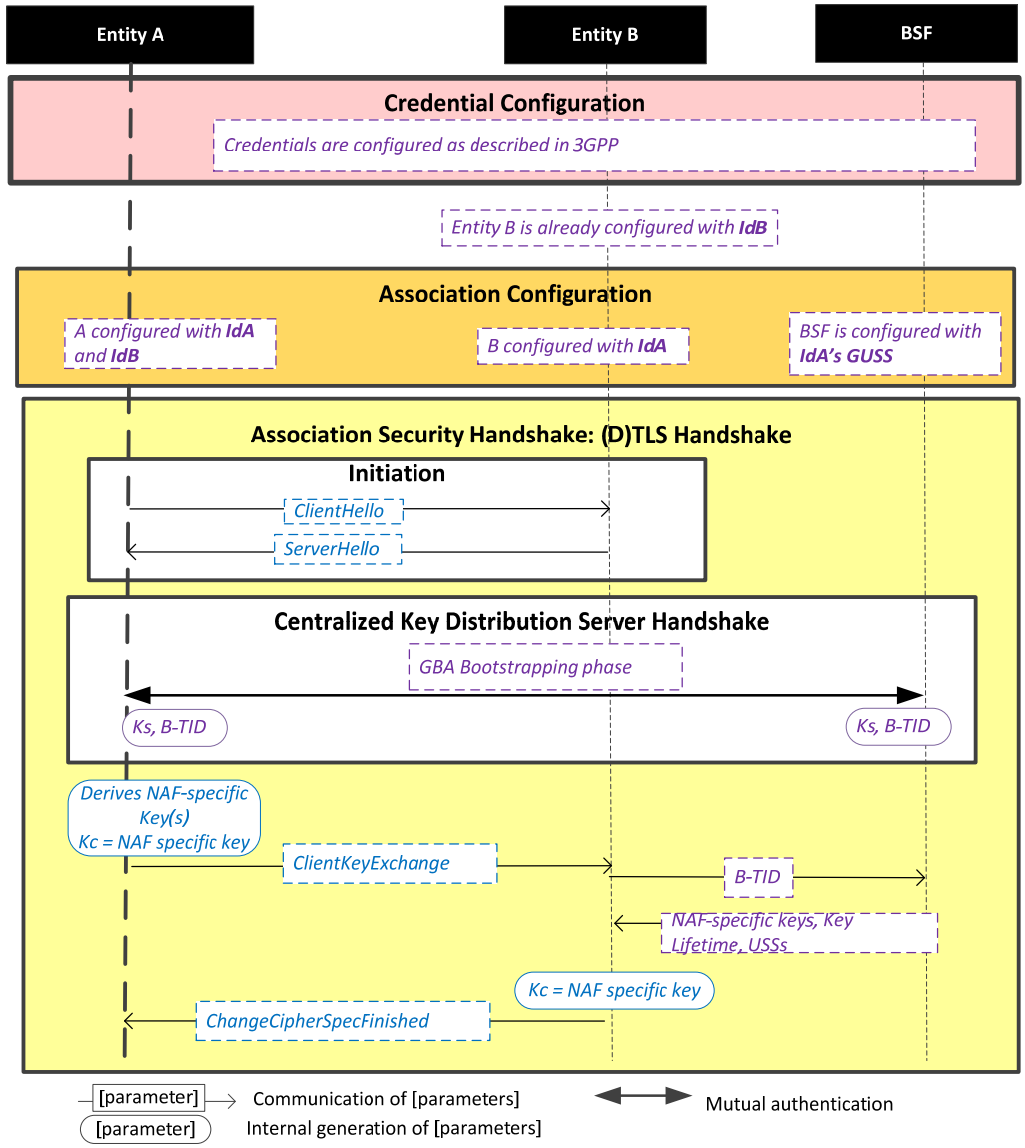


Figure 8.2.3.2-1: The sequence of events when using the GBA-Based Security Association Establishment Framework.

NOTE: The following font colours differentiate the general topic that the text relates to:
Blue italic text highlights details specific to this particular Security Association Establishment Framework.
Purple italic text highlights technical actions that may include steps not specified by oneM2M.

Credential Configuration: The credentials configuration for Entity A, Entity B and the BSF is described in 3GPP specification TS 33.220 [13]. The credentials used to perform mutual authentication between Entity A and BSF are UNSP specific.

Association Configuration: Entity A, Entity B, and the BSF shall be configured with the information needed for the authentication and identification during Centralized Key Distribution Server Handshake and Association Security Handshake:

- Entity A's GBA User Security Settings (GUSS) enables to indicate if Entity A is allowed to establish a NAF-specific key with Entity B (NAF) or/and if the BSF can distribute a NAF specific key to Entity B.

Association Security Handshake with Centralized Key Distribution Server Handshake:

- **Initiation**
    - Entity A and Entity B shall initiate TLS handshake thanks to procedure described in 3GPP TS 33.222 [28]. Informative Annex F of 3GPP TS 24.109 [29] gives signalling flows for TLS-PSK with GBA bootstrapped security association
    - Entity A shall indicate to Entity B that it supports PSK-based TLS by adding one or more PSK ciphersuites to the ClientHello message. This message shall also contain the hostname of Entity B in the server_name extension.
    - Entity B selects one of the PSK-based ciphersuites offered by Entity A and sends it back in the ServerHello message. If UICC is used as M2M Secure Environment supporting Security Association Establishment, GBA-U with Kc = Ks_int_NAF shall be used for authentication and key exchange, In this case the ServerKeyExchange message shall contain a constant string "3GPP-bootstrapping-uicc" as the PSK-identity hint, indicating use of Kc=Ks_int_NAF. Otherwise the ServerKeyExchange message shall contain a constant string "3GPP-bootstrapping" as the PSK-identity hint to indicate that Kc=Ks_NAF in the case of GBA_ME or Kc=Ks_ext_NAF in the case of GBA_U is used for authentication and key exchange. Entity B shall finish the reply to Entity A by sending a ServerHelloDone message.
- **Centralized Key Distribution Server Handshake**
    - The Centralized Key Distribution Server Handshake in Security Association Establishment framework enables the establishment of a GBA bootstrapped key (Ks) shared between the Entity A and the BSF with associated Bootstrapping Transaction Identifier (B-TID) and key lifetime, by performing to the GBA Bootstrapping phase described in 3GPP TS 33.220 [13].
    - If a bootstrapped key Ks is already shared between the Entity A and the BSF and still valid, then the Centralized Key Distribution Server Handshake phase is not needed. The Association Security Handshake can take place with the existing GBA Bootstrapped key Ks.
    - Entity A shall derive the NAF-specific keys by performing the procedure described in 3GPP TS 33.220 [13] using the GBA bootstrapped key Ks and B-TID obtained during Centralized Key Distribution Server Handshake. In case of GBA_ME, the M2M Secure Connection Key (Kc) shall be Ks_NAF. In case of GBA_U, the M2M Secure Connection Key (Kc) shall be Ks_int_NAF if HTTP Client application resides in the UICC. Otherwise, Kc = Ks_ext_NAF.

- Entity A then sends a ClientKeyExchange message with PSK-identity containing a prefix "3GPP-bootstrapping-uicc" if Entity A resides in the UICC is used as the M2M Secure Environment supporting Security Association Establishment procedure or "3GPP-bootstrapping" otherwise, a separator character ";" and the B-TID. Entity A shall conclude the TLS handshake by sending the ChangeCipherSpec and Finished messages to the Network M2M Node.
- Entity B shall extract B-TID from the ClientKeyExchange message and use it to retrieve Kc=Ks_NAF or Kc=Ks_ext_NAF or Kc=Ks_int_NAF, and associated key lifetime from the BSF. The retrieval shall be done over the Zn interface as specified in 3GPP TS 29 109 [30]. As a result, Entity A and Entity B share the NAF-specific key which is to be used as the M2M Secure Connection Key (Kc).
- Entity A shall conclude the TLS handshake by sending the ChangeCipherSpec and Finished message to Entity A.
- The (D)TLS cipher suite profile for the GBA-Based Security Association Establishment Framework is specified in clause 10.2.2.

8.3 Remote Security Provisioning Frameworks

8.3.1 Overview on Remote Security Provisioning Frameworks

8.3.1.1 Purpose of Remote Security Provisioning Frameworks

Security Bootstrap Frameworks enable the provisioning of a symmetric key shared between two entities. Those two entities can be either a CSE/AE and a M2M Authentication Function (MAF) or two CSEs/AEs.

The provisioned symmetric key can be used for Provisioned Symmetric Key Security Association Establishment Framework or MAF-based Symmetric Key Security Association Establishment Frameworks.

- **Provisioned Symmetric Key Security Association Establishment**

Provisioned Symmetric Key Association Establishment uses a symmetric key Kpsa and corresponding KpsaId, shared between two entities (Entity A and Entity B), to establish security associations between those two entities (CSE/AEs), as described in clause 8.2.2.1. This symmetric key Kpsa and corresponding KpsaId shall be either pre-provisioned or remotely provisioned to the two CSE/AEs thanks to Security Bootstrap Frameworks.

- **MAF-based Symmetric Key Security Association Establishment**

The MAF-based Security Association Establishment Framework uses a Master Credential (Km) and corresponding Master Credential Identifier (KmId), shared by a CSE/AE and an M2M Authentication Function, to establish security associations between the CSE/AE and other CSEs and/or AEs as described in clause 8.2.3.

The Master Credential (Km) and corresponding Master Credential Identifier (KmId) shall either be pre-provisioned or remotely provisioned to the CSE/AE and M2M Authentication Function.

- **General**

The method for pre-provisioning can be deployment dependent. An interoperable pre-provisioning framework based on UICC is described in Annex D.

Clause 8.3 describes the set of remotely provisioning mechanisms; called *Remote Security Provisioning Frameworks*. An M2M Enrolment Function facilitates the remote provisioning.

8.3.1.2 Overview on Remote Security Provisioning Frameworks

An AE or CSE that requires remote provisioning of a Master Credential and Master Credential Identifier or a Provisioned Secure Connection Key (Kpsa) and Provisioned Secure Connection Key Identifier (KpsaId) is called an *Enrolee.* The AE or CSE with whom the enrolee is to establish a security association is called *Enrolee B.* The AE or CSE or M2M Authentication Function with whom the enrolee is to establish a shared key is called an *Enrolment Target*.

The oneM2M system supports the following Security Bootstrap Frameworks:

- **Centralized Remote Security Provisioning Frameworks**

  - **Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework:** A symmetric key is pre-provisioned to the Enrolee and M2M Enrolment Function for the mutually authentication of those entities. For more details, see clause 8.3.2.1.

  - **Certificate-Based Remote Security Provisioning Framework:** The Enrolee and M2M Enrolment Function are each issued with

    - a Private Signing Key that is known only to that entity,

    - a Certificate containing the corresponding Public Verification Key, and

    - (In the case of a device certificate, CSE-ID certificate or AE-ID certificate) a Certificate Chain from the entity's Certificate to a Trust Anchor Certificate.

    The Enrolee and M2M Enrolment Function shall validate each other's Certificate before trusting the Public Verification Keys in the Certificate. Within the Security Handshake, the M2M Enrolment Function creates a digital signature of the session parameters using its private signing key and the Enrolee verifies the digital signature using the M2M Enrolment Function's public verification key. Then the roles are reversed: the Enrolee creates a digital signature and the M2M Enrolment Function verifies it. For more details see clause 8.3.2.2.

  - **GBA-based Remote Security Provisioning Framework**. In this case, the role of the M2M Enrolment Function is performed by a GBA Bootstrap Server Function. This framework uses 3GPP or 3GPP2 symmetric keys to authenticate the Enrolee and the M2M Enrolment Function (which is also a GBA BSF). The details are specified by 3GPP [13] and 3GPP2 [14]. For more details see clause 8.3.2.3.

For a more detailed description of the above Remote Security Provisioning Frameworks, it is useful to compare the following aspects of the Remote Security Provisioning Frameworks.

- **Bootstrap Credential Configuration:** The Enrolee and M2M Enrolment Function are pre-provisioned with the Bootstrap Credential that the entity will use to authenticate itself to the other entity. The mechanisms for this pre-provisioning are not described in this specification.

- **Bootstrap Instruction Configuration:** The Enrolee and M2M Enrolment Function are provided with

  - Either the M2M Authentication Function Identifier (MAF-ID) identifying the M2M Authentication Function for which the Enrolee is to be remotely provisioned when used in conjunction with a MAF-based security association establishment framework; or the identifier of Enrolee B (Enrolee B-ID), when used in conjunction with a Provisioned Symmetric Key Security Association Establishment.

NOTE 1: The identity of the M2M Authentication Function or the Enrolee B is assumed to have been configured prior to the Bootstrap Instruction Configuration phase.

  - The M2M Enrolment Function is provided with the CSE-ID or AE-ID that the M2M Authentication Function or Enrolee B is to associate with the Enrolee.

  Additionally, in the case of Certificate-Based Remote Security Provisioning Framework:

  - The Enrolee is configured with the M2M Enrolment Function URI (for the purpose of routing the (D)TLS messages to the M2M Enrolment Function), M2M Enrolment Function Certificate Name and M2M Enrolment Function Trust Anchor Certificatesthat the Enrolee will use to verify the M2M Enrolment Function.

  - The M2M Enrolment Function is configured with the Enrolee Certificate Name and Enrolee Root of Trust that the M2M Enrolment Function will use to verify the Enrolee.

- **Bootstrap Enrolment Handshake:** Identification, authentication and security context establishment between the Enrolee and M2M Enrolment Function.

- **Enrolment Key Generation:** generating a symmetric Enrolment Key,(Ke) and corresponding Enrolment Key Identifier (KeId) shared by the Enrolee and M2M Enrolment Function, which is used for subsequent generation of the Master Credential (Km) or Provisioned M2M Secure Connection Key (Kpsa).

- **Integration to the Association Security Handshake:**

  For MAF-based symmetric Key Security Association, the following steps occur during the Centralized Key Distribution Server Handshake of the MAF-Based Security Association Establishment:

  - The Enrolee derives the Master Credential (Km) from the Enrolment Key (Ke) and M2M Authentication Function Identifier (MAF-ID). Details of the derivation are provided in clause 9.4.

  - The Enrolee generates the Master Credential Identifier (KmId) from Master Credential (Km) as described in clause 9.1, and stores Km and KmId.

  - The Enrolee passes the Enrolment Key Identifier (KeId) to the M2M Authentication Function (see "Centralized Key Distribution Server Handshake" in clause 8.2.3.1)

NOTE 2: When the Enrolee first communicates with the M2M Authentication Function, then the M2M Authentication Function has not yet retrieved the Km from the M2M Enrolment Function. Consequently, the Enrolee provides the KeId to the M2M Authentication function, which is then passed to the M2M Enrolment Function to identify the Enrolment Key. The M2M Enrolment Function then returns the Km from which the M2M Authentication Function can derive the KmId. In subsequent Security Establishments, the Enrolee may provide the KmId or the KeId, and the M2M Authentication Function will know that both identifiers indicate the retrieved Km. For more details, see "Centralized Key Distribution Server Handshake" in clause 8.2.3.1.

