DRAFT INTERNATIONAL STANDARD

ISO/IEC DIS 18013-5

ISO/IEC JTC **1**/SC **17** Secretariat: **BSI**

Voting begins on:
2020-02-05

Voting terminates on:
2020-04-29

Personal identification — ISO-compliant driving licence —

Part 5:
Mobile driving licence (mDL) application

Identification des personnes — Permis de conduire conforme à l'ISO —

Partie 5: Application permis de conduire sur téléphone mobile

ICS: 35.240.15

Reference number
ISO/IEC DIS 18013-5:2020(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

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

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

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

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 17 Cards and security devices for personal identification.

ISO/IEC 18013 consists of the following parts, under the general title Personal identification — ISO-compliant driving licence:

— *Part 1: Physical characteristics and basic data set.* Part 1 describes the basic terms for this document including physical characteristics, basic data element set, visual layout, and physical security features;

— *Part 2: Machine-readable technologies.* Part 2 describes the technologies that may be used for this document, including the logical data structure and data mapping for each technology;

— *Part 3: Access control, authentication and integrity validation.* Part 3 describes the electronic security features that may be incorporated under this document, including mechanisms for controlling access to data, verifying the origin of an IDL, and confirming data integrity;

— *Part 4: Test methods.* Part 4 describes the test methods that can be used to determine if an IDL conforms to the requirements for machine readable technologies specified in Part 2 and to the electronic security features specified in Part 3.

— *Part 5: Mobile Driving Licence (mDL) application.* Part 5 describes interface specifications for the implementation of a driving licence in association with a mobile device.

Introduction

This document describes interface and related requirements to facilitate ISO-compliant driving licence (IDL) functionality on a mobile device. The requirements are specifically intended to enable verifiers not affiliated with or associated with the issuing authority to gain access to and authenticate the information. In addition, the requirements allow the holder of the driving licence to decide what information to release to a verifier. Other major advantages include the ability to update information frequently, and to authenticate information at a high level of confidence.

ISO/IEC 18013 establishes guidelines for the design format and data content of an ISO-compliant driving licence (IDL) with regard to human-readable features (ISO/IEC 18013-1), ISO machine-readable technologies (ISO/IEC 18013-2), access control, authentication and integrity validation (ISO/IEC 18013-3), and associated test methods (ISO/IEC 18013-4). It creates a common basis for international use and mutual recognition of the IDL without impeding individual countries/states in applying their privacy rules and national/community/regional motor vehicle authorities in taking care of their specific needs.

The purpose of an IDL with one or more machine-readable technologies storing IDL data is to

— increase productivity (of data and IDL use),

— facilitate IDL data exchange, and

— assist in authenticity and integrity validation.

— Provide strong security and privacy features

Personal identification — ISO-compliant driving licence —

Part 5:
Mobile driving licence (mDL) application

1 Scope

The purpose of this document is to standardize interface specifications for the implementation of a driving licence in association with a mobile device (mDL). This document standardizes the interface between the mDL and mDL Reader, and the interface between the mDL Reader and the issuing authority infrastructure. The standard also allow parties other than the issuing authority (e.g. other issuing authorities, or mDL Verifiers in other countries) to:

a) use a machine to obtain the mDL data,

b) tie the mDL to the mDL Holder,

c) authenticate the origin of the mDL data, and

d) verify the integrity of the mDL data.

The following items are out of scope for this document:

a) how user consent to share data is obtained

b) requirements on storage of mDL data and mdL private keys

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

BSI TR-03111, *Elliptic Curve Cryptography, Version 2.10, June 2018*

CA/Browser Forum *Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates*

draft-iets-cose-x509-04: *CBOR Object Signing and Encryption (COSE): Headers for carrying and referencing X.509 certificates*

FIPS 186-4:2013, *Digital Signature Standard (DSS)*

FIPS PUB 140-2, *Security requirements for cryptographic modules, May 2001*

ICAO Doc 9303-12, *Machine Readable Travel Documents, Part 12: Public Key Infrastructure for MRTDs, Seventh Edition,* 2015

ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*

ISO 3166-2, *Codes for the representation of names of countries and their subdivisions — Part 2: Country subdivision code*

ISO/IEC 7812:2017, *Identification cards -- Identification of issuers -- Part 1: Numbering system*

ISO/IEC 7816-3:2006, *Identification cards -- Integrated circuit cards -- Part 3: Cards with contacts -- Electrical interface and transmission protocols*

ISO/IEC 7816-4:2013, *Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange*

ISO/IEC 10113-2:2004, *Information technology -- Security techniques -- Hash-functions -- Part 3: Dedicated hash-functions*

ISO/IEC 14443-2:2016, *Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 2: Radio frequency power and signal interface*

ISO/IEC 14443-3:2016, *Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 3: Initialization and anticollision*

ISO/IEC 14443-4:2018, *Cards and security devices for personal identification -- Contactless proximity objects -- Part 4: Transmission protocol*

ISO/IEC 14443-3:2018, *Cards and security devices for personal identification -- Contactless proximity objects -- Part 3: Initialization and anticollision*

ISO/IEC 15408:2009, *Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 1: Introduction and general model*

ISO/IEC 18004:2015, *Information technology – Automatic identification and data capture techniques – QR Code 2005 bar code symbology specification*

ISO/IEC 18013-1:2018, *Information technology -- Personal identification -- ISO-compliant driving licence – Part 1: Physical characteristics and basic data set*

ISO/IEC 18013-2:2008, *Information technology -- Personal identification -- ISO-compliant driving licence -- Part 2: Machine-readable technologies*

ISO/IEC 18013-3:2017, *Information technology -- Personal identification -- ISO-compliant driving licence -- Part 3: Access control, authentication and integrity validation*

ISO/IEC 19785-3:2007, *Information technology — Common Biometric Exchange Formats Framework — Part 3: Patron format specifications*

ISO/IEC 19790:2012, *Information technology -- Security techniques -- Security requirements for cryptographic modules*

NFC Forum, *Bluetooth Secure Simple Pairing Using NFC, NFCForum-AD-BTSSP_1_2, May* 2019

NFC Forum, *Connection Handover, Version 1.5,* 2019

NFC Forum, *Technical Specification - NFC Data Exchange Format (NDEF)*

NIST SP 800-38D, *M. Dworkin, Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007*

NIST SP 800-157, *H. Ferraiolo et al., Guidelines for Derived Personal Identity Verification (PIV) Credentials, December 2014*

OpenID Connect Core 1.0, *N. Sakimura et. al., Defines the core OpenID Connect functionality: authentication built on top of OAuth 2.0 and the use of claims to communicate information about the End-User, November* 2014

OpenID Connect Discovery *N. Sakimura et. al., Defines how clients/readers dynamically discover information about OpenID Providers, November* 2014

OpenID Connect Dynamic Registration *N. Sakimura et. al., Defines how clients/readers dynamically register with OpenID Providers, November* 2014

RFC 2104, *H. Krawczyk et al., HMAC: Keyed-Hashing for Message Authentication, February 2017*

RFC 2616, *R. Fielding et al., Hypertext Transfer Protocol -- HTTP/1.1, June 1999*

RFC 3339, *G. Klyne et al., Date and Time on the Internet: Timestamps, July 2002*

RFC 4122, *P. Leach et al., A Universally Unique IDentifier (UUID) URN Namespace, July 2005*

RFC 5246, *T. Dierks et al., The Transport Layer Security (TLS) Protocol Version 1.2, August 2008*

RFC 5280, *D. Cooper et al., Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008*

RFC 5639, *M. Lochter et al., Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, March 2010*

RFC 5652, *R. Housley, Cryptographic Message Syntax (CMS), September 2009*

RFC 5754, *S. Turner, Using SHA2 Algorithms with Cryptographic Message Syntax, January 2009*

RFC 5869, *H. Krawczyk, HMAC-based Extract-and-Expand Key Derivation Function (HKDF), May 2010*

RFC 6960, *S. Santesson et al., X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP, June 2013*

RFC 7049, *C. Bormann et al., Concise Binary Object Representation (CBOR), Oct 2013*

RFC 7515, *J. Bradley et al., JSON Web Signature (JWS), May 2015*

RFC 7518, *M. Jones et al., JSON Web Algorithms (JWA), May 2015*

RFC 7519, *J. Bradley et al., JSON Web Token (JWT), May 2015*

RFC 7748, *A. Langley et al., Elliptic Curves for Security, Jan 2016*

RFC 7905, *A. Langley et al., ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS), Jun 2016*

RFC 8032, *S. Josefsson et al., Edwards-Curve Digital Signature Algorithm (EdDSA), January 2017*

RFC 8152, *J. Schaad, CBOR Object Signing and Encryption (COSE), July 2017*

RFC 8259, *T. Bray, The JavaScript Object Notation (JSON) Data Interchange Format, December 2017*

RFC 8422, *Y. Nir et al., Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) Versions 1.2 and Earlier, Aug 2018*

RFC 8446, *E. Rescorla et al., The Transport Layer Security (TLS) Protocol Version 1.3, August 2018*

RFC 8610, *H. Birkholz et al., Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures, June 2019*

SP 800-56A Rev. 3, *Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography*

Wi-Fi Alliance *Neighbor Awareness Networking Technical Specification, Version 3.0, December* 2018

Wi-Fi Alliance *Neighbor Awareness Networking Specification v3.0 draft Addendum version 0.0.2., April* 2019

3 Terms and definitions

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

3.1
mobile device
portable computing device that at least:

(i) has a small form factor such that it can easily be carried by a single individual;

(ii) is designed to operate, transmit and receive information without a wired connection;

(iii) possesses local, nonremovable or removable data storage; and

(iv) includes a self-contained power source

(v) includes a display; and

(vi) includes a mean for the user to interact with a device

[SOURCE: NIST SP 800-157, modified]

3.2
mDL
driving licence that fulfils at least the same function as an IDL (ISO/IEC 18013-1) but, instead of being paper or plastic based, resides on a mobile device or requires a mobile device as part of the process to gain access to the driving licence

3.3
mDL Holder
legitimate holder of the driving privileges reflected on an mDL

3.4
mDL Reader
device that can retrieve mDL data for verification purposes

3.5
mDL Verifier
a person or organization using and/or controlling an mDL Reader to verify an mDL

3.6
issuing authority infrastructure
infrastructure under control of the issuing authority

4 Abbreviated terms

APDU	Application Protocol Data Unit
BER	Basic Encoding Rules
BLE	Bluetooth Low Energy
BT SIG	Bluetooth Interest Group
CA	Certificate Authority
CBOR	Concise Binary Object Representation
CDDL	Concise data definition language
COSE	CBOR Object Signing and Encryption
CSPRNG	Cryptographically Secure Pseudo-random Number Generator
CRL	Certificate Revocation List

DER	Distinguished Encoding Rules
DO	Data Object
DS	Document Signer
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EdDSA	Edwards-curve Digital Signature Algorithm
GATT	Generic Attribute Profile
HKDF	HMAC-based Extract-and-Expand Key Derivation Function
IA	Issuing Authority
IACA	Issuing Authority Certificate Authority
IAPC	Issuing Authority Point of Contact
IDL	ISO-compliant driving licence
IKM	Input Keying Material
JWT	JSON Web Token
JWS	JSON Web Signature
JWA	JSON Web Algorithms
KDF	Key Derivation Function
MAC	Message Authentication Code
MITM	Man-in-the-middle attack
ML	Master List
MSO	Mobile Security Object
MTU	Maximum Transmission Unit
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OIDC	OpenID Connect
PIX	Proprietary Application Identifier Extension
PKI	Public Key Infrastructure
RID	Registered Application Provider Identifier
TLS	Transport Layer Security

TLV	Tag Length Value
UHF	Ultra High Frequency
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
UTC	Coordinated Universal Time
UUID	Universally unique identifer

5 Conformance requirement

An mDL is in conformance with this document if it meets all mandatory requirements specified directly or by reference herein. Compliance with ISO/IEC 18013-1, ISO/IEC 18013-2, ISO/IEC 18013-3 and ISO/IEC 18013-4 is not required for compliance with this document, except for those clauses directly referenced in this document.

An mDL Reader is in conformance with this document if it meets all mandatory requirements specified directly or by reference herein.

An issuing authority infrastructure is in conformance with this document if it meets all mandatory requirements specified directly or by reference herein.

6 mDL overview

6.1 Introduction

Figure 1 shows the interfaces in scope for this document. The explanation of each interface is:

1) This is the interface between the issuing authority infrastructure and the mDL. This interface is out of scope for this document.

2) This is the interface between the mDL and the mDL Reader. This interface is specified in this document. The interface can be used for connection setup and for offline data retrieval.

3) This is the interface between the issuing authority infrastructure and the mDL Reader. This interface is specified in this document. The interface can be used for the online data retrieval method.

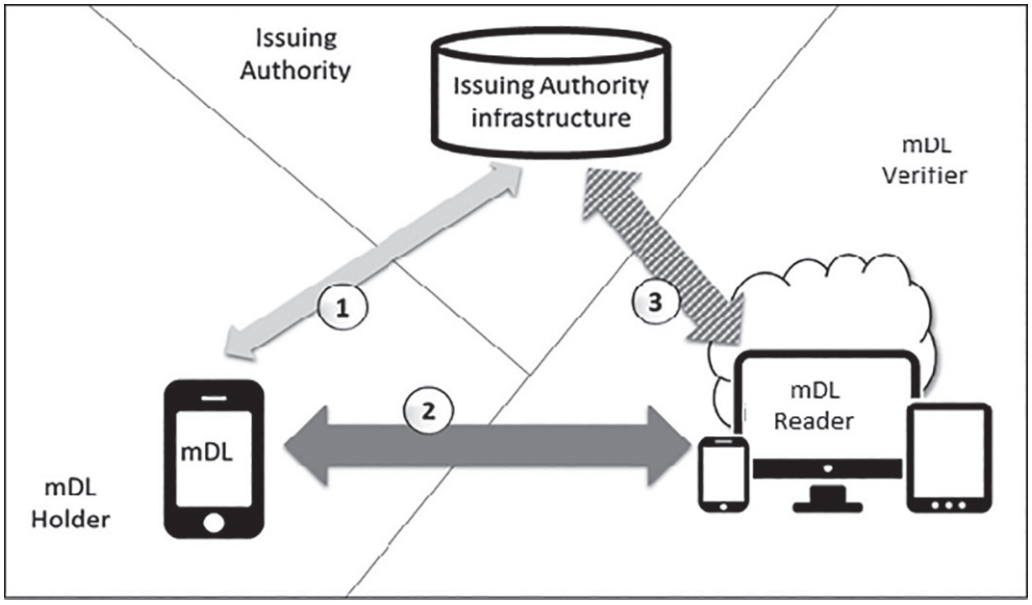

Figure 1 — mDL interfaces

See Annex A for examples of use cases.

6.2 Functional requirements

The specific functional requirements covered in this part of ISO/IEC 18013 for the mDL include at least:

a) An mDL Verifier together with an mDL Reader shall be able to request, receive and verify the integrity and authenticity of an mDL whether online connectivity is present or not on either the mDL or mDL Reader.

b) Verifiers not associated with the issuing authority shall be able to verify the integrity and authenticity of an mDL.

c) An mDL Verifier shall be enabled to confirm the binding between the person presenting the mDL and the mDL Holder.

d) The interface between the mDL and the mDL Reader shall support the selective release of mDL data to an mDL Reader.

The interface between the issuing authority and the mDL shall support the ability to update information. The mechanism of this ability is out of scope for this document.

6.3 Technical requirements

6.3.1 Data model

The mDL data model is described in 7.

6.3.2 Data exchange

6.3.2.1 Overview

Data exchange is divided into three phases: initialization phase, device engagement phase and data retrieval phase (see 8 and Figure 2). After initialization between the mDL and the mDL Reader three main transaction flows are distinguished:

— Device engagement, followed by exchange of data by offline retrieval between the mDL and the mDL Reader (see (1) in Figure 2)

— Device engagement, followed by exchange of online token using offline retrieval between the mDL and the mDL Reader, followed by exchange of data by online retrieval between the mDL Reader and the issuing authority. (see (2) in Figure 2)

— Device engagement, followed by exchange of data by online retrieval between the mDL Reader and the issuing authority infrastructure. (see (3) in Figure 2)

For offline retrieval, there is no requirement for any device involved in the transaction to be connected to the internet.

NOTE The transaction has been designed such that it is not be necessary for the mDL Holder to physically hand over the mobile device to the mDL Verifier.

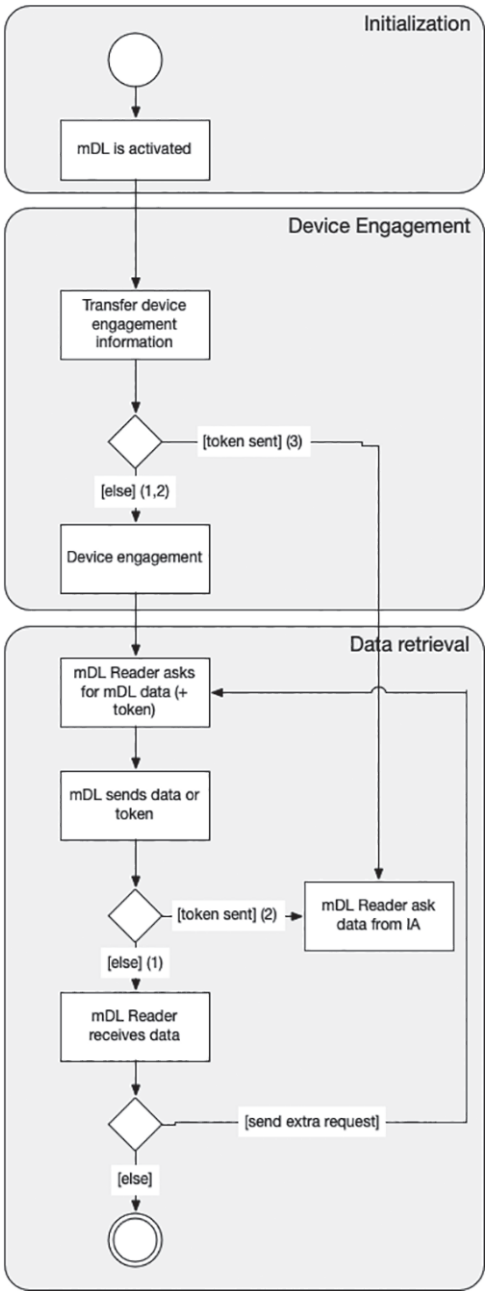

Figure 2 — mDL transaction flow

6.3.2.2 Initialization

During initialization, an mDL is activated (by the mDL Holder, or potentially triggered by NFC). No requirements are specified for this phase

6.3.2.3 Device engagement

During device engagement, information required to setup and secure data retrieval is exchanged between the mDL to the mDL Reader. Transmission technologies available to transfer the device engagement data are as follows:

a) NFC (Type 4 tag Platform using NDEF, see NFC Forum, *Technical Specification - NFC Data Exchange Format (NDEF))*

b) Barcode (QR code)

Table 1 shows the different device engagement technologies for Device Engagement.

Table 1 — Device Engagement Technologies

Transmission technologies	Support		Reference
	mDL	mDL Reader	
NFC	C[a]	M	8.1.2.1
Barcode	C[a]	M	8.1.2.3
Key C/M: is conditional (C) or mandatory (M) [a] Support for at least one of these methods is mandatory.			

To ensure that the mDL can be verified an mDL shall support at least one of the transmission technologies. An mDL Reader shall support all transmission technologies.

The device engagement information, described in 8.1.1, is transferred using one of the transmissions technologies, described in 8.1.2.

6.3.2.4 Data retrieval architecture

Figure 3 shows the different data retrieval interfaces and the flow of the messages.

When using offline retrieval, the mDL and mDL Reader communicate using mDL Reader Request and mDL Response messages encoded with CBOR. These messages are then transported using different data retrieval methods. The data retrieval methods are agnostic to the information that is transferred.

After device engagement, if the mDL Reader sets up an offline retrieval connection, the mDL Reader asks for data as defined in 8.2.1.1.2.1. The mDL sends a mDL Response according to 8.2.1.1.2.2. The mDL Reader Request may include a request for a token used to perform online retrieval. If the token is requested, the mDL shall not return both the mDL data and the token.

If the mDL Reader retrieves the token and URL from the mDL, either during device engagement or offline data retrieval, it can retrieve mDL data from the issuing authority using the internet. Either OIDC or WebAPI is used to retrieve the information.

The different data retrieval methods are described in 6.3.2.5

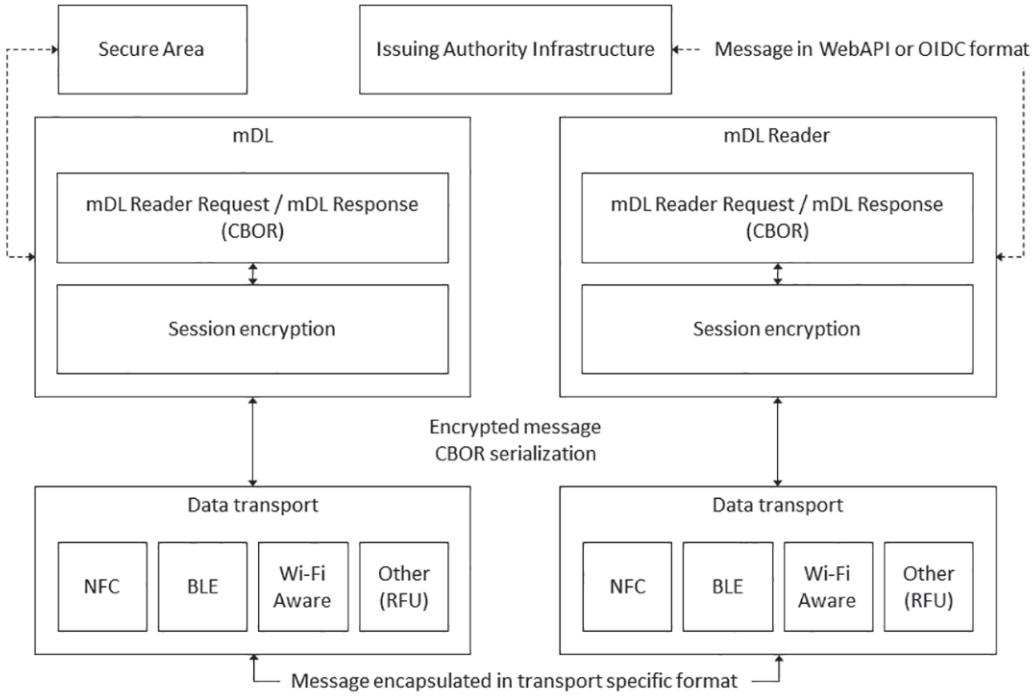

Figure 3 — Data retrieval architecture

NOTE The secure area as present in the figure indicates an area that provides additional protection of sensitive mDL related data. Security requirements regarding storage of credential information, including the mDL private key are out of scope for this document. It is the responsibility of the Issuing Authority to ensure that all data stored on the mDL is stored securely.