  - Upon receipt of the KeId, the M2M Authentication Function determines if it already has the corresponding Km and CSE-ID or AE-ID of the Enrolee

- If the M2M Authentication Function already has the corresponding Km and CSE-ID or AE-ID of the Enrolee, then the Km is used for mutual authentication (see "Centralized Key Distribution Server Handshake" in clause 8.2.3.1)
- If the M2M Authentication Function does not have the corresponding Master Credential (Km) and CSE-ID or AE-ID of the Enrolee, then the following steps are followed.
  - The M2M Authentication Function (securely) passes the KeId to the M2M Enrolment Function, along with the M2M Authentication Function's URI.
  - The M2M Authentication Function initiates establishing a mutually-authenticated TLS Session with the M2M Enrolment Function.
    - The M2M Authentication Function authenticates itself to the M2M Enrolment Function using an FQDN certificate containing the FQDN of the M2M Authentication Function.
    - The M2M Enrolment Function authenticates itself to the M2M Authentication Function using an FQDN certificate containing the FQDN of the M2M Enrolment Function.
  - The M2M Enrolment Function derives the Km from the Ke and MAF-ID. Details of the derivation are provided in clause clause 10.3.2“Derivation of Master Credential from Enrolment Key”.
  - The M2M Enrolment Function returns the Km to the M2M Authentication Function. The M2M Enrolment Function also passes the CSE-ID or AE-ID of the Enrolee.
  - The M2M Authentication Function generates the Master Credential Identifier (KmId) from Master Credential (Km) as described in clause 9.1, and stores Km and KmId.
  - The Master Credential (Km) is used for mutual authentication and generation of Kc and KcId as described in MAF-Based Security Association Establishment Framework (see "Centralized Key Distribution Server Handshake" in clause 8.2.3.1).
  - The Enrolee and M2M Authentication Function set Master Credential Identifier (KmId) to the value of the Enrolment Key Identifier (KeId).
  - The Enrolee and M2M Authentication Function store Km and KmId.

For Provisioned Symmetric Key Security Association Establishment, similar procedure applies where Enrolee B plays the role of M2M Authentication Function, Kpsa plays the role of Km, KpsaId is generated instead of KmId. Generation of Kpsa is described in 10.3.3 “Derivation of Provisioned Secure Connection Key from Enrolment Key”.

Figure 8.3.1.2-1 provides a summary of the above defined Remote Security Provisioning Frameworks.

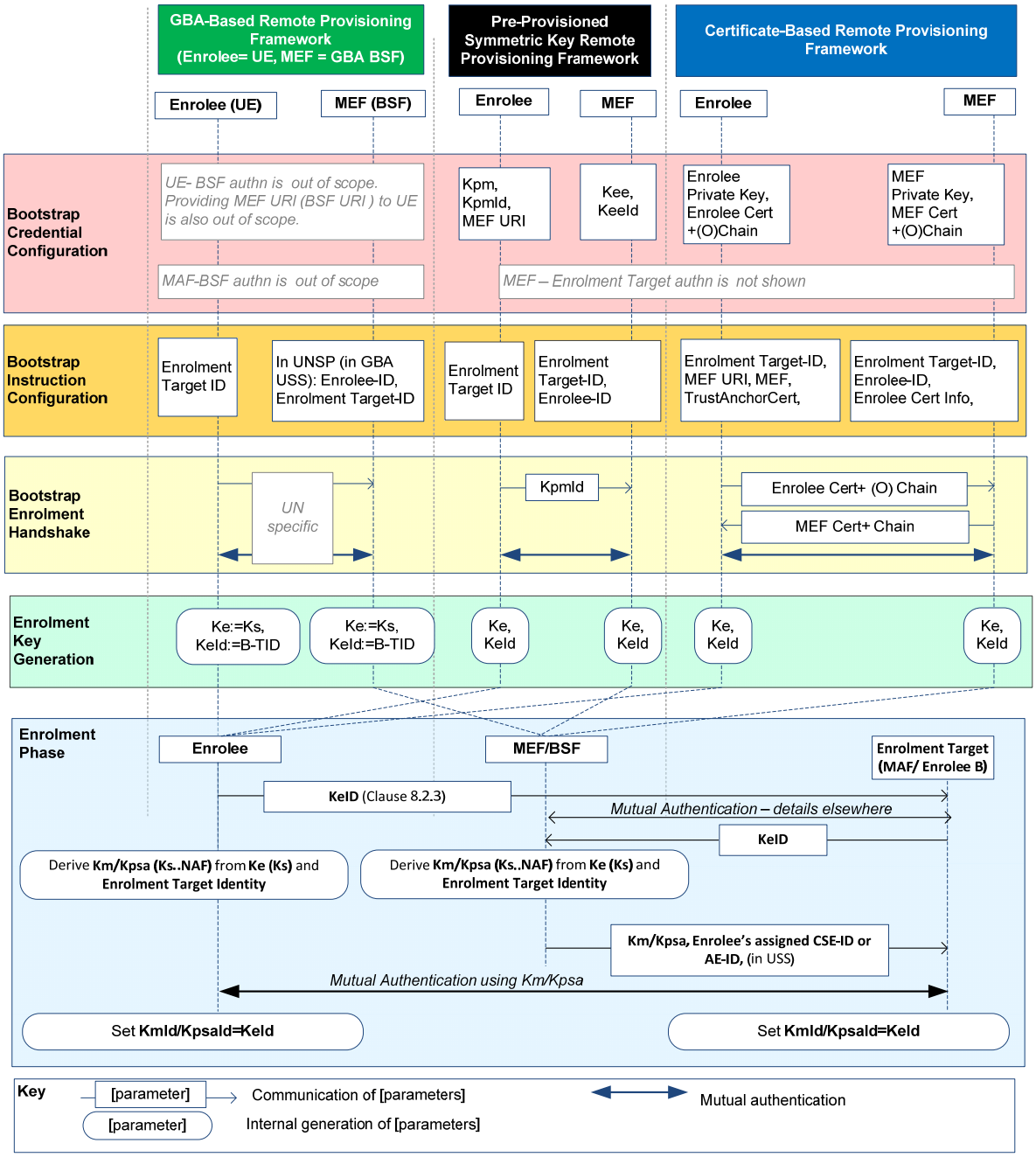


Figure 8.3.1.2-1: Overview of the Remote Security Provisioning Frameworks supported by oneM2M

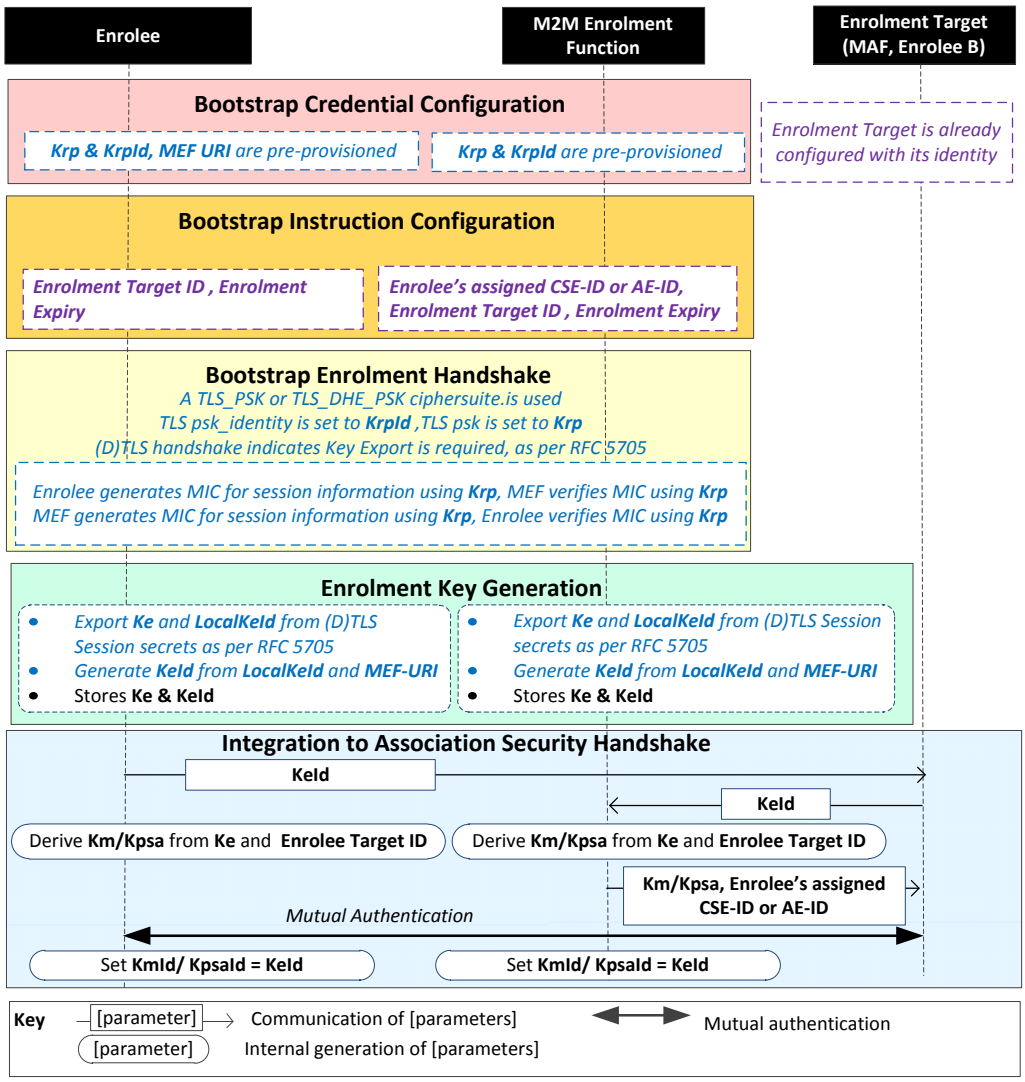
8.3.2 Centralized Remote Security Provisioning Framework

8.3.2.1 Pre-Provisioned Symmetric Key Remote Security Provisioning Framework

This clause describes the Pre-Provisioned Symmetric Key Remote Security Provisioning Framework. The Bootstrap Credential for this framework is a long-term symmetric key that has been pre-provisioned into the Enrolee and M2M Enrolment Function; this key is called a Pre-Provisioned Symmetric Enrolee Key and is denoted Kpm.

NOTE: Long term Pre-Provisioned Symmetric Enrolee Keys can pose a security risk if not adequately secured, and for this reason it is recommended that Long term Pre-Provisioned Symmetric Enrolee Keys are stored in Secure Environments.

Figure 8.3.2.1-1 illustrates the sequence of events when using the Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework.



NOTE: The following font colours differentiate the general topic that the text relates to:
Black text contains Remote Security Provisioning-Framework-independent details
Blue italic text highlights details specific to this particular Remote Security Provisioning Framework.
Purple italic text highlights technical actions that may include steps not specified by oneM2M.

Figure 8.3.2.1-1: The sequence of events when using the Pre-Provisioned Symmetric Key Remote Security Provisioning Framework

Bootstrap Credential Configuration: The Pre-Provisioned Symmetric Enrolee Key (Kpm) and the corresponding Pre-Provisioned Symmetric Enrolee Key Identifier, denoted KpmId, are pre-provisioned to both entities. The Enrolee is also provisioned with the M2M Enrolment Function's URI (MEF URI), for the purpose of routing the (D)TLS exchange.

NOTE 1: This pre-provisioning (by definition) uses mechanisms not specified by oneM2M.

Bootstrap Instruction Configuration: The Enrolee and M2M Enrolment Function are configured with the information needed for authorizing the remote provisioning:

- The Enrolee is configured with (or otherwise obtains) the following arguments to initiate remote provisioning :
  - The Enrolment Target identity: Identifying the Enrolment Target for which the Enrolee is to be provisioned.
  - The Enrolee associates these arguments with the M2M Enrolment Function. The M2M Enrolment Function can be identified to the Enrolee using the Pre-Provisioned Symmetric Enrolee Key Identifier (KpmId) or the M2M Enrolment Function URI.
- M2M Enrolment Function is configured with the following arguments to authorize the M2M Enrolment Function to remotely provision the Enrolee for an Enrolment Target:
  - The Enrolment Target Identity: Identifying the Enrolment Target for which the Enrolee is to be provisioned.
  - Enrolee's assigned CSE-ID or AE-ID (Enrolee-ID). The M2M Enrolment Function is to provide this entity identity for the Enrolee with the Km or Kpsa to the Enrolment Target, when requested by the Enrolment Target.
  - The M2M Enrolment Function associates these arguments with an Enrolee. The Enrolee can be identified to the M2M Enrolment Function using the Pre-Provisioned Symmetric Enrolee Key Identifier (KpmId).

Bootstrap Security Handshake: The Enrolee and M2M Enrolment Function perform a (D)TLS-PSK handshake [15] to establish a secure session.

- The "psk_identity" parameter [15] is set to the value of the Pre-Provisioned Symmetric Enrolee Key Identifier (KpmId).
- The "psk" parameter [15] is set to the value of the Pre-Provisioned Symmetric Enrolee Key (Kpm).
- The (D)TLS cipher suite profile for this 's assigned CSE-ID or AE-ID is specified in clause 10.2.2 "TLS and DTLS Ciphersuites for TLS-PSK-Based Security Frameworks".

Enrolment Key Generation:

a) The Enrolment Key (Ke) and RelativeKeId is generated from the (D)TLS session secrets by the Enrolee and M2M Enrolment Function using TLS Key Export (RFC 5705) [18], as described in clause 10.3.1 "TLS Key Export Details".

b) The Enrolment Key Identifier (KeId) is generated from the RelativeKeId and the M2M Enrolment Function's FQDN by the Enrolee and M2M Enrolment Function, as described in clause 10.3.4 "Generating KeId".

c) The Enrolee and M2M Enrolment Function store the Enrolment Key (Ke) and Enrolment Key Identifier (KeId).

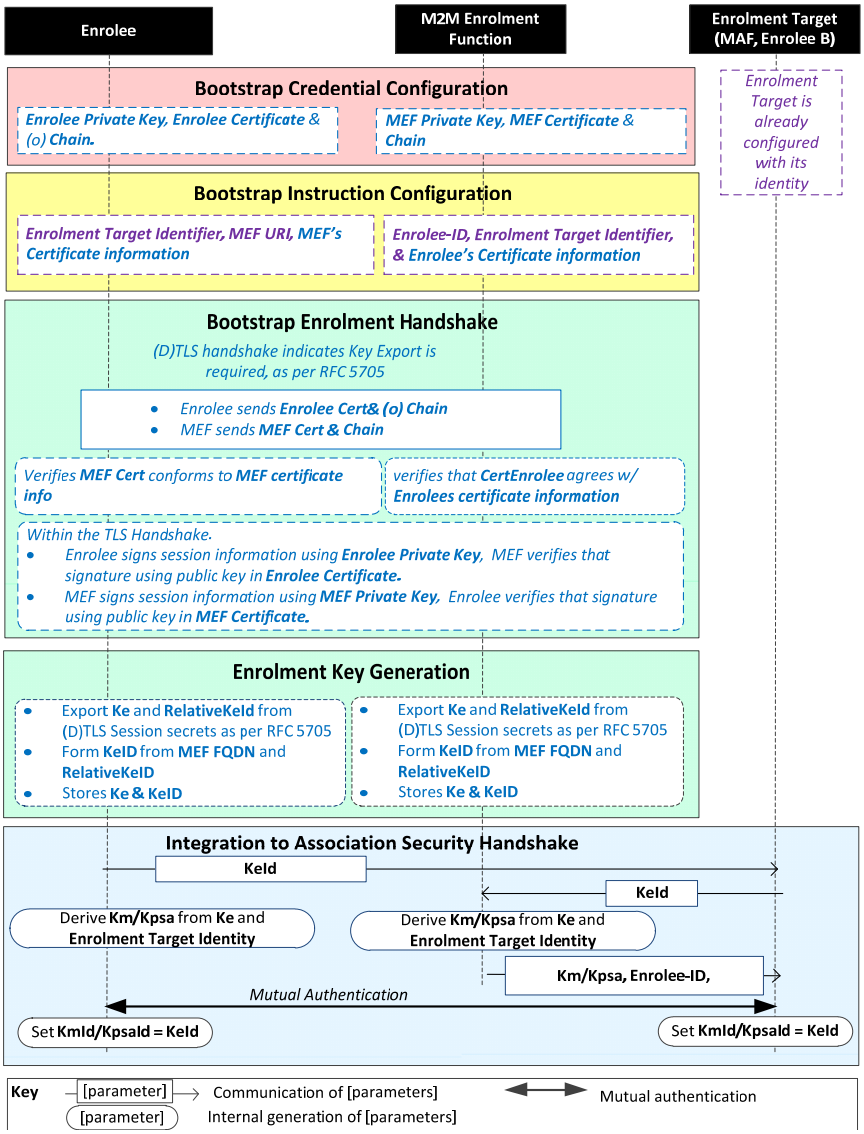
NOTE 2: The Enrolment Key Generation for the Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework is identical to the Enrolment Key Generation for the Certificate-Based Remote Security Provisioning Framework.