6.3.2.5 Data retrieval methods

mDL data can be retrieved in two ways:

a) Using Offline retrieval (interface 2 in Figure 1), see 8.2.1.1.

b) Using Online retrieval (interface 3 in Figure 1), see 8.2.1.2, where the online token can be retrieved from the mDL during Device Engagement or during Offline retrieval

Table 2 shows the different transmission technologies for the different data retrieval methods:

Table 2 — Data retrieval methods

Data retrieval method	Transmission technology	Support		Reference
		mDL	mDL Reader	
Offline retrieval	BLE	C [a]	M	8.2.2.1
	NFC	C [a]	M	8.2.2.2
	Wi-Fi Aware	O	R	8.2.2.3
Online retrieval	Internet (WebAPI)	O	R	8.2.2.4
	Internet (OIDC)	O	R	8.2.2.5

Key

M/C/R/O: Is mandatory (M), Conditional (C), Recommended (R) or optional (O)

[a] Support for at least one of these methods is mandatory.

To ensure that the mDL can be verified, an mDL shall support the BLE or NFC transmission technologies. An mDL Reader shall support the BLE and NFC technologies.

6.3.3 Security mechanisms

Security mechanisms to preserve confidentiality, integrity and authenticity of the mDL are described in 9.

7 Data model

7.1 Overview

This clause presents data model for an mDL. 7.2 describes in what way the data is structured for an mDL, 7.3 defines CDDL items used in structures and 7.4 lists all data elements.

7.2 Encoding of data structure and data elements

In this document CDDL (Concise Data Definition Language) is used to define data structures to express CBOR and JSON. CDDL as used in this document is specified in RFC 8610, *H. Birkholz et al., Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures.* JSON is standardized in RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format.*

For offline retrieval, messages and data elements are encoded with CBOR. For online retrieval, messages and data elements are encoded with JSON.

Section 3.9 in RFC 7049 describes four rules for 'Canonical CBOR'. Three of those rules shall be implemented for all CBOR structures as follows:

— Integers must be as small as possible.

— The expression of lengths in major types 2 through 5 must be as short as possible.

— Indefinite-length items must be made into definite-length items.

These are further described in 3.9 in RFC 7049. The fourth rule regarding sorting of map keys is not required. Furthermore, maps (major type 5) shall not have multiple entries with the same key.

7.3 nameSpace and DocType

7.3.1 General

The items described in the following subclause are to identify nameSpace for the data elements and the DocType provided.

7.3.2 DocType

DocType encapsulates the document type requested or returned. The mDL DocType shall be "org.iso.18013.5.1.mDL". The number "1" in the DocType might be increased in future versions of the standard.

NOTE The DocType field follows the following general format: [Reverse Domain].[Domain Specific Extension]. The reverse domain (org.iso) was selected to avoid collisions. This approach can be used to define other doctypes.

7.3.3 nameSpace

nameSpace provides the definition within which the data elements of the document are defined. A document may have multiple nameSpaces. The meaning of data elements is dependent on which nameSpace it belongs to.

The nameSpace for the mDL data defined in 7.4 shall be “org.iso.18013.5.1”. The number “1” in the NameSpace might be increased in future versions of the standard.

NOTE The nameSpace field follows the following general format: [Reverse Domain].[Domain Specific Extension].

A country and/or an issuing authority shall define its own nameSpace to accommodate domestic data, using the structure described (see 7.4.9).

7.4 mDL data

7.4.1 Overview

The data elements described in Table 3 belong to nameSpace “org.iso.18013.5.1”.

— The Identifier column is used for DataItemNames in the mDL Reader request structure (see 8.2.1.1.2.1).

— The Presence column indicates whether the presence of the element on an mDL is mandatory (M), recommended (R) or optional (O).

NOTE This does not indicate if granting access of these elements to an mDL Reader is mandatory.

— The Field format column defines the format of data elements; F followed by number as fixed length indicated by its number, V followed by number as variable length with maximum length indicated by its number, A as an alphabetic character, N as a numeric character, S as a special character. The key at the bottom of Table 3 further defines A, N and S. N/A indicates that there are no further limitations to the content of the value beyond what is imposed by the definition and the encoding.

— The encoding column indicates how to encode the data element and is defined as per CBOR RFC 7049 for both online and offline. Encoding types are defined in CDDL RFC 8610. For online data retrieval methods, the data elements are converted to JSON (see 4.1 of RFC 7049, bstr data elements are encoded as base64url-without-padding string).

NOTE the allowed characters are always limited by the field format if applicable, after which the encoding rules are applied.

— Dates in this document use either date-time or full-date from RFC3339, unless otherwise indicated. date-time is encoded according to RFC 7049 2.4.1 and uses Tag value 0 of major type 6, described as tdate (tdate = #6.0(tstr)). For full-date the tag #6.18013(tstr) will be used.

NOTE A request for a semantic tag for full-date will be requested and the tag used in this document is therefore subject to change in future versions of the standard.

Table 3 — Data elements

Identifier	Name	Definition	Presence	Field format	Encoding
family_name	Family name	Last name, surname, or primary identifier, of the licence holder	M	V36AS	tstr
given_name	Given names	First name(s), other name(s), or secondary identifier, of the licence holder	M	V36AS	tstr
birth_date	Date of birth	Day, month, year on which the licence holder was born. If unknown, approximate Date of birth	M	full-date	#6.18013(tstr)

Table 3 *(continued)*

Identifier	Name	Definition	Presence	Field format	Encoding
issue_date	Date of Issue	Date licence document was issued	M	full-date or date-time	tdate or #6.18013(tstr)
expiry_date	Date of Expiry	Date licence document expires	M	full-date or date-time	tdate or #6.18013(tstr)
issuing_country	Issuing country	country code as alpha 2 code, defined in ISO 3166-1, which issued the mDL or within which the licensing authority is located	M	F2A	tstr
issuing_authority	Issuing authority	Name of licensing authority, or issuing country if separate licensing authorities have not been authorized. See 7.4.3.	M	V65ANS	tstr
document_number	Licence number	The number assigned or calculated by the issuing authority	M	V25AN	tstr
administrative_number	Administrative number	An audit control number assigned by the licensing authority	O	V25ANS	tstr
driving_privileges	Categories of vehicles/ restrictions/ conditions	Driving privileges the licence holder is authorized to drive. It consists of category issue date, expiry date, restriction/condition sign code, restriction/condition sign and restriction/condition value. See 7.4.4.	M	See 7.4.4	See 7.4.4.
un_distinguishing_sign	UN distinguishing sign	Distinguishing sign of the issuing country according to 18013-1 annex F NOTE this field is added for purposes of the UN conventions on driving licences	R	N/A	tstr
gender	Gender	Licence holder's gender: M for male, F for female, X for not specified	O	F1A	tstr
height	Height (cm) [a]	Licence holder's height in centimetres	O	F3N	uint
weight	Weight (kg) [a]	Licence holder's weight in kilograms	O	F3N	uint
eye_color	Eye colour	Licence holder's eye colour: blue, brown, black, hazel, green, grey, pink, dichromatic	O	V12A	tstr
hair_color	Hair colour	Licence holder's hair colour: brown, black, blonde, grey, red, auburn, sandy, white, bald	O	V12A	tstr
birth_place	Place of birth	Country and municipality or state/province where the licence holder was born	O	V33A	tstr
resident_address	Permanent place of residence	The place where the licence holder resides and/or may be contacted (street/house number, municipality etc.)	O	V108ANS	tstr
portrait	Portrait of mDL Holder	A reproduction of the licence holder's portrait. See 7.4.2.	M	N/A	bstr
portrait_capture_date	Portrait image timestamp	Date when picture was taken	O	date-time	tdate
age_in_years	Age attestation: How old are you (in years)?	The age of the mDL Holder	O	V3N	uint
age_birth_year	Age attestation: In what year were you born?	The year when the mDL Holder was born	O	F4N	uint
age_over_NN	Age attestation: Nearest "true" attestation above request	See 7.4.5	O	N/A	bool
issuing_jurisdiction	Issuing jurisdiction	Country subdivision code as defined in clause 8, ISO 3166-2. The first part of the code shall be the same as the value for issuing_country. This element is intended to be used in cases where the issuing jurisdiction is different from the issuing authority.	O	N/A	tstr

Table 3 *(continued)*

Identifier	Name	Definition	Presence	Field format	Encoding
nationality	Nationality	Nationality of the mDL Holder as two letter country code (alpha-2 code) defined in ISO 3166-1	O	F2A	tstr
resident_city	Resident city	The city where the mDL Holder lives	O	ANS	tstr
resident_state	Resident state/province/district	The state/province/district where the mDL Holder lives	O	ANS	tstr
resident_postal_code	Resident postal code	The postal code of the mDL Holder	O	ANS	tstr
biometric_template_xx	Biometric template XX	See 7.4.6	O	N/A	bstr
name_national_character	Full name of holder in full UTF-8 character set	The full name of the mDL Holder in his/her national characters	O	N/A	tstr
signature_usual_mark	Signature / usual mark	Image of the signature or usual mark of the mDL Holder	O	See 7.4.7	See 7.4.7
online_token_xxxx	Online token	See 7.4.8	O	N/A	tstr
online_url_xxxx	Online URL	See 7.4.8	O	N/A	tstr

Key

Presence:

M/R/O: The presence is mandatory (M), recommended (R) or optional (O)

Field format:

A: alphabetic character, hexadecimal ranges '41' – '5A' (Latin capital letters), '61' – '7A' (Latin small letters), 'C0' – 'D6', 'D8' – 'F6' and 'F8' – 'FF' of ISO/IEC 8859-1

N: numeric character, hexadecimal range '30' – '39' (digits 0 to 9) of ISO/IEC 8859-1

S: special character, hexadecimal ranges '20' – '2F' (<space> ! " # $ % & ' ( ) * + , - . /), '3A' (:), '3C' – '40' (< = > ? @), '5B' – '60' ([\]^_`),'7B'–'7E'({|}~),'A1'–'AC'(¡¢£¤¥¦§¨©a«¬),'AE'–'A5'(®¯°±2 3 ´µ),and'A7'–'BF'(·¸ 1 ° » 1/4 1/2 3/4 ¿) of ISO/IEC 8859-1

NOTE to entry A, N and S: In this definition ISO/IEC 8859-1 is used for identification of the character and not for encoding.

full-date and date-time: according to RFC3339. For date-time, further refined in 3.3 of RFC4287

[a] The mDL Reader is expected to be able to convert to the local scale.

If any data element is returned to the mDL Reader, then information necessary to verify that the person presenting the mDL is the mDL Holder shall be returned to the mDL Reader if that information is requested.

NOTE In this document, from the set of data items with mandatory presence in column 'presence' from Table 3, the portrait of the mDL Holder is the only data item for verifying that the person presenting the mDL is the mDL Holder.

An mDL may require mDL Reader authentication (see 9.2.4) before releasing data elements not marked as mandatory in Table 3. An mDL shall not require mDL Reader authentication as a precondition for the release of any of the mandatory elements. However, an mDL may offer functionality to the mDL Holder to pre-authorize the release of mandatory data elements selected by the mDL Holder to mDL Readers using mDL Reader authentication.

NOTE The intention of this requirement is that the mDL Holder is always able to use the mDL as a driving licence if it chooses to do that, in the case that an mDL Reader does not use mDL Reader authentication.

When offline retrieval is used, any element in Table 3 (except for online_token_xxxx and online_url_xxxx) that is returned by the mDL, shall be returned as part of the IssuerSignedItems. Domestic

data elements (see 7.4.97.4.7) and/or Online token elements (see 7.4.8) shall be returned in either IssuerSignedItems or DeviceSignedItems.

NOTE Issuing authorities have the responsibility to ensure that controls to prevent the unintended release of mDL data are implemented, including providing mDL Holders with the means to control what data is released.

7.4.2 Portrait of mDL Holder

The portrait of mDL Holder consists of one portrait image and shall follow the requirements on the face image as specified in Annex E, ISO/IEC 18013-2:2008. One of the following image formats shall be used: JPEG or JPEG2000. For the offline data retrieval method, it shall be binary encoded.

7.4.3 Issuing authority

The "Issuing authority" element identifies the administrative authority entitled to issue the driving licence, or the issuing country if separate licensing authorities have not been authorised. The Issuing authority element is represented by a string.

NOTE The contents of this field correspond to the contents of the Issuing authority element on the IDL, and can indicate a local, regional or national organisation. Please note that, like in 18013-1, the term issuing authority can also refer to a central government agency, acting on behalf of multiple local or regional Issuing authorities.

7.4.4 Categories of vehicles/restrictions/conditions

The categories of vehicles/restrictions/conditions contain information describing the driving privileges of the mDL Holder. The definition of the elements in the DrivingPrivilege structure can be found in clause 5, ISO/IEC 18013-1:2018. The possible values for the elements are defined in Annex A, ISO/IEC 18013-2

For data retrieval the driving privileges shall have the following CDDL structure:

```
DrivingPrivileges = [
    * DrivingPrivilege
]

DrivingPrivilege  = {
    "vehicle_category_code" : tstr      ; Vehicle category code as per ISO 18013-2 Annex A
    ? "issue_date" : #6.18013(tstr)     ; Date of issue encoded as full-date per RFC 3339
    ? "expiry_date" : #6.18013(tstr)    ; Date of expiry encoded as full-date per RFC 3339
    ? "codes" : [+Codes]                ; Array of code info
}

Codes = {
    ? "code": tstr                      ; Code as per ISO/IEC 18013-2 Annex A
    ? "sign": tstr                      ; Sign as per ISO/IEC 18013-2 Annex A
    ? "value": int                      ; Value as per ISO/IEC 18013-2 Annex A
}
```

An informative example can be found in D.2.

7.4.5 Age attestation: Nearest “true” attestation above request

This set of elements is used to convey to a reader, in a data minimized fashion, if the mDL Holder is as old or older than a specified age. To achieve this, the mDL contains age attestation identifiers. An age attestation identifier has the format age_over_NN where NN is a value from 00 to 99.

When a verifier includes age_over_NN in a request, it has the meaning of “provide the nearest age attestation equal to or larger than NN”. The response to this request is the closest age attestation equal to or above NN present in the mDL, if one exists, expressed as “age_over_NN is true”.

Since the meaning of a "false" value can be ambiguous if that age was not requested. An element with the value "false" should only be returned if the data item name exactly matches the requested data element.

age_over_xx data elements shall be calculated at the value of the timestamp of the “validFrom” element in the MSO from 9.2.2.4

7.4.6 Biometric template

This element contains optional facial, fingerprint, iris or other biometric information of the holder. Biometric information is encoded according to the biometric templates defined in Table 6, ISO/IEC 18013-2:2008. The XX in the identifier is replaced with the corresponding “Abstract value name” found in Table 7, ISO/IEC 19785-3:2007 according to the following convention: capitalized characters are replaced with their lowercase equivalent and spaces or non-alphanumeric characters are replaced by underscores (_). As an example the “FACE” template corresponds to “biometric_template_face” and the “SIGNATURE/SIGN” template corresponds to “biometric_template_signature_sign”. If the facial image template is used, the portrait image shall follow the requirements on the face image as specified in 7.4.2.

7.4.7 Signature or usual mark

The signature or usual mark of the mDL Holder consists of one image. One of the following image formats shall be used: JPEG or JPEG2000. For the offline data retrieval method it shall be binary encoded.

7.4.8 Online token

This element lets the mDL provide the identity token of the user and URL of the issuing authority to facilitate the mDL Reader’s use of the online data retrieval method. The token and URL can be either OIDC or WebAPI type, xxxx denotec "oidc" or "webapi". For example, “online_token_oidc” or “online_token_webapi” would be used respectively for each token case.

If the online token is returned by the mDL in the mDL Response as defined in 8.2.1.1.2.2, it shall be returned in either “IssuerSignedItems” or “DeviceSignedItems”. The following data in Table 4 shall be returned:

Table 4 — Online token

Name	Definition	Encoding
online_token_xxxx	Token identifying the mDL Holder	tstr
online_url_xxxx	Base URL of the issuing authority	tstr

7.4.9 Domestic data elements

Domestic data are data elements which are not specified in this document. An issuing authority is able to specify its own data as domestic data within its namespace. Since the namespace for standardized mDL data in this document is “org.iso.18013.5.1”, it is recommended that country-specific and issuer-specific data append the ISO 3166-1 alpha-2 country code or the ISO 3166-2 region code after a period (e.g., the United States namespace will be “org.iso.18013.5.1.US” and the Iowa namespace will be “org.iso.18013.5.1.US-IA”).

7.5 Country codes

Within this document, ISO 3166-1 are used as a source for country code identifiers. If no applicable country code is available in ISO 3166-1, an IA may use one of the user-assigned country code elements, as indicated in ISO 3166-1. In this case the IA should ensure there is no collission with other IA’s.

8 Transaction

8.1 Device engagement

8.1.1 Device engagement information

8.1.1.1 Device engagement structure

The device engagement structure shall have the following CDDL structure:

```
DeviceEngagement = [
    tstr,               ; Version of device engagement structure, currently "1.0"
    Security,           ; Security structure
    TransferMethods,    ; Transfer methods
    Options,            ; Optional elements
   [*DocType],         ; May be used to indicate the supported DocType(s)
    ? ApplicationSpecific ; Application specific elements
]

DocType = tstr          ; See 7.3.2

Security = [
    uint                ; cipher suite identifier, see Table 22
   DeviceKeyBytes,    ; EDeviceKey.Pub,  see 9.2.1
]

EDeviceKeyBytes = #6.24(bstr .cbor EDeviceKey)
EDeviceKey = COSE_Key   ; containing the EDeviceKey.Pub

TransferMethods = [
    * TransferMethod    ; details for transfer methods
]

Options = {
    ?"webApi" : WebApi,
    ?"oidc" : Oidc
}

ApplicationSpecific = {
    * tstr => any       ; any number of elements
}

TransferMethod = [
    uint,               ; type
    uint,               ; version
    * TransferOptions   ; specific option(s) to the type
]

TransferOptions = WifiOptions / BleOptions / NfcOptions / any

; The value of these fields are defined in table 6 of
; Wi-Fi Alliance, Neighbor Awareness Networking Specification v3.0
; draft Addendum version 0.0.2, April 2019
WifiOptions = {
    ? 0: tstr       ; Password Info Password
    ? 1: uint       ; Channel Info Operation Class
    ? 2: uint       ; Channel Info Channel Number
}

BleOptions = {
      ? 0 : bool,       ; Indicates support for Peripheral Server mode
      ? 1 : bool,       ; Indicates support for Central Client mode
      ? 10 : bstr,      ; Optional UUID, encoded as big-endian for Peripheral Server mode
? 11 : bstr       ; Optional UUID, encoded as big-endian for Client Central mode
}

NfcOptions = {
0 : uint       ; Maximum length of command data field supported by mobile device,
```

```
; as defined in ISO/IEC 7816-4.
; NOTE: a value over 255 bytes indicates extended length.
}

Oidc = [
  uint,                 ; version, currently 1
  tstr,                 ; url
  tstr                  ; token
]

WebApi = [
  uint,                 ; version, currently 1
  tstr,                 ; url
  tstr                  ; token
]
```

An informative example can be found in D.3.

The above device engagement structure contains the following:

— Version: the version of the device engagement structure, currently “1.0”

— Security: an array that contains two mandatory values (mDL.EPub, cipher). The mDL.Epub is defined in 9.2.1 and cipher suites are defined in Table 22. The mDL public ephemeral key must be of a type allowed by the indicated cipher suite.

— TransferMethods: an array that contains one or more transferMethod arrays when performing device engagement using the QR code and is empty when using NFC to perform device engagement. This array is for offline data retrieval methods. A transferMethod array holds two mandatory values (type and version) and may contain extra elements that can hold specific options for each connection. Table 5 indicates values of each field depending on the data retrieval method.

— Options: an array that contains optional information on the online data retrieval method (OIDC and WebAPI). Both an OIDC array and a WebAPI array hold three mandatory values (version, url and token). The online retrieval options are explained in 8.1.1.2.

— ApplicationSpecific: an array that may contain any number of options for additional functionalities, country specific features, application specific data etc.

Table 5 — Transfer method parameters

	NFC	BLE	Wi-Fi Aware
type	1	2	3
version	1	1	1
options	‘NfcOptions’	‘BleOptions’	‘WifiOptions’

8.1.1.2 Online retrieval options

The online token and URL may be included as part of the device engagement data. There are two arrays that can be transferred, the ‘Oidc’ and the ‘WebApi’ array as defined in 8.1.1.1. Both arrays consist of three fields, the version, the URL and the token. The version indicates the version of the transfer methods, currently 1 for both OIDC and WebAPI. The URL and the token field are defined in 7.4.8.