Integration to the Association Security Handshake: See "Overview of Remote Security Provisioning Frameworks" in clause 8.3.1.

8.3.2.2 Certificate-Based Remote Security Provisioning Framework

This clause describes the Certificate-Based Remote Security Provisioning Framework. The Bootstrap Credentials for this framework are Certificates.

Figure 8.3.2.2-1 illustrates the sequence of events when using the Certificate-Based Remote Security Provisioning Framework.



NOTE: The following font colours differentiate the general topic that the text relates to:
Black text contains Remote Security Provisioning -Framework-independent details.
Blue italic text highlights details specific to this particular Remote Security Provisioning Framework.
Purple italic text highlights technical actions that may include steps not specified by oneM2M.
Red italic text highlights security-related properties.

Figure 8.3.2.2-1: The sequence of events when using the Certificate-Based Remote Security Provisioning Framework.

Bootstrap Credential Configuration: For this Remote Security Provisioning Framework, Enrolee and M2M Enrolment Function authenticate each other using a Public Key Certificate. The Bootstrap Credentials for the Enrolee and M2M Enrolment Function are pre-provisioned as described in clause 8.1.1.2.1 "Credential Configuration for Certificate-Based Security Frameworks".

NOTE 1: The identities of the M2M Enrolment Function and Enrolment Target are assumed to have been configured prior to this phase.

Bootstrap Instruction Configuration: The Enrolee and M2M Enrolment Function are configured with the information needed for authorizing the remote provisioning:

- The Enrolee is configured with (or otherwise obtains) the following arguments to initiate remote provisioning:

- The URI of the M2M Enrolment Function which will facilitate the remote provisioning, for the purpose of routing the (D)TLS exchange.
- Information needed for certificate authentication of the M2M Enrolment Function using an MEF certificate as described in clause 8.1.1.2.4 “Information Needed for Certificate Authentication of another Entity”;
- The Enrolment Target Identity : Identifying the Enrolment Target for which the Enrolee is to be provisioned.

- The M2M Enrolment Function is configured with the following arguments describing Enrolee authorized to perform Security Handshake with M2M Enrolment Function:
  - Information needed for certificate authentication of the Enrolee, as described in clause 8.1.1.2.4 “Information Needed for Certificate Authentication of another Entity”
  - The Enrolment Target Identity : Identifying the Enrolment Target for which the Enrolee (authenticated using the above Enrolee Certificate information) is to be provisioned.
  - The Enrolee’s assigned CSE-ID or AE-ID (Enrolee-ID). The M2M Enrolment Function is to provide this entity identity for the Enrolee with the Km or Kpsa to the Enrolemnt Target, when requested by the Enrolment Target.

Bootstrap Security Handshake: The Enrolee and M2M Enrolment Function perform a (D)TLS handshake as specified in TLS 1,2 RFC 5246 [16] and DTLS 1.2 RFC 6347 [17] specifications.to establish a secure session.

- Each entity (Enrolee and M2M Enrolment Function) verifies the other entity's certificate as described in clause 8.1.1.2.5 "Certificate Verification".
- The Enrolee and M2M Enrolment Function authenticate each other using the validated certificates as specified in TLS 1.2 RFC 5246 [16] and DTLS 1.2 RFC 6347 [17] specifications.
- The (D)TLS cipher suite profile for this ’s assigned CSE-ID or AE-ID is specified in clause 10.2.3 “TLS and DTLS Ciphersuites for Certificate-Based Security Frameworks”.

Enrolment Key Generation:

The steps are identical to those shown for “Enrolment Key Generation” in clause 8.3.2.1 “Pre-Provisioned Symmetric Key Remote Security Provisioning Framework”

Integration to the Association Security Handshake: See "Overview of Remote Security Provisioning Frameworks" in clause 8.3.1.

8.3.2.3 GBA-Based Remote Security Provisioning Framework

To share a long term Master Credential (Km) or Provisioned Secure Connection Key (Kpsa) between an Application Service/Middle Node and an Enrolment Target, the M2M Application Service/Middle Node shall perform a successful GBA bootstrapping and derive a NAF key (Ks_(ext/int)_NAF). This NAF key is the Master Credential (Km) or Provisioned Secure Connection Key (Kpsa).

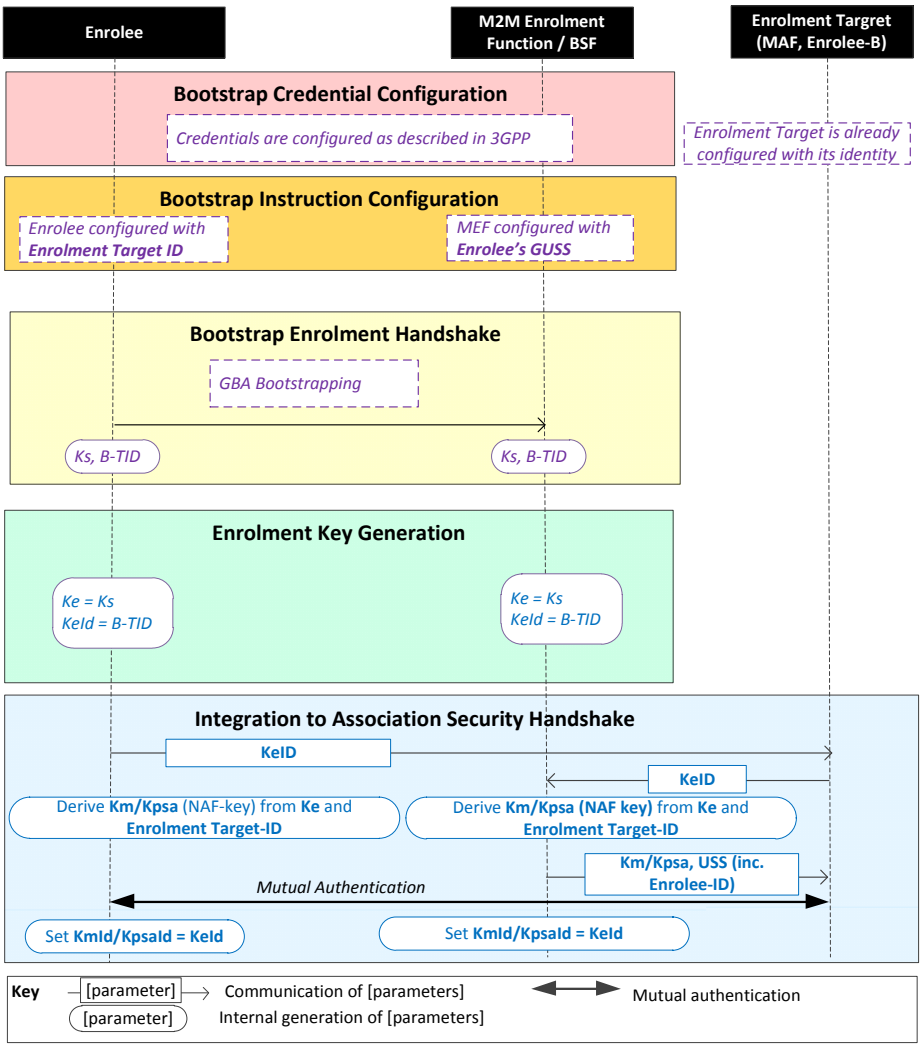


Figure 8.3.2.3-1: The sequence of events when using the GBA-Based Remote Security Provisioning Framework.

NOTE: The following font colours differentiate the general topic that the text relates to:
Black text contains Remote Security Provisioning -Framework-independent details.
Blue italic text highlights details specific to this particular Remote Security Provisioning Framework.
Purple italic text highlights technical actions that may include steps not specified by oneM2M.

Bootstrap Credential Configuration: The credentials configuration for Enrolee and M2M Enrolment Function (MEF) is described in 3GPP specification TS 33.220 [13]. The MEF plays the role of the BSF. The credentials used to perform mutual authentication between Enrolee and MEF are UNSP specific.

Bootstrap Instruction Configuration: the Enrolee, the MEF and the MAF shall be configured with the information needed for authorizing the remote provisioning.

- The Enrolee shall be configured with the Enrolment Target Identity: identifying the Enrolment Target for which the Enrolee is to be provisioned.
- The MEF shall be configured with the Enrolee-ID and the Enrolment Target Identity
    - The Enrolment Target Identity: Identifying the Enrolment Target for which the Enrolee (authenticated using the GBA) is to be provisioned

- The Enrolee's assigned CSE-ID or AE-ID (Enrolee-ID), The M2M Enrolment Function is to provide this entity identity for the Enrolee with the Km or Kpsa to the Enrolment Target, when requested by the Enrolment Target.
- Enrolee's GBA User Security Settings (GUSS) enables to indicate if Enrolee is allowed to establish a NAF-specific key with the MAF or/and if the BSF can distribute a NAF specific key to the MAF.

Bootstrap Enrolment Handshake:

The Bootstrap Enrolment Handshake enables the establishment of a GBA bootstrapped key (Ks) shared between the Enrolee and the MEF with associated Bootstrapping Transaction Identifier (B-TID) and key lifetime, by performing to the GBA Bootstrapping phase described in 3GPP TS 33.220 [13].

If a bootstrapped key Ks is already shared between Enrolee and the MEF and still valid, then the Bootstrap Enrolment Handshake phase is not needed. The Enrolment Key Generation phase can take place with the existing GBA Bootstrapped key Ks.

Enrolment Key Generation phase

The Enrolment Key (Ke) shall be the GBA Bootstrapped key (Ks) established during the Bootstrap Enrolment Handshake.

The Enrolment Key Identifier (Ke-ID) shall be the Bootstrapping Transaction Identifier (B-TID) generated during the Bootstrap Enrolment Handshake.

Integration to the Association Security Handshake:

- The Enrolee and the Enrolment Target shall establish the Master Credential (Km) or the Provisioned Secure Connection Key (Kpsa) thanks to procedures described in 3GPP TS 33.220 [13] using the Enrolment Key (Ke) as GBA bootstrapped key Ks and the Enrolment Key Identifier (Ke-ID) as B-TID. The Enrolment Target plays the role of a NAF.
  - The Enrolee and the Enrolment Target shall establish NAF-specific key(s) as described in 3GPP TS 33.220 [13]. A key lifetime is associated to the NAF-specific keys. The Enrolment Target also receives the Enrolee's User Security Settings (USS) from the MEF/BSF.
    - In case of GBA_ME, NAF-specific key is Ks_NAF
    - In case of GBA_U, NAF-specific keys are Ks_int_NAF and Ks_ext_NAF.
  - The Master Credential (Km)) or the Provisioned Secure Connection Key (Kpsa) shall be the NAF-specific key:
    - In case of GBA_ME, Km/Kpsa = Ks_NAF
    - In case of GBA_U, Km/Kpsa = Ks_int_NAF if HTTP Client application resides in the UICC. Otherwise, Km/Kpsa = Ks_ext_NAF.
  - The Enrolee and the Enrolment Target shall set the Master Credential Identifier (Km-Id) or the Provisioned Secure Connection Key Identifier (Kpsa-Id) to the value of KeId.

Enrolee and Enrolment Target shall perform (D)TLS-PSK handshake (RFC 4279 [15]) with the Master Credential (Km) or Provisioned Secure Connection Key (Kpsa) as Pre-Shared Key in compliance with clause 10.2.2 "TLS and DTLS Ciphersuites for TLS-PSK-Based Security Frameworks". If UICC is used as Secure Environment supporting Remote Security Provisioning, GBA-U with Kc = Ks_int_NAF shall be used for authentication and key exchange.

9 Security Framework Procedures and Parameters

This clause specifies procedures and parameters of the phases of Security Association Establishment Frameworks (clause 8.2) and Remote Security Provisioning Frameworks (clause 8.3).

9.1 Security Association Establishment Framework Procedures and Parameters

9.1.1 Credential Configuration Parameters

The following Credential Configuration procedures are described in the present clause:

- Credential Configuration of Entity A and Entity B , see Clause 9.1.1.1,
- Credential Configuration of M2M Authentication Functions, see Clause 9.1.1.2,

The following Credential Configuration procedures are specified by other organizations:

- Credential Configuration of Underlying Network Service Provider authentication server (e.g. HLR, HSS or AAA) for the GBA-Based Security Association Establishment Framework. These details are specified by 3GPP TS 33.220 [13], 3GPP2 S.S0109-A [14].
- Credential Configuration of Entity A in the Field-Domain for the GBA-Based Security Association Establishment Framework. These details are specified by 3GPP TS 33.220 [13], 3GPP2 S.S0109-A [14].

9.1.1.1 Credential Configuration of Entity A and Entity B

Table 9.1.1.1-1 lists the parameters configured to a Field-Domain Security Association End-Points in the Credential Configuration phase.

Table 9.1.1.1-1 Parameters configured to a Field Domain Security Association end-point during the Credential Configuration phase.

Security Association Establishment Framework		Parameter
Provisioned M2M Secure Connection Key		Kpsa
		KpsaId
Certificate Based	Entity authenticates itself using a Raw Public Key Certificate	Entity's Private Key
		Entity's Raw Public Key Certificate
	Entity authenticates itself using a Device Certificate	Entity's Private Key
		Entity's Certificate and Chain
	Entity authenticates itself using a CSE-ID Certificate	Entity's CSE-ID
		Entity's Private Key
		Entity's Certificate and Chain
	Entity authenticates itself using an AE-ID Certificate	Entity's AE-ID
		Entity's Private Key
		Entity's Certificate and Chain
MAF-Based	Entity A	MAF Identifier (MAF-ID)
		Master Credential (Km Id)
		Master Credential Identifier (KmId)
	Entity B	Entity B and MAF shall be able to establish mutually-authenticated secure communication. The details are not specified in the present document
GBA-Based	Entity B plays role of NAF. Details specified in 3GPP 33.220 [13]. Where this description differs from [13], [13] takes precedence	NAF Private Key
		NAF Certificate and Chain
		BSF trust anchor certificate

The Credential Configuration of Entity A, Entity B and the MAF for the Provisioned M2M Secure Connection Key Security Association Establishment Framework, or the MAF-Based Security Association Establishment Framework is achieved through either:

- Pre-provisioning via mechanisms which are not specified in the present document.
- Remote provisioning via one of the Remote Security Provisioning Frameworks in Clause 8.3.

The Credential Configuration of Entity A and Entity B for the Certificate Security Association Establishment Frameworks is performed by pre-provisioning via mechanisms which are not specified in the present document .