If the online retrieval information is transferred in the Device Engagement structure, it is not protected by mDL authentication (9.2.3). Therefore, the IA is responsible for the online token sent in the device engagement structure to authorize the request as part of the transaction by or on behalf of the mDL Holder.

8.1.2 Device engagement transmission technology

8.1.2.1 Device engagement using NFC

The Connection Handover protocol is initiated by the mDL Reader as described in NFC Forum, *Connection Handover, Version 1.5, 2019*. Only the type 4 Tag Platform shall be used. The mDL may support the Static Handover and/or Tag NDEF Exchange Protocol (TNEP) for Negotiated Handover. The mDL Reader shall support both handover methods

NOTE As the connection handover protocol is still under development. The final decision on mandatory negotiated handover support by mDL Readers will be made after the applicable standards are final, when implementation consequences can be fully appreciated.

When static handover is performed, the retrieved Handover Select Message shall contain at least one alternative carrier ("ac") record. The "ac" records indicate which transmission technologies are supported. Therefore, the device engagement structure shall not contain any connection profiles when Device Engagement is performed with NFC. The "ac" record content is defined in the references in Table 6. Carrier specific service names or UUIDs will be indicated in the "ac" record. For each "ac" record, "Auxiliary Data Reference" points to the NDEF record which contain device engagement structure as defined in 8.1.1.

When negotiated handover is performed, the mDL shall anounce the "urn:nfc:sn:handover" service in the Service Parameter record in the initial NDEF message. The mDL Reader shall take the role of the Handover Requester; it shall send the Handover Request message to the mDL after the Service Select Message is sent. The Handover Request Message shall contain all available alternative carriers that are supported by mDL Reader. The mDL confirms the handover by returning a Handover Select Message containing exactly one selected alternative carrier.

Table 6 — Handover "ac" record

ac/connection method	Reference
NFC	8.1.2.2
BLE	NFC Forum, *Bluetooth Secure Simple Pairing Using NFC, NFCForum-AD-BTSSP_1_2, May 2019*
Wi-Fi Aware	Wi-Fi Alliance, *Neighbor Awareness Networking Technical Specification Addendum, Version 3.0.2, April 2019*

The DeviceEngagement structure shall be transferred is part of an auxiliary data record with the type "iso.org:18013:deviceengagement" and the ID reference "mDL". When Static Handover is used, the DeviceEngagement structure shall be present in the Handover Select message. When Negotiated Handover is used the Device Engagement structure shall be present in the Initial NDEF message or in the Handover Select Message or in both messages. If it is present in the Initital NDEF message and in the Handover Select Message, the Device Engagement structure sent in the Handover Select message is valid.

When device engagement using NFC is carried out, the TransferMethods array in the device engagement structure shall be empty.

8.1.2.2 mDL NFC alternative carrier record for NFC connection handover

Table 7 describes the ac record used by the NFC connection handover protocol to indicate support for NFC. The NFC "ac" record shall only be part of static handover.

NOTE "ac" records can be provided during negotiated handover that are not present in the static handover records.

Table 7 — Binary content of a NFC Handover Select message

Content	Length (Octets)	Description	Notes
0x91	1	NDEF Record Header: MB=1b, ME=0b, CF=0b, SR=1, IL=0b, TNF=001b	
0x02	1	Record Type Length: 2 octets	
0x10	1	Payload Length: 16 octets	
0x48, 0x73	2	Record Type: “Hs”, Handover Select Message	
0x15	1	NFC Connection Handover Specification 1.5	
0xD1	1	NDEF Record Header: MB=1b, ME=1b, CF=0b, SR=1, IL=0b, TNF=001b	Alternative Carrier Record, “nfc”
0x02	1	Record Type Length: 2 octets	
0x0A	1	Payload Length: 10	
0x61, 0x63	2	Record Type: “ac”, Alternative Carrier Record	
0x01	1	Carrier Flags: “active” (CPS=1)	
0x03	1	Carrier Data Reference Length: 3 octets	
0x6E, 0x66, 0x63	3	Carrier Data Reference: “nfc”	
0x01	1	Auxiliary Data Reference Count: 1	
0x03	1	Auxiliary Data Reference Length: 3 octets	
0x6D, 0x44, 0x4C	3	Auxiliary Data Reference Char: “mDL”	
0x1C	1	NDEF Record Header: MB=0b, ME=0b, CF=0b, SR=1, IL=1b, TNF=100b (NFC Forum External Type)	Carrier Data Record, “nfc”
0x0D	1	Record Type Length: 13 octets	
0x03	1	Payload Length: 3 octets	
0x03	1	ID Field Length: 3 octets	
0x69, 0x73, 0x6F, 0x2E, 0x6F, 0x72, 0x67, 0x3A, 0x31, 0x38, 0x30, 0x31, 0x33	13	Record Type: “iso.org:18013”, ISO/IEC 18013 mDL application	
0x6E, 0x66, 0x63	3	ID Field: “nfc”	
0x10	1	mDL NFC Connection Handover Version. Major Version: 1, Minor Version: 0	
0xYY or 0xYY 0xYY 0xYY 0xYY	2	Maximum length of command data field supported by mobile device, as defined in ISO/IEC 7816-4. NOTE: a value over 255 bytes indicates extended length. Indicated as an unsigned integer.	

Table 7 *(continued)*

Content	Length (Octets)	Description	Notes
0x5C	1	NDEF Record Header: MB=0b, ME=1b, CF=0b, SR=1b or 0b, IL=1b, TNF=100b (NFC Forum External Type). NOTE: the SR may be 0 depending on the size of the CBOR structure (if NDEF Payload size is greater than 255 bytes).	Auxiliary Data Record, "mDL"
0x1E	1	Record Type Length: 30 octets	
0xYY	1 or 4	Payload Length: YY octets. NOTE: this may be a 4 bytes content depending on the size of the CBOR structure (if NDEF Payload size is greater than 255 bytes).	
0x03	1	ID Field Length	
0x69, 0x73, 0x6F, 0x2E, 0x6F, 0x72, 0x67, 0x3A, 0x31, 0x38, 0x30, 0x31, 0x33, 0x3a,0x64, 0x65, 0x76, 0x69, 0x63, 0x65, 0x45, 0x6e, 0x67, 0x61, 0x67, 0x65, 0x6d, 0x65, 0x6e, 0x74	13	Record Type: "iso.org:18013:deviceengagement", ISO18013 mDL application	
0x6D, 0x44, 0x4C	3	ID Field: "mDL"	
…	YY	deviceEngagement binary data of CBOR structure	

8.1.2.3 Device engagement using QR code

Device engagement using QR code shall consist of a barcode compliant with ISO/IEC 18004 (QR code). The QR code is a URI with "mDL:" as scheme and the device engagement structure encoded using base64url-without-padding as path. Therefore the QR code is a concatenation of "mDL:" followed by the base64url-without-padding encoded device engagement structure.

An mDL Reader shall decide one of transmission technologies for data retrieval which is provided in the device engagement structure.

8.2 Data retrieval

8.2.1 Data retrieval methods

8.2.1.1 Offline retrieval

8.2.1.1.1 General

All messages in mDL Reader Request and mDL Response shall be encoded with an encoding of CBOR. These messages shall be formatted according to 8.2.1.1.2.1. The resulting mDL Reader Request and mDL Response CBOR byte string, shall be encrypted in accordance with 9.2.1 and transmitted over the selected transport method.

The identifier and format of the data elements are described in Table 3. Domestic data elements (see 7.4.9) should not be returned unless requested by the mDL Reader.

8.2.1.1.2 Message structure

8.2.1.1.2.1 mDL Reader Request

The mDL Reader Request shall have the following CDDL structure:

```
OfflineRequest = {
    "version" : tstr,                   ; Version of the structure, currently 1.0
    "docRequests" : [+ DocRequest]      ; Requested DocType, NameSpace and data elements
}

DocRequest = {
    "itemsRequest" : ItemsRequestBytes,
    ? "readerAuth" : ReaderAuth
}

ItemsRequestBytes = #6.24(bstr .cbor ItemsRequest)

ReaderAuth = COSE_Sign1
; Use a nil value (null value defined in RFC 8152) for the payload of ReaderAuth
; the detached content is the ReaderAuthentication structure, see 9.2.4.4

ItemsRequest = {
    ? "docType" : DocType,
    "nameSpaces" : NameSpaces,
    ? "requestInfo" : {* tstr => any}   ; Additional info the reader wants to provide
}

DocType = tstr                          ; See 7.3.2

NameSpaces = {
    + NameSpace => DataElements         ; Requested data elements for each NameSpace
}

NameSpace = tstr                        ; See 7.3.3

DataElements = {
   + DataElement => IntentToRetain
; Requested data elements with Intent to Retain value for each requested element

DataElement = tstr                      ; See Table 3
IntentToRetain = bool                   ; See definition below
```

The IntentToRetain variable indicates that the mDL Verifier intends to retain the received data element. A value of false means the mDL Verifier shall not retain the data. A value of true means that the mDL Verifier intends to retain the data.

If an mDL Reader prefers to use online retrieval, the mDL Reader shall include the online token and url data element as a part of the mDL Reader Request.

The informative example of an mDL Reader Request can be found in D.4.1.1.

8.2.1.1.2.2 mDL Response

The mDL Response shall have the following CDDL structure:

```
OfflineResponse = {
    "version" : tstr,               ; Version of the structure, currently 1.0
    ? "documents" : [+Documents],   ; Returned documents
    "status" : uint                 ; Status codes according to Table 8
}

Documents = {
    + DocType => ResponseData
}

DocType = tstr                          ; See 7.3.2

ResponseData = {
    "issuerSigned" : IssuerSigned,  ; Responded data elements signed by the issuer
    "deviceSigned" : DeviceSigned,  ; Responded data elements signed by the mDL
    ? "errors" : Errors
}

IssuerSigned = {
```

```
    ? "nameSpaces" : IssuerNameSpaces,
    "issuerAuth" : IssuerAuth
}

IssuerAuth = COSE_Sign1     ; The payload is the MobileSecurityObject, seesee 9.2.2.4


IssuerNameSpaces = {
    + NameSpace => [ + IssuerSignedItemBytes ]
}

IssuerSignedItemBytes = #6.24(bstr .cbor IssuerSignedItem)

NameSpace = tstr                        ; See 7.3.3

IssuerSignedItem = {
    "digestID" : uint               ; DigestID according to 9.2.2.5
    "random" : bstr                 ; Random value according to 9.2.2.5
    "elementIdentifier" : tstr      ; Data element identifier according to Table 3
    "elementValue" : any            ; Data element value according to Table 3
}

DeviceSigned = {
    "nameSpaces" : DeviceNameSpacesBytes,
    "deviceAuth" : DeviceAuth
}

DeviceNameSpacesBytes = #6.24(bstr .cbor DeviceNameSpaces)

DeviceNameSpaces = {
    * NameSpace => DeviceSignedItems
}

DeviceSignedItems = {
    + DataItemName => DataItemValue
}

DataItemName = tstr                     ; See Table 3
DataItemValue = any                     ; See Table 3

DeviceAuth = {
    "deviceSignature" : DeviceSignature, //     ; NOTE: // means or
    "deviceMac" : DeviceMac
}

; For the following structures, use a nil value for the payload. The detached
; content is the DeviceAuthentication structure, see 9.2.2.4
DeviceSignature = COSE_Sign1
DeviceMac = COSE_Mac0

Errors = {
    * NameSpace => [ + ErrorItem ]
}

ErrorItem = {
    DataItemName => ErrorCode         ; See Table 9
}

ErrorCode = uint                       ; Error codes according to Table 9
```

Clause 8.2.1.1.2.3 contain the status and error codes for use in the mDL Response.

The informative example of mDL Response can be found in D.4.1.2

8.2.1.1.2.3 mDL Response status and error codes

The status codes in Table 8 shall be included as part of “Response”.

Table 8 — Response status

Status code	Status message	Description	Actions required
0	OK	This status message shall be returned if no other status is returned	No specific action required
10	General Error	The mDL returns an error without any given reason	The mDL Reader may inspect the problem. The mDL Reader may continue the transaction.
20	mDL Reader authentication Error	The mDL indicates there is an error with mDL Reader authentication (see 9.2.4)	The mDL Reader may inspect the problem. The mDL Reader may continue the transaction.
21	Request Rejected	The mDL indicates that the request is rejected	The mDL Reader may inspect the problem. The mDL Reader may continue the transaction.

The error codes in Table 9 are specific to the data element requested within a nameSpace and if returned shall be returned individually as part of "ResponseData". Returning an error code from Table 9 is optional.

Table 9 — Data handling error

Error code	Error code message	Description
0	Data Not Returned	The mDL does not provide the requested data element without any given reason, this element may be used in all cases.
1	Invalid Format	The mDL cannot process the requested data element due to formatting error
2	Data Not Found	The requested nameSpace or data element within a nameSpace is not found
3	Data Request Denied	The release of requested data element was rejected by the mDL Holder

8.2.1.2 Online retrieval

8.2.1.2.1 General

Data retrieval using online retrieval shall make use of WebAPI or OIDC. 8.2.1.2.2 explains the message structure of online retrieval and 8.2.1.2.3 explains how WebAPI and OIDC shall be implemented. All data protection mechanisms for online retrieval (TLS and JWS) are described in 9.3.

The mDL is identified by the IA using the token as provided by the mDL Reader. The information encoded in the token is outside of the scope of this document.

NOTE The token can contain information about which data is to be provided to the mDL Reader. The IA can use a separate interface with the mDL to retrieve information about which data is to be provided to the mDL Reader.

NOTE The issuing authority is present in each transaction in the on-line solution; therefore, the issuing authority knows when an mDL is used and what data is shared. If tracking is a concern, the issuing authority is advised to implement mitigating strategies to ensure the mDL and the mDL Holder are not tracked.

8.2.1.2.2 Message structure

Online retrieval uses a JSON Web Token (JWT) structure as defined in RFC 7519. The identifier and format of the data elements (called 'claims' in a JWT) are described in Table 3. Additional domestic data elements (see 7.4.9) are supported through the use of namespaces. Domestic data elements should not be returned unless requested by the mDL Reader.

Whenever any mDL data is sent to an mDL Reader, the following claims shall be present in the JWT: exp (Expiration Time) and iat (Issued AT) as defined in RFC 7519 as well as all information necessary to

verify that the person presenting the mDL is in fact the mDL Holder shall be made available to the mDL Reader. This can be done by providing the portrait of the mDL Holder.

8.2.1.2.3 Implementations

8.2.1.2.3.1 WebAPI

Issuer URL refers to the base server URL of an "`/identity`" WebAPI endpoint. The request will use a JSON object which contains the requested items. The Content-Type header-field shall be set to "`application/json`". The host field content shall be derived from the Issuer URL element. The request method is POST.

The following CDDL structure defines the request structure.

NOTE The request structure is a JSON object.

```
OnlineRequest = {
    "version" : tstr,                 ; Version of the structure, currently 1.0
    "token" : tstr,                   ; Token
    ? "docType" = tstr,               ; Optional doctype field, see 7.3.2
    "nameSpaces" : NameSpaces         ; See 7.3.3
}

NameSpaces = {
    + NameSpace => DataElements       ; Requested data elements for each NameSpace
}

NameSpace = tstr

DataElements = {
    + DataElement => IntentToRetain   ; Requested data elements with Intent to Retain
                                      ; value for each requested element

DataElement = tstr                    ; See Table 3
IntentToRetain = bool                 ; See definition below
```

The request "`token`" attribute contains the full token which identifies the mDL. The DocType is an optional field that describes the DocType requested according to 7.3.2. The requested claims are always part of a namespace according to 7.3.3. A request may contain additional namespaces besides the namespaces defined in this document. The IA shall ignore all unknown claims in the request when processing the request. The informative example can be found in D.4.2.1.1.

The IntentToRetain variable indicates that the mDL Verifier intends to retain the received data element. A value of false means the mDL Verifier shall not retain the data. A value of true means that the mDL Verifier intends to retain the data.

A successful response contains the HTTP status "200 OK", the "`Content-Type`" header is set to "`application/jwt`", and the "`Content-Length`" header is set correctly. In the response body, a JWT structure is included which contains the released attributes.

The following CDDL structure defines the response structure.

NOTE The response structure is a JSON object.

```
OnlineResponse = {
    "version" : tstr,           ; Version of the structure, currently 1.0
    "docType" : tstr,           ; DocType field, see 7.3.2
    "nameSpaces" : NameSpacesResponse,  ; See 7.3.3
    * tstr => any               ; Shall include at least exp and iat
}

NameSpacesResponse = {
    + NameSpace => DataElementsValues ; Requested data elements for each NameSpace
}

DataElementsValues = {
    + ElementIdentifier => ElementValue
```

```
}

ElementIdentifier = tstr          ; See Table 3
ElementValue = any                ; See Table 3
NameSpace = tstr
```

The informative example can be found in D.4.2.1.2.

In case the IA requires interaction with the mDL Holder, the IA would require user input to negotiate data to be shared with the mDL after the request has been received. The mDL Reader has to wait for the result using long or short polling. In case of long polling it is recommended to set the timeout to 120 seconds in order to avoid requests to timeout. Another option is to send a HTTP 202 response, including a retry interval in the header in milli-second with an empty response body. The mDL Reader has to periodically check for the final response.

Table 10 contains all the allowed responses.

Table 10 — HTTP status codes

HTTP status code	HTTP status message	Description
200	OK	Successful HTTP request.
202	Accepted	The HTTP has been accepted for processing but is not yet completed.
400	Bad Request	The HTTP request was invalid or malformed.
403	Forbidden	The mDL Reader is denied to access of any data either via explicit user consent, configured policies, pre-selection or other means.
401	Unauthorized	The provided token was invalid.
500	Internal Server Error	The server encountered an internal server error and was not able to process the request successfully.

8.2.1.2.3.2 OIDC

Issuer URL refers to the base server URL address of the issuing authority Open ID provider. The mDL Reader should use the "`.well-known/openid-configuration`" endpoint as described in the section 4, OpenID Connect Discovery, *N. Sakimura et. al., Defines how clients/readers dynamically discover information about OpenID Providers, November 2014*, to retrieve the further endpoints to be used during the OIDC data retrieval method described in 8.2.2.5.

In order to use the concept of namespaces within the OIDC framework, the following convention for naming claims shall be used. Each element as defined in 7.4 shall get the namespace as defined in 7.3.3 as a prefix as [namespace]:[identifier] An example of this is "org.iso.18013.5.1:portrait".

NOTE OIDC currently does not support the intentToRetain specified in 8.2.1.1.2.2. The capability to use that is expected in a future version of this document.

As part of the response, three items are mandatory; the exp and iat elements as defined in 8.2.1.2.2 and a DocType element, for which the value is defined in 7.3.2.

The mDL Reader may take the action of registering with the issuing authority and calling the issuing authority's authorize endpoint with the full token as input to the "login_hint" parameter as specified in OpenID Connect Core 1.0.

8.2.2 Data retrieval transmission technologies

8.2.2.1 Data retrieval using Bluetooth Low Energy (BLE)

8.2.2.1.1 General

Bluetooth low energy (BLE) may be used for transferring the data between the mDL and mDL Reader. Using BLE as a transmission technology consists of two phases, connection setup and data retrieval. During connection setup, the mDL and mDL Reader connect to each other. After the connection is setup, data retrieval can be initiated. Security is ensured by the application layer, use of BLE-specific security may be used. All BLE session information shall be removed after each transaction.

As part of device engagement, an mDL indicates whether it will act in BLE central mode (scanning), BLE peripheral mode (advertising) or both. An mDL Reader shall support both modes. When the mDL supports both modes, the mDL Reader should act as BLE central mode.

NOTE It is possible for the mDL or mDL Reader to act in both modes, until data retrieval has started.

If the mDL acts as BLE central mode, it shall act as a GATT client. This mode is therefore called mDL Central Client mode. If the mDL acts as BLE peripheral mode, it shall act as a GATT server. This mode is therefore called mDL Peripheral Server mode.

8.2.2.1.2 Device identification

After performing device engagement, the mDL and mDL Reader will start connection setup. As part of the connection setup, the scanning device has to identify the correct advertising device. Methods are described in how this can be performed in a multi-advertiser environment.

As part of the BLE advertisement process, the service UUID and device name shall be advertised according to the BLE specification. The device name advertised can be unique. To facilitate the identification of devices, multiple advertising and scanning options exist.

NOTE 1 BLE stacks in mobile devices might use scan filter and caching methods to manage congested environments and manage scan intervals for device energy consumption control. This might advertently influence the connection time required when using ephemeral UUIDs.

The methods described for device authentication in this clause are only applicable when the QR code is used for Device Engagement. When NFC is used, NFC Forum, *Bluetooth Secure Simple Pairing Using NFC, NFCForum-AD-BTSSP_1_2, May 2019* describes how the device authentication and connection is performed.

For mDL Peripheral Server mode the mDL shall advertise one of the following UUIDs a 16 byte UUID individual to the mDL, transferred to the mDL Reader during Device Engagement. The mDL Reader shall scan for the UUID transferred during Device Engagement.

For mDL Central Client Mode the mDL Reader shall advertise a 16 byte UUID individual to the reader if it is transferred during Device Engagement from the mDL to the mDL Reader, otherwise the mDL Reader shall advertise the BT SIG registered UUID for the mDL Reader. The mDL shall scan for the applicable UUID.

Currently no BT SIG registered UUID is available, but it is expected to be available in the future. Until that time the following UUID shall be used: 5c8256b5-225f-45e6-a102-f9307a4d30c4.

When Central Client Mode is used and no UUID is transferred during Device Engagement, the mDL has to connect to the correct mDL Reader. In an environment with multiple mDL Readers present, multiple mDL Readers might adertise the same UUID, in this case other methods have to be used to ensure connection to the correct device. One such method is described in NOTE3, but other methods may also be used. To ensure that Data retrieval only starts when the correct devices are connected, the Ident characteristic as described in 8.2.2.1.5 shall be used to verify that the correct devices ares connected, when a non-unique UUID is used. Verifying the Ident characteristic requires an actual BLE connection

to be setup and takes a lot of time. It should therefore not be used as the primary means of verifying the correct mDL Reader to connect to.