9.1.1.2 Credential Configuration of M2M Authentication Functions

Table 9.1.1.2-1 lists the parameters configured to M2M Authentication Functions in the Credential Configuration phase. The M2M Authentication Function's identifier (MAF-ID) is presumed to have been configured prior to the Credential Configuration phase.

Table 9.1.1.2-1 Parameters configured to a M2M Authentication Functions during the Credential Configuration phase.

Security Association Establishment Framework		Parameter
MAF-Based	A-to-MAF Authentication	Entity A's CSE-ID or AE-ID (IdA)
		masterCredential (Km)
		masterCredentialIdentifier (KmId)
	B-to-MAF Authentication	Entity B and MAF shall be able to establish mutually-authenticated secure communication. The details are not specified in the present document

The Credential Configuration of M2M Authentication Framework shall be achieved through either:

- Business logic of the Stakeholder operating the M2M Authentication Function, and the details are not described in this specification.
- Remote provisioning via one of the Remote Security Provisioning Frameworks in Clause 8.3.

9.1.2 Association Configuration Procedures and Parameters

The following Association Configuration procedures are described in this clause:

- Association Configuration of Entity A and Entity B, see Clause 9.1.2.1,
- Association Configuration of M2M Authentication Functions, see Clause 9.1.2.2, and
- Association Configuration of Underlying Network Service Provider authentication servers (e.g. HLR, HSS or AAA), see Clause 9.1.2.3.

9.1.2.1 Association Configuration of Entity A and Entity B

Table 9.1.2.1-1 lists the parameters configured to Entity A and Entity B in the Credential Configuration phase.

Table 9.1.2.1-1 Parameters configured to Entity A and Entity B during the Credential Configuration phase.

Security Association Establishment Framework		Parameters specific to the Security Association Establishment Frameworks
Pre-Provisioned M2M Secure Connection Key		None
Certificate Based	Other entity is authenticated using Raw Public Key Certificate	Other entity's identity (IdA or IdB)
		Other entity's Public key identifier
	Other entity is authenticated using Device Certificate	Other entity's identity (IdA or IdB)
		Other entity's globally unique hardware instance identifier
		Other entity's trust anchor certificates
	Other entity is authenticated using CSE-ID Certificate	Other entity's CSE-ID (IdA or IdB)
		Other entity's trust anchor certificates
	Other entity is authenticated using AE-ID Certificate	Other entity's AE-ID (IdA or IdB)
		Other entity's trust anchor certificates
MAF-Based	Configured to Entity A	Entity B's CSE-ID or AE-ID (IdB)
GBA-Based	Configured to Entity A	Entity B's CSE-ID or AE-ID (IdB)

Mechanisms for Association Configuration of an entity shall authenticate the configuration source and provide integrity protection for the configured information communicated from the configuration source to the entity.

9.1.2.2 Association Configuration of M2M Authentication Functions

Table 9.1.2.2-1 lists the parameters configured to M2M Authentication Functions in the Association Configuration phase.

Table 9.1.2.2-1 Parameters configured to a M2M Authentication Functions during the Association Configuration phase.

Security Association Establishment Framework		Parameter
MAF-Based	A-to-MAF Authentication	Entity B's CSE-ID or AE-ID (IdB)

This specification assumes that Association Configuration of the M2M Authentication Functions will utilize business logic of the Stakeholder that operates the M2M Authentication Function, and the details are not described in this specification.

9.1.2.3 Association Configuration of UNSP Authentication Servers

Table 9.1.2.3-1 lists the parameters configured to an Underlying Network Service Provider authentication server (e.g. HLR, HSS or AAA) in the Association Configuration phase.

Table 9.1.2.3-1 Parameters configured to an Underlying Network Service Provider authentication server (e.g. HLR, HSS or AAA) during the Association Configuration phase.

Security Association Establishment Framework	Parameter
GBA-Based. Parameters are configured to UE's GBA User Security Settings (GUSS). GUSS details are specified in 3GPP 33.220 [13]. Where this description differs from [13], [13] takes precedence	Entity A's CSE-ID or AE-ID (IdA)
	Entity B's CSE-ID or AE-ID (IdB)

The Association Configuration of the Underlying Network Service Provider authentication server is achieved by updating the GBA User Security Settings (GUSS) (3GPP 33.220 [13]) of the User Equipment (UE) upon which the Enrolee is executed. This specification assumes that this Association Configuration will utilize business logic of the Underlying Network Service Provider, and the details are not described in this specification.

9. 2 Remote Security Provisioning Framework Procedures and Parameters

9.2.1 Bootstrap Credential Configuration Procedures and Parameters

The following Bootstrap Credential Configuration procedures are described in this clause:

- Bootstrap Credential Configuration of Enrolees and Enrolment Targets (except for the GBA-Based case as discussed below), see Clause 9.2.1.1,
- Bootstrap Credential Configuration of M2M Enrolment Functions (except for the GBA-Based case as discussed above), see Clause 9.2.1.2.

The following Bootstrap Credential Configuration procedures are specified by other organizations:

- Bootstrap Credential Configuration of Underlying Network Service Provider authentication servers (e.g. HLR, HSS or AAA) for the GBA-Based Security Association Establishment Framework. These details are specified by 3GPP TS 33.220 [13], 3GPP2 S.S0109-A [14].

- Bootstrap Credential Configuration of Enrolees for the GBA-Based Security Association Establishment Framework. These details are specified by 3GPP TS 33.220 [13], 3GPP2 S.S0109-A [14].

9.2.1.1 Bootstrap Credential Configuration of Enrolee and Enrolment Targets

Table 9.2.1.1-1 lists the parameters configured to Enrolees and Enrolment Targets in the Bootstrap Credential Configuration phase for authentication with the M2M Enrolment Function in the Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework and Certificate-Based Remote Security Provisioning Framework.

Table 9.2.1.1-1 Parameters configured to Enrolees and Enrolment Targets during the Bootstrap Credential Configuration phase.

Remote Security Provisioning Framework		Parameter
Pre-Provisioned M2M Secure Connection Key authentication. Not applicable to MAF.		Kpm
		KpmId
		MEF URI
Certificate-Based authentication	Entity authenticates itself using a raw public key	Entity's Private Key
		Entity's Raw Public Key Certificate
	Entity authenticates itself using a device certificate	Entity's Private Key
		Entity's Certificate and Chain
	Entity authenticates itself using a CSE-ID or AE-ID certificate	Entity's Private Key
		Entity's Certificate and Chain

The Bootstrap Credential Configuration of an Enrolee or Field Domain Enrolment Target for the Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework and Certificate-Based Remote Security Provisioning Framework shall authenticate the configuration source and shall provide confidentiality and integrity protection of the configured information communicated from the configuration source to the secured environment of the Enrolee or Field Domain Enrolment Target. The present document does not specify any such mechanisms.

The Bootstrap Credential Configuration of an Infrastructure Domain Enrolment Target (including an M2M Authentication Functions) expected to use business logic of the Stakeholder operating the Infrastructure Domain Enrolment, and the details are not described in this specification.

9.2.1.2 Bootstrap Credential Configuration of M2M Enrolment Functions

It is assumed that an M2M Enrolment Function already knows its FQDN.

Table 9.2.1.2-1 lists the parameters configured to M2M Enrolment Functions in the Bootstrap Credential Configuration phase for mutual authentication with Enrolees and Enrolment Targets using the Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework and Certificate-Based Remote Security Provisioning Framework.

Table 9.2.1.2-1 Parameters configured to the M2M Enrolment Function during the Bootstrap Credential Configuration phase for mutual authentication with Enrolees and Enrolment Targets using the Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework and Certificate-Based Remote Security Provisioning Framework.

Remote Security Provisioning Framework	Parameters specific to the Remote Security Provisioning Frameworks
Pre-Provisioned Symmetric Enrolment Key authentication of Enrolee or Enrolment Target	Kpm
	KpmId
Certificate Based authentication of Enrolee or Enrolment Target	MEF PrivateKey
	MEF Certificate and Chain

The Bootstrap Credential Configuration of M2M Enrolment Functions is expected to use business logic of the stakeholder operating the M2M Enrolment Function, and the details are not described in this specification.

9.2.2 Bootstrap Instruction Configuration Procedures and Parameters

The following Bootstrap Instruction Configuration procedures are described in this clause:

- Bootstrap Instruction Configuration of Enrolees, see Clause 9.2.2.1,
- Bootstrap Instruction Configuration of Enrolment Targets, see Clause 9.2.2.2,
- Bootstrap Instruction Configuration of M2M Enrolment Functions, see Clause 9.2.2.3
- Bootstrap Instruction Configuration of Underlying Network Service Provider authentication servers (e.g. HLR, HSS or AAA), see Clause 9.2.2.4.

9.2.2.1 Bootstrap Instruction Configuration of Enrolees

Table 9.2.2.1-1 lists the parameters configured to an Enrolee during the Bootstrap Instruction Configuration phase which are common to all Remote Security Provisioning Frameworks.

Table 9.2.2.1-1 Parameters configured to an Enrolee during the Bootstrap Instruction Configuration phase of which are common to all Remote Security Provisioning Frameworks.

Parameter common to all Remote Security Provisioning Frameworks
Enrolment Target Identifier (Enrolee B's AE-ID or CSE-ID, or MAF-ID)

Table 9.2.2.1-2 lists the Remote Security Provisioning Framework–specific parameters configured an Enrolee in the Bootstrap Instruction Configuration phase of the Remote Security Provisioning Framework.

Table 9.2.2.1-2 Remote Security Provisioning Framework –specific parameters configured to an Enrolee during the Instruction Configuration phase of the Remote Security Provisioning Framework.

Remote Security Provisioning Framework	Remote Security Provisioning Framework-specific Parameters
Pre-Provisioned Symmetric Enrolment Key	*None*
Certificate Based	MEF URI
	MEF TrustAnchor Certificates
GBA-Based	*None*

Mechanisms for Bootstrap Instruction Configuration of Enrolees shall authenticate the configuration source and shall provide at least integrity protection of the configured information communicated from the configuration source to the Enrolee.

9.2.2.2 Bootstrap Instruction Configuration of Enrolment Targets

Table 9.2.2.2-1 lists the parameters configured to Enrolment Targets during the Bootstrap Instruction Configuration phase.

Table 9.2.2.2-1 Parameters configured to Enrolment Targets during the Bootstrap Instruction Configuration phase.

Mechanism for authenticating the Enrolment Target	Remote Security Provisioning Framework-specific Parameters
Pre-Provisioned Symmetric Enrolment Key. *Not applicable to MAF.*	*None*
Certificate Based	MEF/BSF Trust Anchor Certificates

Mechanisms for Bootstrap Instruction Configuration of Enrolment Targets shall authenticate the configuration source and shall provide at least integrity protection of the configured information communicated from the configuration source to the Enrolment Targets.

This specification assumes that Bootstrap Instruction Configuration of Infrastructure Domain Enrolment Targets (including M2M Authentication Functions) utilizes business logic of the Stakeholder that operates the M2M Authentication Function, and the details are not described in this specification.

9.2.2.3 Bootstrap Instruction Configuration of M2M Enrolment Functions

Table 9.2.2.3-1 lists the parameters configured to an M2M Enrolment Function during the Bootstrap Instruction Configuration phase which are common to the Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework and Certificate-Based Remote Security Provisioning Framework.

Table 9.2.2.3-1 Parameters configured to M2M Enrolment Functions during the Bootstrap Instruction Configuration phase of which are common to the Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework and Certificate-Based Remote Security Provisioning Framework.

Parameter common to all Remote Security Provisioning Frameworks
Enrolee's assigned CSE-ID or AE -ID
Enrolment Target Identity (Enrolee B's CSE-ID or AE-ID, or MAF-ID)

Table 9.2.2.3-2 lists the Remote Security Provisioning Framework–specific parameters configured to an M2M Enrolment Functions in the Bootstrap Instruction Configuration phase of the Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework and Certificate-Based Remote Security Provisioning Framework.

Table 9.2.2.3-2 Remote Security Provisioning Framework–specific parameters configured to an M2M Enrolment Function during the Instruction Instruction Configuration phase of the Pre-Provisioned Symmetric Enrolee Key Remote Security Provisioning Framework and Certificate-Based Remote Security Provisioning Framework.

Remote Security Provisioning Framework		Remote Security Provisioning Framework-specific Parameters
Pre-Provisioned Symmetric Enrolment Key		*None*
Certificate Based	Enrolee is authenticated using a raw public key certificate	Enrolee's Public key identifier
	Enrolee is authenticated using a device certificate	Enrolee's M2M Device ID
		Enrolee's Trust Anchor Certificates
	Enrolee is authenticated using a CSE-ID or AE-ID certificate	Enrolee's Trust Anchor Certificates

Table 9.2.2.3-3 lists the parameters configured to an M2M Enrolment Functions for authentication of the Enrolment Target when the Enrolment Target is Enrolee B (a CSE or AE).

Table 9.2.2.3-3 Remote Security Provisioning Framework–specific parameters configured to an M2M Enrolment Function during the Bootstrap Instruction Configuration phase for authentication of the Enrolment Target when the Enrolment Target is Enrolee B (a CSE or AE)..

Mechanism for Authenticating Enrolee B		Parameters specific to the mechanism for authenticating Enrolee B
Pre-Provisioned Symmetric Enrolment Key		*None*
Certificate Based	Enrolee B is authenticated using a raw public key certificate	Enrolee B's Public key identifier
	Enrolee B is authenticated using a device certificate	Enrolee B's M2M Device ID
		Enrolee B's Trust Anchor Certificates
	Enrolee B is authenticated using a CSE-ID or AE-ID certificate	Enrolee B's Trust Anchor Certificates

Table 9.2.2.2-4 lists the parameters configured to an M2M Enrolment Function for authenticating the Enrolment Target when the Enrolment Target is an M2M Authentication Function.

Table 9.2.2.2-4 The parameters configured to an M2M Enrolment Function for identifying and authenticating the Enrolment Target when the Enrolment Target is an M2M Authentication Function.

Parameter common to all Remote Provisioning Frameworks
MAF Trust Anchor Certificates

This specification assumes that Bootstrap Instruction Configuration of the M2M Enrolment Functions utilizes business logic of the Stakeholder that operates the M2M Enrolment Function, and the details are not described in this specification.

9.2.2.4 Bootstrap Instruction Configuration of UNSP Authentication Server

Table 9.2.2.4-1 lists the parameters configured to an Underlying Network Service Provider authentication server (e.g. HLR, HSS or AAA) during the Bootstrap Instruction Configuration phase of the GBA-Based Remote Security Provisioning Framework.

Table 9.2.2.4-1 Parameters configured to M2M Enrolment Functions during the Bootstrap Instruction Configuration phase of the GBA-Based Remote Security Provisioning Framework.

Parameter	Mandatory /Optional for all Remote Security Provisioning Frameworks
Enrolee's assigned CSE-ID or AE-ID	Mandatory
Enrolment Target Identifier (Enrolee B's CSE-ID or AE-ID, or MAF-ID)	Mandatory

The Bootstrap Instruction Configuration of the Underlying Network Service Provider authentication server is achieved by updating the GBA User Security Settings (GUSS) (3GPP TS 33.220 [13]) of the User Equipment (UE) upon which the Enrolee is executed. This specification assumes that this Bootstrap Instruction Configuration utilizes business logic of the Underlying Network Service Provider, and the details are not described in this specification.

10 Protocol and Algorithm Details

10.1 Certificate-Based Security Framework Details

10.1.1 Certificate Profiles

NOTE: These certificate profiles are compliant with the CoAP specification RFC 7252 [38].

10.1.1.1 Common Certificate Details

All certificates shall conform to the following profile:

- Certificates shall conform to RFC 5280 [34].
- The certificate shall include a SubjectPublicKeyInfo that indicates an algorithm of id-ecPublicKey with namedCurves secp256r1 [34]; this curve is equivalent to the NIST P-256 curve [39].
- The public key format shall be uncompressed [46].
- The hash algorithm shall be SHA-256.
- The key usage extension shall be included and shall indicate at least digitalSignature.

10.1.1.2 Raw Public Key Certificate Profile

Raw public key certificates shall conform to clause 10.1.1.1 "Common Certificate Details" and RFC 7250 [37].

10.1.1.3 Details Common to Certificates with Certificate Chains

Certificates with Certificate Chains shall conform to the following description:

- These certificates shall conform to clause 10.1.1.1 “Common Certificate Details”.
- Certificates shall be signed with ECDSA using secp256r1, and the signature shall use SHA-256.
- Certificate chains should limit the number of intermediate CA certificates to avoid having a negative impact in constrained environments.

10.1.1.4 Profile for Device Certificates and their Certificate Chains

10.1.1.4.1 Profile for Device Certificates

Device certificates shall conform to the following description:

- Device certificates shall conform to clause 10.1.1.3 “Details Common to the Certificates with Certificate Chains”.
- The subjectAltName extension of device certificates shall include one or more globally unique hardware instance identifiers.

Example. Annex H “Object Identifier Based M2M Device Identifier” TS-0001 [1] defines an object identifier -based M2M Device ID that can be used for providing a one or more globally unique hardware instance identifier. An object identifier -based M2M Device ID can be representing in an otherName field in the subjectAltName extension, where:

- otherName “type-ID” component is set to the M2M Device Indication ID (Annex H.2.1 “M2M Device Indication ID” TS-0001 [1]) arc of the object identifier M2M Device ID, and
- The otherName “value” component is set to the remainder of the object identifier M2M Device ID: Manufacturer ID arc, Model ID arc, Serial Number ID arc and optional Expanded ID arc (see Annex H.2 “OID Based M2M Device Identifier” TS-0001 [1])

NOTE: Providing the Model ID as part of the M2M Device ID can have privacy implications in some scenarios.

10.1.1.4.2 Profile for Certificate Authority Certificates for Device Certificates

Certificate Authority Certificates in the certificate chain for a device certificate shall conform to the following description:

- These certificates shall conform to clause 10.1.1.3 “Details Common to the Certificates with Certificate Chains”.
- Certificate Authority Certificates for device certificates are recommended to use the name constraints extension (see clause 4.2.1.10 “Name Constraints” of RFC 5280 [34]) to constrain the globally unique hardware instance identifiers in subsequent device certificates in a certification path.

Example. Name constraints are defined in terms of permitted or excluded name subtrees. Subtrees of an object identifier based M2M Device ID name space are represented by an otherName field with

- “type-ID” set to the M2M Device Indication ID (Annex H.2.1 “M2M Device Indication ID” TS-0001 [1]) arc of the applicable object identifier M2M Device ID name space, and
- “value” set to set to the remainder of the object identifier identifying the subtree.

10.1.1.5 Profile for CSE-ID Certificates, AE-ID Certificates and their Certificate Chains

CSE-ID certificates and AE-ID certificates and all other certificates in the corresponding certificate chain shall conform to clause 10.1.1.3 “Details Common to Certificates with Certificate Chains”.

The full URI representation of the CSE-ID or AE-ID shall be included in the subjectAltName extension.

The certificate used to sign the CSE-ID certificates and AE-ID certificate shall include nameConstraints satisfied by the hostname part of the full URI representation of the CSE-ID or AE-ID.

CSE-ID certificates and AE-ID certificates shall not include wildcards.

10.1.1.6 Profile for FQDN Certificates and their Certificate Chains

FQDN Certificates and all other certificates in the corresponding certificate chain shall conform to clause 10.1.1.3 "Details Common to Certificates with Certificate Chains".

An FQDN Certificate shall include the FQDN of the subject M2M Enrolment Function or M2M Authentication Function in the subjectAltName extension.

FQDN Certificates shall not include wildcards.

10.1.2 Public Key Identifiers

The public key identifier for a raw public key certificate shall calculated as described in Section 2 of RFC 6920 [40] using the SHA-256 hash algorithm. The public key identifier shall be generated using one of the sha-256-120, sha-256-128 or sha-256 hash algorithms specified in RFC 6920 [40].

It is recommended that the public key identifier be as long as practical within the deployment constraints.

The trusted public key identifier (received during Association Configuration or Bootstrap Instruction Configuration) is matched against the raw public key certificate (received during the Security Handshake) using the following procedure:

1. A check digest value is computed according to Section 2 of RFC 6920 [40] using the hash algorithm identified in the trusted public key identifier.

2. The check digest value is compared against the digest value encoded in the trusted public key identifier. If the values are identical then the raw public key certificate matches the trusted public key identifier. Otherwise, the raw public key certificate does not match the trusted public key identifier.

10.1.3 Support Requirements for each Public Key Certificate Flavour

Table 10.1.3 lists, for each of the various types of entity (Field Domain CSE, Field Domain AE, IN-CSE, IN-AE, M2M Authentication Function and M2M Enrolment Function), the flavour of certificate that may be issued to the entity and the flavour of other entity's certificates that the entity is required to be able to process. . In this table "O" indicates optional, "M" indicates Mandatory, "CA" indicates that the option is required if the entity supporting the certificate-based security association establishment framework, "CB" indicates conditional on the entity supporting certificate-based Remote Security Provisioning framework.

Table 10.1.3-1 Applicability of certificate flavours issued to an entity and flavours of other entity's certificates that the entity is required to be able to process.

Entity	Flavour of certificate may be issued to entity					Flavour of other entity's certificates that the entity is recommended to be able to process.				
	Raw	Device	CSE-ID	AE-ID	FQDN	Raw	Device	CSE-ID	AE-ID	FQDN
Field Domain CSE	O	O	O	-	-	CA	CA	CA	CA	CB
Field Domain AE	O	O	-	O	-	CA	CA	CA	-	CB
IN-CSE	O	-	O	-	-	CA	CA	CA	CA	-
IN-AE	O	-	-	O	-	CA	-	CA	-	-
MAF	-	-	-	-	M	-	-	-	-	M
MEF	-	-	-	-	M	CB	CB	-	-	M

Mutual authentication between remote management servers and remote management clients is not considered in the present document.

10.2 TLS and DTLS Details

10.2.1 TLS and DTLS Versions

Where TCP payloads are to be secured, TLS v1.2 [16] shall be used.

Where UDP payloads are to be secured, DTLS v1.2 [17] shall be used, noting that the DTLS v1.2 ciphersuites are identical to the TLS v1.2 ciphersuites.

Implementations shall support the Server Name Indication (SNI) to indicate their authority in the SNI HostName field as defined in Section 3 of RFC 6066 [44]. This is needed so that when a host that acts as a virtual server for multiple Authorities receives a new TLS or DTLS connection, it knows which keys to use for the TLS or DTLS session.

10.2.2 TLS and DTLS Ciphersuites for TLS-PSK-Based Security Frameworks

The following Security Association Establishment Frameworks:

- Pre-Provisioned Symmetric Key Security Association Establishment Framework;
- MAF-Based Security Association Establishment Framework;
- GBA-Based Security Association Establishment Framework;
- Pre-Shared Key Remote Security Provisioning Framework;
- GBA-Based Remote Security Provisioning Framework;

shall use one of the key exchange algorithms defined in RFC 4279 [15]:

TLS implementations supporting these security frameworks shall implement at least the following TLS ciphersuite:

- TLS_PSK_WITH_AES_128_CBC_SHA256 (RFC 5487 [42]).

DTLS implementations supporting these security frameworks shall implement at least the following ciphersuites

- TLS_PSK_WITH_AES_128_CCM_8 (RFC 6655) [31].

The security considerations of Section 7 of RFC4279 [15] apply. In particular, applications should carefully weigh whether or not they need Perfect Forward Secrecy (PFS) and select an appropriate ciphersuite (Section 7.1 of RFC4279 [15]).

10.2.3 TLS and DTLS Ciphersuites for Certificate-Based Security Frameworks

The following Security Frameworks:

- Certificate-Based Security Association Establishment Framework:
- Certificate-Based Security Bootstrap Framework;

shall use the standard TLS handshake (RFC 5246 [16]) with the ECDHE_ECDSA Key Exchange (RFC4492 [43]).

TLS implementations supporting these security frameworks shall implement at least the following ciphersuite:

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, (RFC 5289) [32].

DTLS implementations supporting these security frameworks shall implement at least the following TLS ciphersuite:

- TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8, RFC 7251 [45]

Implementations supporting these security frameworks shall support authenticating other entities using all available public key certificate flavours (see clause 8.1.1.2.1 “Public Key Certificate Flavours”)

- Raw public key certificate: using the mechanism specified in RFC 7250 [37], Implementation shall support receiving and processing raw public keys compliant with Section 9.1.3.2 “Raw Public Key Certificates” in RFC 7252 [38].

- All other certificates: X.509 certificates including device hardware identifier. Implementation shall support receiving and processing raw public keys compliant with Section 9.1.3.3 “X.509 Certificates” in RFC 7252 [38].

10.3 Direct Security Bootstrap Framework Algorithm Details

10.3.1 TLS Key Export Details

Following successful TLS authentication between the Enrolee and M2M Enrolment Function, the Enrolment Key (Ke) and RelativeKeId are generated from the (D)TLS session secrets by the Enrolee and M2M Enrolment Function by applying TLS Key Export (RFC 5705) [18] using the label “EXPORTER-oneM2M-Bootstrap” and length 48. The Enrolment Key (Ke) is set to the value of the 32 least significant bytes, while RelativeKeId is set to the value of the 16 most significant bytes.

10.3.2 Derivation of Master Credential from Enrolment Key

This clause describes the details when generating a Master Credential (Km) from an Enrolment Key (Ke) in Security Bootstrap Frameworks.

The following information shall be used when generating Km from Ke:

- The value of the Enrolment Key (Ke);

- The M2M Authentication Function Identifier (MAF-ID) shall be encoded to an octet string according to UTF-8 encoding rules as specified in IETF RFC 3629 [19] and apply Normalization Form KC (NFKC) as specified in [20].

The value of Km shall be generated as

Km := HMAC-SHA-256(Ke, “oneM2M Enrolment Key to Master Credential derivation” || MAF-ID),

where HMAC-SHA-256 is defined in RFC 2014 [33].

10.3.3 Derivation of Provisioned Secure Connection Key from Enrolment Key

This clause describes the details when generating a Provisioned Secure Connection Key (Kpsa) from an Enrolment Key (Ke) in Remote Provisioning Frameworks.

The following information shall be used when generating Kpsa from Ke:

- The value of the Enrolment Key (Ke);

- Enrolee B’s CSE-ID or AE-ID (Enrolee-B-ID), which shall be encoded to an octet string according to UTF-8 encoding rules as specified in IETF RFC 3629 [19] and apply Normalization Form KC (NFKC) as specified in [20].

The value of Kpsa shall be generated as

Kpsa := HMAC-SHA-256(Ke, “oneM2M Enrolment Key to Provisioned Secure Connection Key derivation” || Enrolee-B-ID),

where HMAC-SHA-256 is defined in RFC 2014 [33].

10.3.4 Generating KeId

The KeId value shall be formed as

KeId = base64encode(RelativeKeId)@MEF_FQDN,

where

- base64encode(RelativeKeId) denotes the base64 encoding (RFC 3548 [41]) of the value of RelativeKeId, and
- MEF_FQDN denotes the FQDN of the M2M Enrolment Function.

Annex A (informative): Mapping of 3GPP GBA terminology

Table A.1 provides a mapping of terminology and abbreviations used in GBA according to 3GPP specification [13] to corresponding oneM2M terminology and abbreviations as used within the present document.

Table A.1

GBA entities, keys and processes	oneM2M Security Bootstrap entities, keys & processes
UE	Enrolee
BSF	MEF
NAF	MAF
Bootstrapping Procedure	**Bootstrap Security Handshake + Temporary Enrolment Key Generation**
Ks	Ke
B-TID	KeId
Bootstrapping Usage Procedure	**Usage in Centralized Key Distribution Server Handshake**
NAF FQDN	IdMAF
Ks_(ext/int)_NAF	Km (Master Credential)

Annex B (informative): General Mutual Authentication Mechanism

oneM2M mutual authentication schemes allow oneM2M entities to prove that they know related credentials such as Master Credentials, without having to exchange value of those credentials, and sensitive data such as security identities and security identifiers. To prevent reading and copying of credentials, a secure environment within the Security CSF provides protection against tampering of those credentials and related processed information.

A general mutual authentication protocol is applied to both symmetric and asymmetric key based schemes. Precise protocol messages and parameters depend on the chosen scheme and the security parameters selected. Typically it consists of following steps as shown in figure B.1.

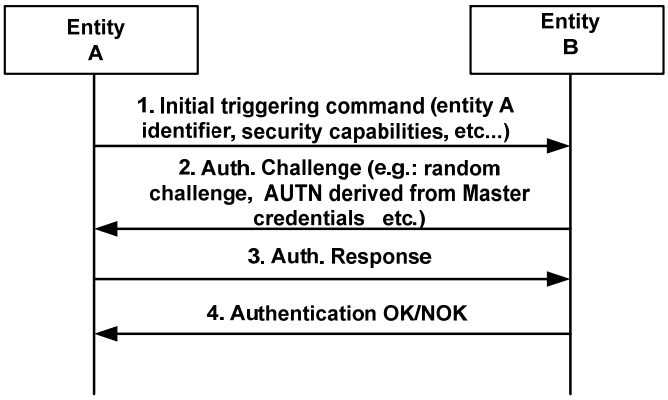


Figure B.1: Mutual Authentication

1. An initial step where an entity A is securely identified to an entity B with whom previous or no previous contact has been made. In this step entity A identifies itself to an entity B protected against eavesdropping, i.e. no exchange of key materials (Master Credentials).

2. In the second step entity B sends a challenge to entity A. The Authentication Challenge consists of a challenge, the authentication token (AUTN) of entity B derived from Master Credentials, etc. The authentication challenge, which may be random or not, depends on the chosen authentication scheme and the security parameters selected for symmetric and asymmetric key based schemes.

3. Entity A replies with an Authentication Response that contains an authentication token (AUTN) derived from its known Master Credentials and the received Authentication Challenge. This Authentication Response is sent if entity B has been successfully authenticated by entity A.

4. Entity B then verifies the relation between entity A's identity and the response received in step 3,. If the verification is positive, entity B is assured that the response has been created by entity A using a secret associated with entity A’s identity provided in step 1.

B.1. Group Authentication

The oneM2M transactions may naturally involve groups of M2M entities rather than individual ones. A number of entities are classified as a group due to their proximate locations, having the same features, belonging to the same owner, or any other reasons [i.10]. To get services, all entities in such a group should be authenticated first. The traditional authentication mechanism has two main solutions, the first authentication mechanism is that the service provider authenticates each entity in the group one by one; the second authentication mechanism is that each entity makes mutual authentication with a group agent, then the group agent makes mutual authentication with the service provider. If the first authentication mechanism is used, the resulting authentication overheads of computation and communication may be too high to afford. If the second authentication mechanism is used, it has the following security weaknesses:

a) It may exist the man-in-the-middle attack by the group agent: The group agent would be placed in unsecure place or owned by different provider rather than the service provider. If the group agent is compromised or lie to service provider, group agent would act as a middle attacker to make fake authentication to entities and report fake identity to service provider since there is no direct authentication from service provider to each M2M entity.

b) Privacy concern: All information from M2M entities is transferred through the group agent, and the group agent knows all information generated by each entity. Based on security consideration, if the group agent is owned by different owner other than the entities' and service providers' owner, the group agent should not get the message.