NOTE 2 Finding the correct device to connect to is a purely a practical problem. Identifying, connecting or transferring data to the wrong device causes no security risks because of the security methods described in Clause 9. Note that these mechanisms do not provide complete protection against a bad actor aiming to cause a denial of service attack.

NOTE 3 BLE allows to transmit the transmitter signal strength while advertising and to determine the received signal strength at the receiving (scanning) side. Due to high UHF channel characteristics a momentarily received signal strength (RSSI – received signal strength indication) might vary. A good RSSI value can be achieved by building a mean value over a few scan intervals. Typical scan intervals on mobile devices can be in the range of 100 ms although the BLE specification allows for much shorter intervals. The distance between advertiser and scanner cannot be exactly determined but well approximated estimates can be done based on the absolute RSSI value together with its change over a shorter period of time (a few scan intervals). In a multi-advertiser environment disambiguation can be done by also comparing other advertiser RSSI values and their mean value trend.

The CBOR structure used to transfer the BLE identification information as part of the TransferMethod structure is defined in 8.1.1.1. UUID's transferred shall be encoded using variant 1 (RFC 4122)

8.2.2.1.3 Service Definition

When the mDL is acting in BLE peripheral server mode, the mDL shall act as a peripheral and GATT server. When the mDL is acting in BLE central client mode, the mDL Reader shall act as peripheral and GATT server.

Table 11 shows characteristics which the mDL service shall expose.

Table 11 — mDL service characteristics

Characteristic Name	UUID	Mandatory properties
State	00000001-A123-48CE-896B-4C76973373E6	PROPERTY_NOTIFY, PROPERTY_WRITE_NO_RESPONSE
Client2Server	00000002- A123-48CE-896B-4C76973373E6	PROPERTY_WRITE_NO_RESPONSE
Server2Client	00000003- A123-48CE-896B-4C76973373E6	PROPERTY_NOTIFY

Table 12 shows characteristics which the mDL Reader service shall expose.

Table 12 — mDL Reader service characteristics

Characteristic Name	UUID	Mandatory properties
State	00000005- A123-48CE-896B-4C76973373E6	PROPERTY_NOTIFY, PROPERTY_WRITE_NO_RESPONSE
Client2Server	00000006- A123-48CE-896B-4C76973373E6	PROPERTY_WRITE_NO_REPSONSE
Server2Client	00000007- A123-48CE-896B-4C76973373E6	PROPERTY_NOTIFY
Ident	00000008- A123-48CE-896B-4C76973373E6	PROPERTY_READ

Each service characteristic that has the PROPERTY_NOTIFY property shall contain the client_characteristic_configuration discriptor, with UUID 0x2902 and default value of 0x0000. This value shall be set to 0x0001 by the client to get notified for the characteristic associated to this descriptor.

8.2.2.1.4 Connection setup

The BLE peripheral broadcasts the service with the UUID as defined in 8.2.2.1.2 in the advertising packet. The BLE central is then able to connect to the advertised service

The BLE central scans for devices advertising a Service with the defined UUID. If mDL central client mode is used and the BT SIG UUID is used, the mDL shall read the ident characteristic to confirm the correct BLE peripheral by varifying the value in the Ident characteristic.

After the connection is setup the GATT client subscribes to notifications of char of characteristic 'State' and 'Server2Client'. For performance reasons, the GATT client should request for an as high an MTU as possible. After these steps, the GATT client makes a write without response request to 'state' where it sets the value to 0x01. This tells the GATT server that the GATT client is ready for the transmission to start.

8.2.2.1.5 Connection state and Ident

The connection state is indicated by the 'state' characteristic. It is encoded as 1 byte binary data. Table 13 describes the different connection state values, which are communicated using write without response and notify.

Table 13 — Connection state values

Command	Data	Sender	Description
Start	0x01	GATT client	This indicates that the mDL Reader may/will begin transmission.
End	0x02	mDL, mDL Reader	Signal to finish/terminate transaction. The mDL Reader shall use this value to signal the end of data retrieval. Both the mDL and the mDL Reader can use this value at any time to terminate the connection.

The Ident characteristic shall contain the SHA-256 hash as binary data of the mDL public ephemeral key (EDeviceKeyBytes) as transferred during Device Engagement.

8.2.2.1.6 Data retrieval

Data retrieval starts by signaling the 'Start' value to the 'state' characteristic.

If the GATT client wants to send a message to the GATT server, it cuts the message in pieces with a length of 3 bytes less than the MTU size. It then sends these pieces to the GATT server using the write without response command via the 'Client2Server' characteristic. The first byte of the message is either 0x01 which indicates more messages are coming, or 0x00 to indicate it's the last part of the message.

If the GATT server wants to send a message to the GATT client, it cuts the message in pieces with a length of 3 bytes less than the MTU size. It then sends these pieces to the GATT client using the notify command via the 'Server2Client' characteristic. The first byte of the message is either 0x01 which indicates more messages are coming, or 0x00 to indicate it's the last part of the message.

Sequence of messages can repeat as long as necessary to finish data retrieval.

Figure 4 describes the schema for data retrieval in mDL Peripheral Server Mode.

Figure 5 describes the schema for data retrieval in mDL Central Client Mode.