Hence, the M2M entities (e.g., ASN or ADN) with the same feature can utilize group authentication to service provider (e.g., infrastructure node) in order to provide end-to-end secure tunnel as well as reducing the communication overhead.

Annex C (informative): Security protocols associated to specific SE technologies

The Secure Environment supporting security functions specified by oneM2M provides a level and a type of protection (e .g. integrity protection, confidentiality, tamper resistance) to the information it contains, independently of the method of protection (e.g. UICC, embedded security element, TEE, etc..). Administration of their content is implementation dependent and relies on existing standards within specific Secure Environment technologies. Some of them are listed below for information:

C.1 UICC

In case of UICC (SE compliant with ETSI TS 102 671 [23]), OTA mechanisms as specified in [7] and [8], and its extensions [9], [10] for 3GPP underlying networks or [11] and [12] for 3GPP2 underlying networks are used to securely administrate the sensitive data of the M2M Service Layer. UICC provides the highest protection level 3 against attacks according the Classification of Protection levels Table 6.2.1-1 in clause 6.2.1.

C.2 Other secure element and embedded secure element with ISO 7816 interface

In case the Secure Environment is implemented as a security element or as an embedded security element supporting an ISO/IEC 7816 interface [26], example of remote administration can be according to GlobalPlatform Remote Administration [a]. An embedded secure element provides the highest protection level 3 against attacks according the Classification of Protection levels Table 6.2.1-1 in clause 6.2.1.

C.3 Trusted Execution Environment

In case the secure environment is implemented as a Trusted Execution Environment (TEE) according to GlobalPlatform [b], remote administration is provided according to GlobalPlatform Remote Administration [21]. TEE provides the medium protection level 2 against attacks according the Classification of Protection levels Table 6.2.1-1 in clause 6.2.1.

C.4 SE to CSE binding

In case the SE is implemented as an independent security element supporting ETSI TS 102 221 [24], the secure channel specified in ETSI TS 102 484 [25] provides logical binding of the SE to a specific CSE or AE. This also protects the information exhanged between the SE and the associated entity on physically exposed interfaces, and is therefore recommended for devices that are physically exposed to attackers.

Annex D (normative): UICC security framework to support oneM2M Services

This annex is applicable when UICC (a type of Independent Security Element compliant with ETSI TS 102 221 [24] and ETSI TS 102 671 [23]) is involved in M2M service layer security, whether it only serves as a mean to pre-provision M2M Service layer material in M2M Devices/Gateways, or it is further used as Secured Environment in an M2M Device/Gateway.

Specifically, the involvement of UICC in oneM2M security may include any of the following steps:

- Pre-provisioning of initial credentials in M2M nodes by any of the following methods:
  - simple pre-provisioning and administration of M2M Service material (initial credentials and other pre-provisioned parameters), i.e. UICC-based M2M service provisioning;
  - support for infrastructure assisted bootstrapping of the M2M symmetric credentials by derivation from symmetric Access Network credentials stored in the UICC, using GBA.
- Derivation of a security association key directly derived from symmetric Access Network Credentials, using GBA. Note that this process can be supported by a Network Access Application on the UICC independently of the presence of the information structure specified in the present annex.

The support of UICC provisioning of M2M service subscription information shall be indicated in the M2M Service Table for the corresponding M2M Service Subscription as specified in the present annex.

The support of key derivation using GBA that may be used for bootstrapping or security association shall always be indicated in the Service Table of the UICC application of the Access Network Operator supporting the GBA infrastructure.

At the most basic level, UICC-based M2M pre-provisioning requires an interoperable framework to store and administrate related information in the UICC. Further involvement requires a framework for discovery of available services offered by the UICC for the hosting M2M field node. The purpose of the present annex is to specify this framework, which enables both initial service provisioning and remote security administration of the subscription information during the subscription lifetime.

A common scenario is where an M2M field node holds a UICC application protecting Access Network security credentials, and these credentials are used to derive M2M Service Layer security credentials used for M2M service bootstrapping or security association establishment in the service layer . As these scenarios require a trust agreement between the involved Access Network operator and M2M Service Provider, UICC support for M2M services in such situation shall be handled within the context of the associated Network Access application on the UICC. In particular, the UICC support for M2M credentials derivation using GBA shall be indicated within the UICC application of the Access Network operator. This is specified in clause D.1.

Even when the M2M Service Layer credentials are not derived from Access Network Credentials, the UICC may be used as a secure environment that securely protects the symmetric or asymmetric credential used to root security in an M2M field node. In such cases, the M2M subscription information and related methods constitute an independent application that resides on a UICC, in the sense of ETSI TS 102 221 [24]. In particular, ETSI TS 102 221 [24] specifies the application independent properties of the UICC/terminal interface such as the physical characteristics and the logical structure.

NOTE: A terminal in the sense of TS ETSI 102 221 [24] is the part of the M2M field node that holds the UICC, e.g. a communication modem or an M2M Node processing environment.

The specific properties of the M2M Service Provider Identity Module application holding symmetric credentials is specified in clause D.2.

The storage of M2M information elements in the UICC and the procedures used for communication between the hosting M2M field node and the UICC shall be as specified in the present annex. The present annex uses abbreviations and coding conventions defined in ETSI TS 102 221 [24].

D.1 Access Network UICC-based oneM2M Service Framework

D.1.1 Access Network UICC-based oneM2M Service Framework characteristics

An Access Network UICC-based oneM2M Service Framework is always associated with a single M2M Service Subscription and consists of a single DF, DF_{1M2M}, complying with the specifications in D.1.3, implemented in the ADF of a Network Access Application on the UICC. This situation addresses the case where a trust relationship has been established between the M2M SP and the AN operator owning the hosting ADF.

NOTE 1: This does not necessarily imply that the Access Network credentials of the corresponding ADF are used to derive the M2M Service Layer Credentials: e.g. an Access Network operator may refuse derivation from Access Network credentials to an M2M Service Provider, but may still accept to provide space on its UICC to pre-provision independent credentials or support service infrastructure-assisted bootstrapping.

There may be several oneM2M service frameworks (DF_{1M2M}) within the ADF of a single Access Network subscription, in case this Access Network subscription is used by several independent M2M Service subscriptions. The file IDs of the DF_{1M2M} in any ADF shall be listed under the corresponding entry in EF_{DIR} as specified in D.1.2.

NOTE 2: A single M2M service layer subscription can also use multiple access networks: such subscriptions are best provisioned in a dedicated ADF as specified in clause D.2.

The content of any DF_{1M2M} in an Access Network application ADF shall be as specified in clause D.1.3.

D.1.2 M2M Service Framework discovery for Access Network UICC

When a UICC Network Access application supports one or more M2M Service subscription, with a DF_{1M2M}, the EF_{DIR} entry corresponding to this UICC Network Access Application shall contain the following M2M related Data Objects:

- oneM2M Service Framework DO: defining the association between the identifier of one M2M Service Subscription provisioned in the ADF and the related DF corresponding to this M2M subscription. Likewise, each M2M Service Subscription is associated to one DF. Each of these DFs is hereafter referred as DF_{1M2M}.

There shall be as many oneM2M Service Framework Data Objects as there are M2M Service Subscriptions provisioned in the ADF.

Table J.1: Coding of oneM2M related DOs

Bytes	Length	Description	Status
1	1	Discretionary template tag = '73'	M
2	1	Length of the discretionary template = X	M
3 to (2+X)	X	Discretionary Template	X

Table J.2: Coding of oneM2M Discretionary Template related DOs

Bytes	Length	Description	Status
1	1	oneM2M service specific data content tag = 'A2'	M
2	1	M2M service specific data content length = Y	M
3 to (2+Y)	Y	M2M service specific data content	M

Table J.3: Coding of oneM2M Service Specific Data Content related DOs

Bytes	Length	Description	Status
1	1	oneM2M supported service provisioning tag = '80'	M
2	1	Length of the M2M supported service provisioning tag = A	M
3 to 4	2	M2M Dedicated File Identifier for following M2M service subscription	M
5 to (A+2)	(A-2)	M2M Subscription Identifier	M

Coding:

- M2M Dedicated File identifier:
  - Contain the file identifier of the DF_{1M2M} associated to the provisioning of the M2M Service subscription identified in the DO.
- M2M Subscription Identifier:
  - The identifier of the M2M service subscription provisioned in the DF_{1M2M} indicated in the Data Object, encoded in binary format.

D.1.3 Content of files at the DF_{1M2M} level

This clause specifies the EFs for the M2M service provisioning specific to a single M2M service provider, defining access conditions, data items and coding. A data item is a part of an EF which represents a complete logical entity.

The file structure for DF_{1M2M} is illustrated in figure D.1:

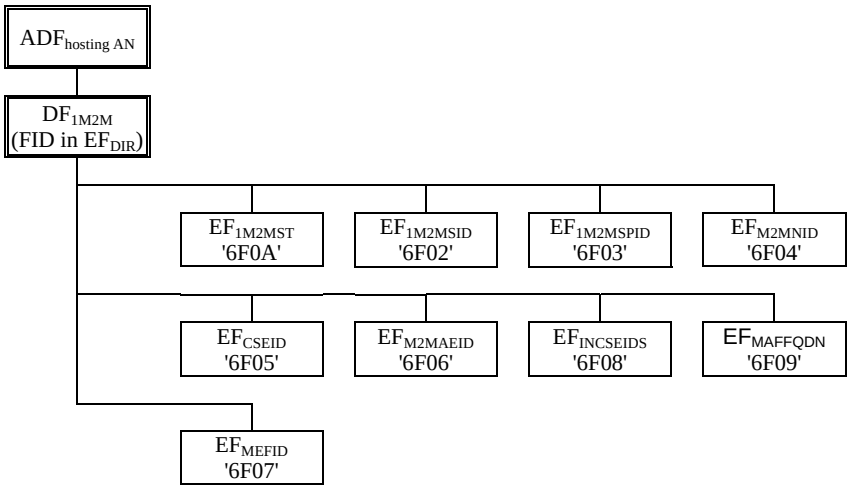


Figure D.1: File identifiers and directory structures of DF_{1M2M} in an hosting Access Network application ADF

D.1.3.1 EF_{1M2MST} (oneM2M Service Table)

This EF indicates which optional oneM2M services are available for the corresponding subscription. If a service is not indicated as available in the oneM2M DF, the hosting M2M field node shall not select this service. The presence of this file is mandatory if optional services are provided by the subscription.

Identifier: '6F0A'	Structure: transparent	Mandatory
SFI: '0A'		
File size: X bytes, X >= 1	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	ADM
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1	Services n°1 to n°8	M	1 byte
2	Services n°9 to n°16	O	1 byte
3	Services n°17 to n°24	O	1 byte
4	Services n°25 to n°32	O	1 byte
etc.			
X	Services n°(8X-7) to n°(8X)	O	1 byte

-Services

Contents:		
	Service n°1:	Local CSE-ID provisioning
	Service n°2	IN-CSE-ID list provisioning
	Service n°3	MAF FQDN provisioning
	Service n°4	Local M2M AE-ID list provisioning
	Service n°5	Bootstrapping: MEF address provisioning
	Service n°6	M2M-Node-ID information
	Service n°7	GBA Secure Provisioning (see Note)
	Service n°8	GBA Secure Connection (see Note)

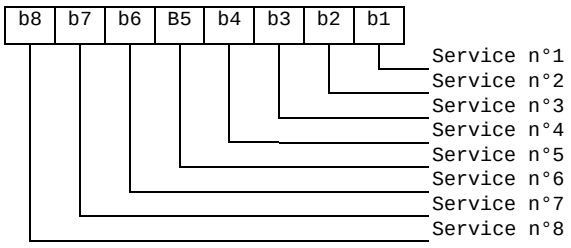
NOTE: Services n°7 and 8 can only be available in a oneM2M Service Table located in a DF_{1M2M} hosted in the ADF of the Network Access Application from which the M2M Service Layer credentials are expected to be derived.

The EF shall contain at least one byte. Further bytes may be included, but if the EF includes an optional byte, then it is mandatory for the EF to also contain all bytes before that byte. Other services are possible in the future and will be coded on further bytes in the EF. Coding:

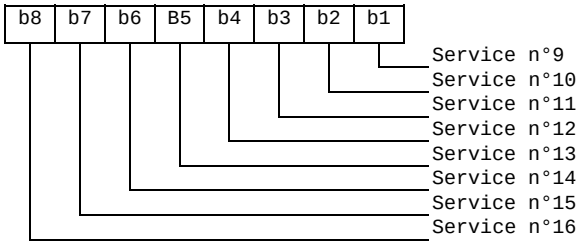
1 bit is used to code each service:
bit = 1: service available;
bit = 0: service not available.

- Service available means that the M2M Service Subscription provisioned in the current DF or ADF has the capability to support the service and that the service is available for the user of the M2M Service Subscription.
  Service not available means that the service shall not be used by the M2M Service Subscription user, even if the M2M Service Subscription has the capability to support the service.

First byte:



Second byte:



etc.

D.1.3.2 $EF_{1M2MSID}$ (oneM2M Subscription Identifier)

This EF contains the oneM2M Subscription Identifier, M2M-Sub-ID. There shall be only one TLV object within this EF.

Identifier: '6F02'	Structure: transparent		Mandatory
SFI: '02'			
File size: X bytes		Update activity: low	
Access Conditions: READ ALW UPDATE ADM DEACTIVATEADM ACTIVATEADM			
Bytes	Description	M/O	Length
1	M2M Subscription Identifier TLV data object	M	X bytes

The M2M Subscription Identifier value field shall contain the M2M-Sub-ID encoded as specified in TS-0004 [4]. The tag value of the oneM2M Subscription Identifier TLV data object shall be '80'.

D.1.3.3 $EF_{1M2MSPID}$ (oneM2M Service Provider Identifier)

This EF contains the oneM2M Service Provider Identifier, M2M-SP-ID, of the M2M Service Provider related to the subscription in $EF_{1M2MSID}$. There shall be only one TLV object within this EF.

Identifier: '6F03'	Structure: transparent		Mandatory
SFI: '03'			
File size: X bytes		Update activity: low	
Access Conditions: READ ALW UPDATE ADM DEACTIVATEADM ACTIVATEADM			
Bytes	Description	M/O	Length
1	M2M-SP-ID TLV data object	M	X bytes

The M2M-SP-ID Value field shall contain the M2M-SP-ID encoded as specified in TS-0004 [TS0004]. The tag value of the M2M-SP-ID TLV data object shall be '80'.

D.1.3.4 EF_{M2MNID} (M2M Node Identifier)

This EF contains the M2M-Node-ID supporting the local CSE. It may be used to logically bind a UICC to a specific M2M Node. If service n°6 is "available", this file shall be present. There shall be only one TLV object within this EF.

Identifier: '6F04'	Structure: transparent		Optional
SFI: '04'			
File size: X bytes		Update activity: low	
Access Conditions: READ ALW UPDATE ADM DEACTIVATEADM ACTIVATEADM			
Bytes	Description	M/O	Length
1 to X	M2M-Node-ID TLV object	M	X bytes

The M2M-Node-ID Value field shall contain the M2M-Node-ID encoded as specified in TS-0004 [4].

D.1.3.5 EF_{CSEID} (local CSE Identifier)

This EF contains the local CSE Identifier, CSE-ID, for the M2M field node associated to the subscription in $EF_{1M2MSID}$. If present, this file is used by the M2M field node to pre-provision the CSE-ID. If service n°1 is "available", this file shall be present. There shall be only one TLV object within this EF.

Identifier: '6F05'	Structure: transparent		Optional
SFI: '05'			
File size: X bytes		Update activity: low	
Access Conditions: READ ALW UPDATE ADM DEACTIVATEADM ACTIVATEADM			
Bytes	Description	M/O	Length
1	CSE-ID TLV data object	M	X bytes

CSE-ID TLV

Contents:

- The CSE-ID Value field shall contain the local CSE-ID formatted as a URI.

Coding:

- The URI shall be encoded to an octet string according to UTF-8 encoding rules as specified in RFC 3629 [19]. The tag value of the URI TLV data object shall be '80'.

D.1.3.6 $EF_{M2MAE-ID}$ (M2M Application Identifiers list)

This EF contains the list of M2M Application Identifiers (AE-IDs) for the local M2M applications supported by the subscription in $EF_{1M2MSID}$. If service n°4 is "available", this file shall be present.

Identifier: '6F06'	Structure: Linear fixed		Optional
SFI: '06'			
Record length: X bytes		Update activity: low	
Access Conditions: READ ALW UPDATE ADM DEACTIVATEADM ACTIVATEADM			
Bytes	Description	M/O	Length
1 to X	M2M AE-ID LV data object	M	X bytes

M2M AE-ID LV

Contents:

- The Value field shall contain the M2M AE-ID formatted as a URI.

Coding:

- TBD

- The URI shall be encoded to an octet string according to UTF-8 encoding rules as specified in RFC 3629 [19].

D.1.3.7 $EF_{INCSEIDS}$ (M2M IN-CSE IDs list)

This EF contains a list of pre-provisioned IN-CSE-ID used to determine the next point of contact after provisioning or M2M Service Bootstrapping. If service n°2 is "available", this file shall be present.

Identifier: '6F08'	Structure: Linear fixed		Optional
Record length: X bytes		Update activity: low	
Access Conditions: READ ALW UPDATE ADM DEACTIVATEADM ACTIVATEADM			
Bytes	Description	M/O	Length
1 to X	IN-CSE-ID LV data object	M	X bytes

IN-CSE-ID LV

Contents:

- The Value field shall contain the IN-CSE-ID formatted as a URI

Coding:

- The URI shall be encoded to an octet string according to UTF-8 encoding rules as specified in RFC 3629 [19]

D.1.3.8 $EF_{MAFFQDN}$ (MAF-FQDN)

This EF is used to pre-provision the FQDN of the MAF to be used for M2M Service Connection after M2M Service Bootstrapping. If service n°3 is "available", this file shall be present. There shall be only one TLV object within this EF.

Identifier: '6F09'	Structure: Transparent	Optional
Length: X bytes	Update activity: low	
Access Conditions: READ ALW UPDATE ADM DEACTIVATEADM ACTIVATEADM		

Bytes	Description	M/O	Length
1	MAF FQDN TLV data object	M	X bytes

MAF FQDN

Contents:

- the FQDN address of the MAF

Coding:

- The MAF-FQDN shall be encoded to an octet string according to UTF-8 encoding rules as specified in RFC 3629 [19]. The tag value of the MAF FQDN TLV data object shall be '80'.

D.1.3.9 EF_{MEFID} (M2M Enrolment Function Identifier)

This EF contains one or more M2M Enrolment Function addresses. The first record in the EF shall be considered to be of the highest priority. The last record in the EF shall be considered to be the lowest priority. If service n°5 is "available", this file shall be present.

Identifier: '6F07'	Structure: linear fixed	Optional
Record length: X bytes	Update activity: low	
Access Conditions: READ ALW UPDATE ADM DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1 to X	MEF Address LV data object	M	X bytes

MEF Address LV data object

Contents:

- Address of MEF, in the format of a FQDN, an IPv4 address, or an IPv6 address.

Coding:

- The format of the data object is as follows:

Field	Length (bytes)
Length	1
Address Type	1
MEF Address	Address Length

- Address Type: Type of the MEF address.
  - This field shall be set to the type of the MEF address according to the following:

Value	Name
0x00	FQDN
0x01	IPv4
0x02	IPv6
All other values are reserved	

- MEF Address: Address of the M2M Service Bootstrap Function.
  - This field shall be set to the address of the M2M Enrolment Function. When the MEF type is set to 0x00, the corresponding MEF Address shall be encoded to an octet string according to UTF-8 encoding rules as specified in RFC 3629 [19].

Unused bytes shall be set to 'FF'.

D.2 oneM2M Service Module application for symmetric credentials on UICC (1M2MSM)

This clause defines the oneM2M Service Module (1M2MSM), an application used for oneM2M Service Layer security functionalities and subscription provisioning based on symmetric credentials. This application resides on the UICC, an IC card specified in ETSI TS 102 221 [24]. In particular, ETSI TS 102 221 [24] specifies the application independent properties of the UICC/terminal interface such as the physical characteristics and the logical structure. There may be several 1M2MSM ADFs on a single UICC, corresponding to independent oneM2M Service Subscriptions.

D.2.1 oneM2M Service Module application file structure

This clause specifies the EFs for the oneM2M service Layer defining access conditions, data items and coding. A data item is a part of an EF which represents a complete logical entity.

D.2.1.1 Content of UICC files at the Master File (MF) level

Files at the UICC MF level are application independent as specified in ETSI TS 102 221 [24]. Only the EF_{DIR} and EF_{ICCID} files are mandatory on UICC for the purpose of 1M2MSM applications. In any case all files shall be as specified in ETSI TS 102 221 [24].

D.2.1.2 Content of files at the 1M2MSM ADF (Application DF) level

The EFs in the 1M2MSM ADF contain oneM2M subscription related information that is required for M2M field nodes operating in an oneM2M environment. This ADF shall be selected using its AID and information in EF_{DIR}. The AID for 1M2MSM applications shall be constructed as specified in ETSI TS 101 220 [27].

NOTE: The ETSI RID can be used for oneM2M pending assignment of a oneM2M dedicated RID in ISO/IEC 7816-5 [i.13].

The File IDs '6F1X' (for EFs), '5F1X' and '5F2X' (for DFs) with X ranging from '0' to 'F' are reserved under the 1M2MSM ADF for administrative use by the card issuer.

The DF_{1M2M} substructure used to isolate the provisioning of network access dependent M2M service related information in a Network Access Application ADF is not needed for access network independent provisioning of an M2M service subscription in a 1M2MSM ADF. Therefore, all the EFs specified in clause D.1.3 shall be present at the 1M2MSM ADF level. The file structure of the ADF_{1M2MSM} is illustrated in figure D.2.

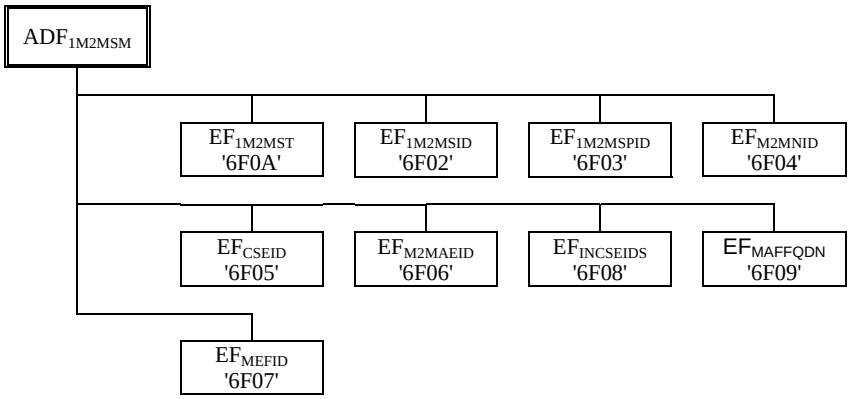


Figure D.2: File identifiers and directory structures of ADF_{1M2MSM}

D.2.2 oneM2M Subscription related procedures for M2M Service

This clause specifies the procedures that shall be executed by M2M field nodes to interact with a oneM2M Service Subscription on UICC. They are applicable independently of the file structure supporting the oneM2M Service Subscription (1M2MSM ADF or DF_{1M2M} under a Network Access Application ADF), unless otherwise indicated.

D.2.2.1 Initialization – 1M2MSM Application selection

This procedure only applies to an M2M subscription supported in a 1M2MSM ADF.

If the M2M field node wants to engage in M2M operation, then after UICC activation (see ETSI TS 102 221 [24]), the M2M field node shall select a 1M2MSM application, if a 1M2MSM application is listed in the EF_{DIR} file, using the SELECT by DF name as defined in ETSI TS 102 221 [24].

After a successful oneM2M application selection, the selected oneM2M AID is stored on the UICC. This application is referred to as the last selected 1M2MSM application. The last selected 1M2MSM application shall be available on the UICC after a deactivation followed by an activation of the UICC.

If a oneM2M application is selected using partial DF name, the partial DF name supplied in the command shall uniquely identify a 1M2MSM application. Furthermore if a 1M2M application is selected using a partial DF name as specified in ETSI TS 102 221 [24] indicating in the SELECT command the last occurrence, the UICC shall select the oneM2M application stored as the last oneM2M application. If, in the SELECT command, the options first, next/previous are indicated, they have no meaning if an application has not been previously selected in the same session and shall return an appropriate error code.

D.2.2.2 1M2MSM session termination

This procedure only applies to a oneM2M subscription supported in a 1M2MSM ADF.
The oneM2M UICC session is terminated by the M2M field node as follows:

- The M2M field node shall indicate to the oneM2M UICC application that the termination procedure is starting, by sending a particular STATUS command.
- Finally, the M2M field node deletes all the M2M subscription related information elements from its memory.
- To actually terminate the session, the M2M field node shall then use one of the mechanisms described in ETSI TS 102 221 [24].

D.2.2.3 oneM2M Service discovery procedure

This procedure is used to discover the oneM2M related services offered by a oneM2M UICC.

The M2M field node shall perform the reading procedure with EF_{1M2MST}. If no oneM2M related service is indicated as available, the M2M field node shall assume that only the provisioning of mandatory parameters is available in this ADF.

D.2.2.4 oneM2M Service provisioning procedures

These procedures are used by an M2M field node in order to bootstrap an M2M service subscription provisioned on the UICC.

The M2M field node shall perform the reading procedure with $EF_{1M2MSID}$ and $EF_{1M2MSPID,}$ and EF_{CSEID}, EF_{M2MNID}, $EF_{INCSEID}$, $EF_{MAFFQDN}$ according to available services indicated in EF_{1M2MST}.

D.2.2.5 oneM2M Application Identifiers provisioning procedure

This procedure provisions a list of M2M Application Identifiers that may be enabled on the M2M node in relation with the oneM2M Service Subscription.

Condition: Service number 4 shall be available in the oneM2M Service Table.

Under this condition, the M2M field node shall perform the reading procedure with $EF_{M2MAEID}$.

D.2.2.6 oneM2M Secure provisioning related procedures

These procedures are used by the M2M field node to perform M2M Secure Provisioning with the assistance of the UICC, depending on available services in EF_{1M2MST} and the supported AUTHENTICATE commands contexts (e.g. GBA support by a Network Access Application) indicated for the hosting ADF.

Secure Provisioning: MEF address Provisioning:

Condition: Service number 5 shall be available in the oneM2M Service Table.

Under this condition, the M2M field node shall perform the reading procedure with EF_{MEFID}, if the related service is available.

GBA Secure Provisioning:

This procedure is dependent on the Authentication Framework supported by the UICC and indicated in the Service Table of the hosting ADF.

After identifying the supported authentication framework, the M2M field node shall check availability of Service number 7 in EF_{1M2MST}: If the service is available, the D/G M2M Node shall perform GBA-related procedures with AUTHENTICATE - GBA security context (Bootstrapping Mode and Derivation Mode) with the parameters for GBA secure provisioning.

D.2.2.7 oneM2M Security Association related procedures

GBA secure connection:

This procedure is dependent on the Authentication Framework supported by the UICC and indicated in the Service Table of the hosting ADF.

After identifying the supported authentication framework, the M2M field node shall check availability of Service number 12 in EF_{1M2MST}: If the service is available, the M2M field node shall perform a GBA-related procedures with AUTHENTICATE - GBA security context (Bootstrapping Mode and Derivation Mode) with the parameters for GBA Security Association.

Annex E (informative): Precisions for the UICC framework to support M2M Services

The present annex provides further practical information related to the UICC framework for oneM2M described in annex D.

E.1 Suggested content of the EFs at pre-personalization

If EFs have an unassigned value, it may not be clear from the main text what this value should be. This annex suggests values in these cases.

File Identification	Description	Value
'6F02'	1M2M Service Subscription Identifier	'8000FF…FF'
'6F03'	1M2M Service Provider Identifier	'8000FF…FF'
'6F04'	M2M Node Identifier	'8000FF…FF'
'6F05'	Local CSE Identifier	'8000FF...FF'
'6F06'	M2M Application Identifiers list	'00FF...FF' for each record
'6F07'	MEF Identifier	'00FF…FF' for each record
'6F08'	IN-CSE Identifiers list	'00FF...FF' for each record
'6F09'	MAF FQDN	'8000FF...FF'
'6F0A'	1M2M Service Table	Operator/Service Provider dependant

E.2 EF changes via Data Download or CAT applications

This clause defines if changing the content of an EF by the UICC OTA protocol or by a CAT Application is advisable. Updating of certain EFs "over the air" or "over the Internet" could result in unpredictable behaviour of the UE; these are marked "Caution" in the table below. Certain EFs are marked "No"; under no circumstances should "over the air / over the internet" changes of these EFs be considered.

File identification	Description	Change advised
'6F02'	1M2M Service Subscription Identifier	No
'6F03'	1M2M Service Provider Identifier	No
'6F04'	M2M Node Identifier	Caution
'6F05'	Local CSE Identifier	Caution
'6F06'	M2M Application Identifiers list	Caution
'6F07'	MEF Identifier	Caution
'6F08'	IN-CSE Identifiers list	Caution
'6F09'	MAF FQDN	Caution
'6F0A'	1M2M Service Table	Caution

E.3 List of SFI values at the ADF_{M2MSM} or DF_{M2M} level

File Identification	SFI	Description
'6F02'	'02'	M2M Service Subscription Identifier
'6F03'	'03'	M2M Service Provider Identifier
'6F04'	'04'	M2M Node Identifier
'6F05'	'05'	Local CSE Identifier
'6F06'	'06'	M2M Application Identifiers list
'6F0A'	'0A'	1M2M Service Table

All other SFI values are reserved for future use.

E.4 UICC related tags defined in annex J

Tag	Name of Data Element	Usage
'80'	MAF FQDN TLV data object	$EF_{MAFFQDN}$
'80'	M2M-Node-ID TLV Data Object	EF_{M2MNID}
'80'	Local CSE-ID TLV data object	EF_{CSEID}
'80'	M2M-SP-ID TLV data object	$EF_{1M2MSPID}$
'80'	M2M Subscription Identifier TLV data object	$EF_{1M2MSID}$
NOTE: The value 'FF' is an invalid tag value.		

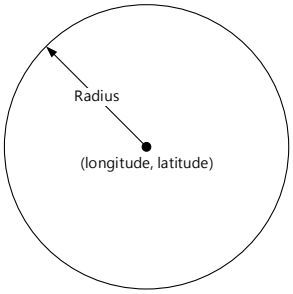
Annex F (normative): Acquisition of Location Information for Location based Access Control

When a request (resource access) is evaluated by a Hosting CSE and an *accessControlLocationRegions* parameter is defined in the *privileges* attribute of the <accessControlPolicy> resources, the Hosting CSE checks whether the location of the Originator of a request is in the specific regions or not. Therefore, the Hosting CSE retains the location of the Originator otherwise the Hosting CSE shall acquire the location or deny the access. This annex describes how to describe the location regions and obtain the location of the Originator.

F.1 Description of Region

F.1.1 Circular Description

The practical way of describing the region or area is the circular presentation and generally the circle is characterised by the co-ordinates of a center point of the circle and a radius. Geographically, the center point and radius is described as longitude and latitude, and meter respectively. For this description, the *accessControlLocationRegions* parameter shall be represented as a circle.



F.1.2 Country Description

Another simple way of describing the region or area is the country presentation. ISO-3166-1 alpha 2 codes [i.12] are two-letter country codes to represent countries and special regions of geographical interest. For example, KR is a code for Korea, Republic of.

F.2 Acquisition of Location Information

As mentioned above, when *accessControlLocationRegions* parameter is defined, the Hosting CSE shall check the location of the Originator for access control. This clause describes how the Hosting CSE checks or obtains the location. The procedures shall be varies based on the region description, circle and country.

F.2.1 Circular Description

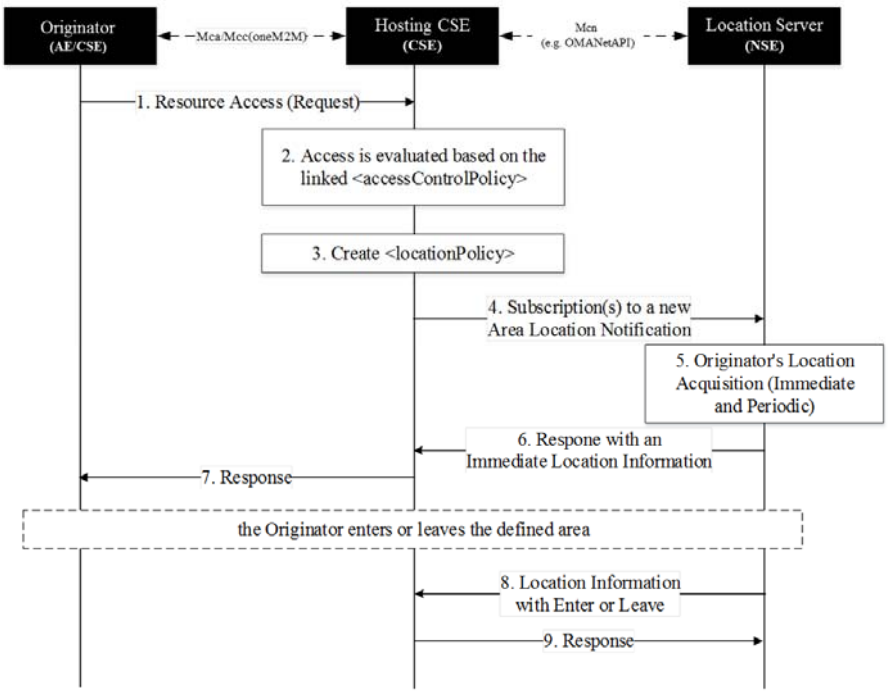
If the circular description is used as the location context constraints, the Hosting CSE shall check whether it has the current location of Originator or not. If not, it shall obtain the location of Originator. TS-0001[1] defines a resource type for acquisition of location of a Target Node, <locationPolicy>. In order to , therefore, obtain the location of Originator, the Hosting CSE shall create <locationPolicy> and set the relevant attributes as follows:

- ***locationSource*** : Reliability of the location information is crucial so the location shall be obtained from trusted network. If the location is obtained by the other sources, the location information can be easily masqueraded. (i.e. GPS spoofing). Therefore, the *locationSource* attribute shall be set to 'network-based'.
- ***locationTargetID* :** The Target Node shall be the Originator that needs to authorize the sent requests. The *locationTargetID* attribute shall be set to identifier of the Originator.

Note that the other attributes are determined by local policies of Hosting CSE as described in clause 9.6.9 of TS-0001 [1] and in order to obtain the location from the network, the Hosting CSE shall transform the oneM2M specified location request into network specified request.

NOTE: Refer to TS-0004 [4] that describes how to convert the oneM2M-specified request to 'OMA RESTful NetAPI for Terminal Location' specified request in clause annex F.

Since the region information (circular description) is defined by the *accessControlLocationRegions* parameter, the Hosting CSE can utilize the circular region information when it requests the location information from the network. OMA RESTful NetAPI for Terminal Location specification [i.11] specifies resource types as an area (region)-based location notification service, 'CircleNotificationSubscription'. If therefore the Hosting CSE subscribes to the notification service with circular region defined as *acccessControlLocationRegions* parameter, the Hosting CSE can always determine whether the Originator is in the regions or not. The figure below demonstrates how to acquire the location of the Originator when the *accessControlLocationRegions* parameter is defined.



1. The Originator sends a request to access a resource.

2. The Hosing CSE shall evaluate the received request against the linked <accessControlPolicy> resource. If one of rule tuples that is about the request originator contains the *accessControlLocationRegions* parameter (circular description) and the Hosting CSE does not store the location of the Originator, the Hosting CSE shall do either continue the next step or deny the access.
   If the Hosting CSE has the location of the Originator, it is used for applying access control policy.

   NOTE: The Hosting CSE shall deny the access due to the fact that the Originator is not subscriber of the network or any other reasons. (e.g., connection lost, server malfunction)

3. The Hosting CSE creates the <locationPolicy> resource and set relevant attributes as mentioned above.
4. The Hosting CSE subscribes to a new area location notification service toward Location Server in the Network. The area information shall be based on the area defined by the *accessControlLocationRegions* parameters. If the multiple regions are defined, the multiple subscriptions shall be set.
5. The Location Server immediately obtains the location's Originator.
   NOTE: After the immediate location acquisition, the Location Server periodically obtains the location's Originator to check whether the Originator is in the area or not. The frequency and duration can be defined by local policies.
6. The Location Server responses the immediate location information of the Originator toward Hosting CSE.
7. Based on the received location of the Originator and other access control policies the request can be either granted or denied. The Hosting CSE responses regarding the request (step 1).
8. When the Originator crossed in(enter) or out(leave) the area, the Location Server shall notify of the Hosting CSE the location change. Thus, the Hosting CSE can keep track of the location's Originator and easily evaluate the access against location context constraint.
9. The Hosting CSE response the notification.

F.2.2 Country Description

Generally, the Originator's country-scale location can be determined by the Originator's IP address. If the Hosting CSE can distinguish the country using the Originator's IP address and it is also matched with the defined *acccessControlLocationRegions* parameter, the Hosting CSE shall grant the request subject to the acceptance of the other access control policies. Note that how to transform the IP address into country is out of scope.

However, if Hosting CSE cannot distinguish the country using the Originator's IP address, The Hosting CSE shall obtain the location coordinate (i.e., longitude and latitude) of the Originator from network and the Hosting CSE can distinguish the country using the location if available. The way of obtaining the location coordinate is defined in annex F of TS-0004 [4]. Note that how to transform the location into country is out of scope.

Annex G (informative): Access Control Decision Request

An Access Control Decision Request as introduced in the Authorization Architecture in clause 6.2.2 is generated by a PEP according to an Originator's access request and extra information provided by the hosting CSE using the format specified by the PDP. The PEP can send the Access Control Decision Request to a PDP for an access control decision.

The PDP asks the PRP to retrieve all applicable access control policies according to the Access Control Decision Request, and then uses the Access Control Decision Request to evaluate the retrieved access control policies for an access control decision. An Access Control Decision Request from PEP to PDP can contain the following information:

- An Originator: It represents the ID of the Originator that sends an access request to the target resource.
- A Resource: It represents the URI of the target resource which the Originator wants to access.
- An Operation: It represents the operation which the Originator wants to perform on the target resource.

- An AccessTime: It represents the time of access.
- A LocationRegion: It represents the location of the Originator.
- An IPAddress: It represents the IP Address of the Originator.

The URI of the target resource is used to locate the target resource and then find the associated access control policies.

The ID of Originator is used to compare with the rule component subjects in order to check if a rule is applicable to the Access Control Decision Request.

The operation is used to compare with the rule component operations in order to check if the operation is permitted by the rule.

The AccessTime, LocationRegion and/or LocationRegion are used to check the rule component contexts in order to ensure some extra conditions are satisfied to using the rule for making an access control decision.

History

Draft history (to be removed on publication)		
V.0.0.0	07 Aug 2013	Initial version agreed at SEC#4 in oneM2M-SEC-2013-0026R01-Skeleton_TS_Security_Solutions
V0.1.0	18 Oct 2013	Incorporates following contributions agreed at SEC 7.0: oneM2M-SEC-2013-0041R01-SecurityCSF_Architecture oneM2M-SEC-2013-0044R04-Network-based_bootstrap oneM2M-SEC-2013-0050R03-PKI-Based_Post_Provisioning oneM2M-SEC-2013-0051R01-secure_remote_administration
V0.2.0	21 Jan 2014	Incorporates following contributions agreed at SEC 8.0: oneM2M-SEC-2013-0066R01-Bootstrapping_Definition oneM2M-SEC-2013-0070R03-GBA_framework oneM2M-SEC-2013-0073R04-Security_CSF_Architecture_Figure oneM2M-SEC-2013-0075R01-Clean_up_Security_TS
V0.3.0	25 Mar 2014	Incorporates following contributions agreed at SEC 9.0: SEC-2014-0044R03-SEC_CSF_Architecture_Figure_Update_for_Security_TS SEC-2014-0045R02-SEC_CSF_Authorization_Description_Update_for_Security_TS SEC-2014-0055R04-MAS_and_MTF_description
V0.4.0	15 Apr 2014	Incorporates following contributions agreed at SEC 10.0: SEC-2014-0240R02-SEC_CSF_Authorization_Procedure_for_Security_TS SEC-2014-0241R03-SEC_CSF_Architecture_General_Description_for_Security_TS SEC-2014-0259R01-Credentials_definitions SEC-2014-0260R01-Credentials_abbreviations SEC-2014-0243R03-Overview_of_Security_Frameworks SEC-2014-0242R03-Certificate-Based_Security_Framework_Common_Details

Draft history (to be removed on publication)		
		SEC-2014-0247R03-Overview_of_Security_Bootstrap_Frameworks SEC-2014-0244R04-Certificate-Based_Security_Bootstrap_Framework SEC-2014-0252R03-Pre-Provisioned_Symmetric_Enrolee_Key_Security_Bootstrap_Framework SEC-2014-0224R07-Overview_of_Security_Association_Establishment_Frameworks SEC-2014-0227R05-Certificate-Based_Security_Association_Establishment_Framework SEC-2014-0226R05-Pre-Provisioned_Symmetric_Association_Key_Framework SEC-2014-0254R02-Precisions_on_M2M_Trust_Functions SEC-2014-0239R01-Clarifications_to_TS-0003 Following actions have been resolved and changes incorporated into this version of the document: A-WG4-TP10_006: extract table from document SEC-2014-0263-Temporary_Enrolment_Key_Terminology and add to TS 0003 as an informative Annex A-WG4-TP10_A02: make the necessary alignments regarding "Master Credential" when integrating the agreed contributions into the TS Several editorial corrections incorporated.
V0.5.0	02 June 2014	Incorporates following documents agreed during conference calls SEC-2014-0288R01-TS0003_V0_4_0_CSFcleanup SEC-2014-0277R03-cleanup_TS0003_clause8 SEC-2014-0249R05-General_Mutual_Authentication_Mechanism SEC-2014-0285R05-AccessControlPolicy_Processing_Summary

Draft history (to be removed on publication)		
V0.6.0	23 June 2014	Incorporates following contributions agreed at SEC 11.0: SEC-2014-0274R03-SA_Security_Association SEC-2014-0276R05-Remote_Admin_Clarification SEC-2014-0308R01-Clause_9_1_1_Credential_Configuration SEC-2014-0309R02-Clause_9_1_2_Association_Configuration SEC-2014-0310R02-Clause_9_2_1_Bootstrap_Credential_Configuration SEC-2014-0311R02-Clause_9_2_2_Bootstrap_Instruction_Configuration SEC-2014-0313R03-Group_Authentication_Necessity_for_Security_TS SEC-2014-0320-GBA-based_Security_Bootstrap SEC-2014-0329R02-SE_pre-provisioning_framework SEC-2014-0330R01-Completion_of_generic_description_of_GBA SEC-2014-0332R03-Stage3_GBA-based_Security_Association SEC-2014-0338-MAF-based_Security_Association SEC-2014-0340R01- Centralized_Key_Distribution_Server_Handshake_in_Security_Association SEC-2014-0342-Cleaned-up_baseline_for_R1_freeze Note: SEC-2014-0299-Identity_Protection_Description_for_Security_TS was agreed to be incorporated into the next release of the specification. It is therefore not yet implemented in this version of the document.

Draft history (to be removed on publication)		
V0.7.0	29 July 2014	Incorporates following changes agreed by email correspondence: SEC-2014-0302R02-Clause_10_2_TLS_and_DTLS_Details SEC-2014-0303R02-Clause_10_3_Direct_Security_Bootstrap_Framework_Algorithm_Details In addition following contributions agreed at SEC 12.0 are integrated: SEC-2014-0328R03-Request_for_Changing_the_Text_of_SEC-2014-0249R05 SEC-2014-0356-TS-0003_CR_on_Definitions SEC-2014-0357R03-TS-0003_CR_updating_Annex_D_SE_Provisioning SEC-2014-0358-TS-0003_CR_Informative_Annex_on_SE_provisioning SEC-2014-0362R05-AccessControl_clarifications SEC-2014-0363R01-Configuration_of_Protection_levels SEC-2014-0364R03-Security_Considerations_on_AE SEC-2014-0365R02-oneM2M_Access_Control_Decision_Request_for_Annex SEC-2014-0366R03-Acquisition_of_Location_Information_for_Location_based_Access_Control SEC-2014-0367R01-Proposal_for_Changing_PAP_to_PRP SEC-2014-0368R06-TS-0003_Clause_8_1_etc_Certificate_Text_CR SEC-2014-0371R02-Clause_6_cleanup_CR SEC-2014-0373R02-TS-0003_Clause_10_2_TLS_and_DTLS_Details_Update SEC-2014-0377R01-GBA_missing_references SEC-2014-0378-TS-0003_CR_to_complete_Action_Items_on_8_1 SEC-2014-0379R01-TS-0003_CR_changing_MEF_bootstrap_classification SEC-2014-0380R01-TS-0003_CR_generalizing_purpose_of_MEF_bootstrap_scenario SEC-2014-0384-TS-0003_CR_Clause_617_cleanup SEC-2014-0385R03-Integrated_introductory_clause_from_0359_0382_0371 SEC-2014-0386R03-TS-0003_Clause_3_Updated_Definitions_CR SEC-2014-0387R01-TS-0003_Clause_8_2_etc_Security_Association_Cleanup_CR SEC-2014-0388R02-TS-0003_Clause_8_3_etc_Remote_Provisioning_Cleanup_CR SEC-2014-0389R02-TS-0003_Clause_9_1_cleanup_CR SEC-2014-0390R03-TS-0003_Clause_9_2_cleanup_CR SEC-2014-0393R02-TS-0003_Clause_10_3_Remote_Provisioning_Details_CR