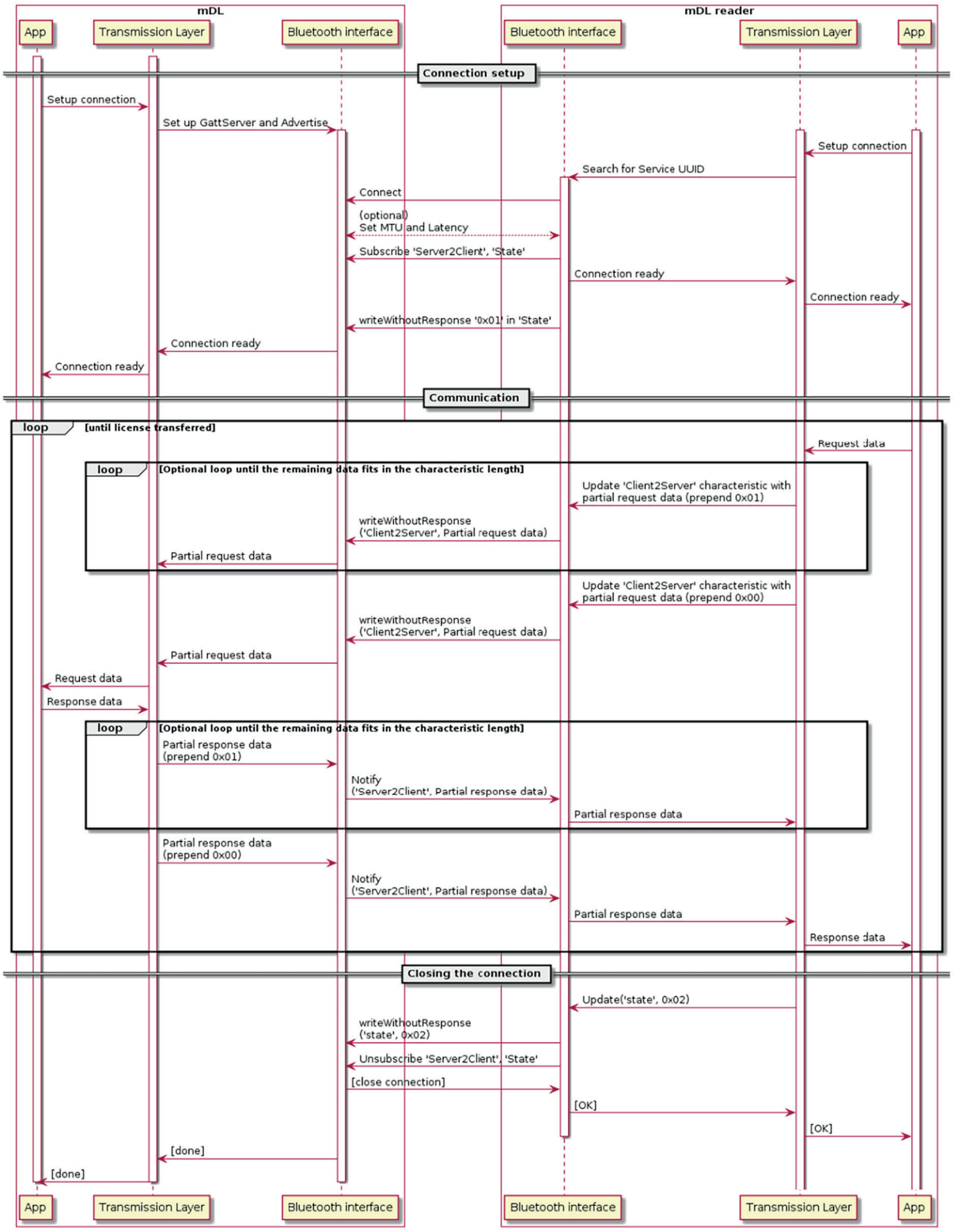

Figure 4 — mDL Peripheral Server Mode schema

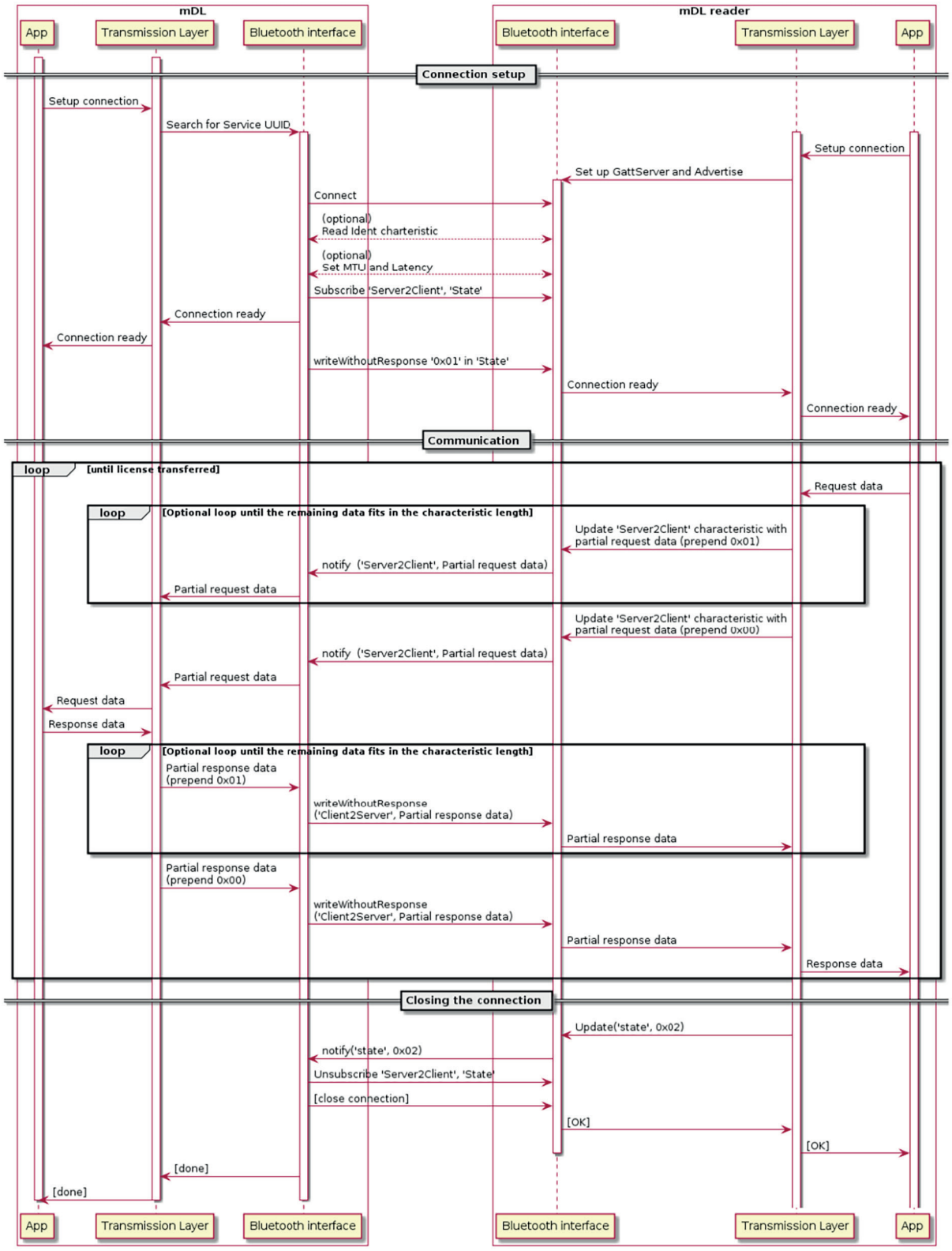

Figure 5 — mDL Central Client Mode schema

8.2.2.1.7 Connection closure

After data retrieval, the GATT client unsubscribes from both the 'State' and 'Server2Client' characteristics and disconnects from the peripheral.

8.2.2.1.8 Connection re-establishment

In case of a lost connection before the ‘state’ characteristic has been set to a value of 0x01 (e.g. the transmission has not yet started), the mDL and mDL Reader should terminate their current BLE connection interface and try to reconnect according to 8.2.2.1.4.

In case of a lost connection after the 'state' characteristic has been set to value 0x01 (e.g. the transmission of data has started), a connection shall not be re-established and a completely new mDL transaction shall be initiated if required.

8.2.2.2 Data retrieval using Near Field communication (NFC)

NFC may be used for data retrieval. The mDL shall support PICC mode and the mDL Reader shall support PCD mode. These modes are defined in ISO/IEC 14443. Extended length fields as specified in ISO/IEC 7816-4 should be supported by an mDL and mDL Reader.

The Application IDentifier (AID) of the mDL shall be: 'A0 00 00 02 48 04 00'.

The AID of the mDL consists of registered application provider identifier (RID) ('A0 00 00 02 48') followed by the proprietary application identifier extension (PIX) (’04 00’). An mDL application shall be selected using SELECT command defined in ISO/IEC 7816-4 with the AID listed above.

Table 14 and Table 15 specify SELECT command and response.

Table 14 — SELECT command

CLA	INS	P1-P2	Lc field	Data field	Le field
As defined in 5.4.1 in ISO/IEC 7816-4	‘A4’	’04 0C’	‘07’	‘A0 00 00 02 48 04 00’	Absent

Table 15 — SELECT response

Data field	SW1 – SW2
Absent	See Table 5 and Table 6 in ISO/IEC 7816-4

After the mDL application is select, the mDL Reader can start data retrieval. The ENVELOPE command shall be used by the mDL Reader to exchange the mDL Reader Request.

The ENVELOPE command (with INS=’C3’) is a transmission command that serves to transfer a payload that shall be a data object. Odd INS (INS=’C3’) indicates that payload (data field) shall be encoded in BER-TLV data object. For data retrieval, data field is DO’53’ encapsulating an encrypted CBOR blob.

The ENVELOPE response returning the mDL Response CBOR blob shall be nested in a DO'53' as well.

Table 16 and Table 17 specify ENVELOPE command and response.

Table 16 — ENVELOPE command

CLA	INS	P1-P2	Lc field	Data field	Le field
As defined in 5.4.1 in ISO/IEC 7816-4	‘C3’	‘0000’ (any other value is RFU)	Absent for Nc = 0, or present for Nc > 0	Absent or DO or DO fragment	Absent for encoding Ne = 0, present for encoding Ne > 0

Table 17 — ENVELOPE response

Data field	SW1 – SW2
DO'53' or absent	See Table 5 and Table 6 in ISO/IEC 7816-4

For oversize incoming payload (from an mDL Reader to an mDL), several ENVELOPE commands shall be chained (CLA with bit5 set to 1 except for the last command of the chain) and the command data field of the last command of the chain shall be absent to indicate “end of data string” according to 12.2.7 in ISO/IEC 7816-3.

For oversize outgoing payload (from an mDL to an mDL Reader), GET RESPONSE command, defined in ISO/IEC 7816-4, shall be used.

Table 18 and Table 19 specify GET RESPONSE command and response.

Table 18 — GET RESPONSE command

CLA	INS	P1-P2	Lc field	Data field	Le field
As defined in 5.4.1 in ISO/IEC 7816-4	‘C0’	’00 00’	Absent	Absent	Present for encoding Ne > 0

Table 19 — GET RESPONSE response

Data field	SW1 – SW2
Encrypted mDL Response or absent	See Table 5 and Table 6 in ISO/IEC 7816-4

The maximum supported command size is indicated by the mDL during device engagement.

When extended length is used, if an mDL returns '6100' in the SW1 - SW2, this shall be interpreted as that there are more than 255 bytes available for retrieval. An mDL Reader should subsequently send a GET RESPONSE command with a LE field of ’00’. This indicates to the mDL, that the mDL can respond with as many bytes as it can. In case the reader doesn’t support the size of the incoming message, is has to send a GET RESPONSE with Le different from ’00’, indicating that the mDL shall not use extended length for the response.

If the NFC connection is lost during data retrieval, a completely new mDL transaction (including Device Engagement) shall be initiated.

8.2.2.3 Data retrieval using Wi-Fi Aware

8.2.2.3.1 General

Wi-Fi Aware may be used for data retrieval.

Wi-Fi Aware as specified in Wi-Fi Alliance, *Neighbor Awareness Networking Technical Specification, Version 3.0, December 2018* and Wi-Fi Alliance, *Neighbor Awareness Networking Specification v3.0 draft Addendum version 0.0.2, April 2019* may be used to transfer mDL data. The data retrieval using Wi-Fi Aware consists of 3 phases, connection setup, data retrieval and closure.

8.2.2.3.2 Connection setup

Wi-Fi aware can be setup using static or negotiated handover methods. The static handover operation shall be used when QR-code or NFC with static handover is used during the device engagement phase. The negotiated handover operation shall be used when NFC with negotiated handover is used during the device engagement phase.

The service name shall be calculated per transaction using HKDF instantiated with SHA-256 according to RFC 5869. The inputs are, IKM: mDL ephemeral public key (EDeviceKeyBytes), salt: 0x01. The resulting length shall be 16 bytes. The resuling bytestring shall be converted to a text string using capital letters and no spaces, e.g. “94AB45CDBDEF675162183B12AC35EFAA”.

Use of a cipher suite for security is mandatory. The mDL Reader shall support The NCS-SK-128 and NCS-PK-2WDH-128 cipher suites, as indicated in table 1 of Wi-Fi Alliance, *Neighbor Awareness Networking*

Specification v3.0 draft Addendum version 0.0.2, April 2019. The mDL shall support at least one of the NCS-SK-128 and NCS-PK-2WDH-128. Use of the NCS-PK-2WDH-128 cipher suite is recommended.

When QR-code for Device Engagement is used and the password is not transferred, a passphrase shall be calculated using HKDF instantiated with SHA-256 according to RFC 5869. The inputs are, IKM: mDL ephemeral public key (EDeviceKeyBytes), salt: 0x02. The resulting length shall be 32 bytes. These bytes are then encoded using base64url-without-padding to get the passphrase. When NFC is used for Device Engagement, either the Password info or the DH info shall be explicitly transferred from the mDL to the mDL Reader during device engagement according to Wi-Fi Alliance, *Neighbor Awareness Networking Specification v3.0 draft Addendum version 0.0.2, April 2019*.

During the Wi-Fi Aware service discovery procedure, the mDL shall serve as the Service Publisher, and the mDL Reader shall serve as the Service Subscriber. Once the Wi-Fi Aware service discovery is completed, the mDL Reader shall initiate the data path setup, and serve as the NDP Initiator; while the mDL shall serve as the NDP Responder.

8.2.2.3.3 Data retrieval

mDL data is transferred using the HTTP protocol, with the mDL serving as the HTTP and TCP servers, and the mDL Reader serving as the HTTP and TCP client. A unique local IPv6 address should be used for each connection. The data retrieval shall use the following minimal set of HTTP commands to transfer mDL data. HTTP request messages shall have the following structure:

```
POST /mDL HTTP/1.1
Host: [IPv6 address of the mDL]
Content-Length: [CBOR Request Length]
Content-Type: application/CBOR

[CBOR encoded mDL Reader Request]
```

HTTP response message shall have the following structure:

```
HTTP/1.1 200 OK
Content-length: [CBOR Response Length]
Content-type: application/CBOR

[CBOR encoded mDL Response]
```

Error responses are defined in 6.1.1, RFC 2616.

8.2.2.3.4 Closure

After the last response message, the connection is closed by the mDL Reader.

8.2.2.4 Data retrieval using WebAPI

The WebAPI data retrieval method may be used for mDL transfer. There is no specific information required regarding transmission technology for data retrieval using WebAPI.

8.2.2.5 Data retrieval using OpenID Connect ('oidc')

8.2.2.5.1 General

The OIDC data retrieval method may be used for mDL transfer. The corresponding interface is defined in 8.2.1.2 and by OpenID Connect Core 1.0, *N. Sakimura et. Al., Defines the core OpenID Connect functionality: authentication built on top of Oauth 2.0 and the use of claims to communicate information about the End-User, November 2014*.

The mDL Reader shall use and the issuing authority shall accept the Authorization Code Flow Grant. The mDL Reader is considered a client of the issuing authority OpenID Provider (OP).

8.2.2.5.2 Connection Setup

The mDL Reader may pre-register with the issuing authority through their client registration process to obtain a client_id.

The mDL Reader may use dynamic client registration as specified in OpenID Connect Dynamic Registration, *N. Sakimura et. Al., Defines how clients/readers dynamically register with OpenID Providers, November 2014*.

In either registration process, the mDL Reader shall provide the same redirect_uri during registration as will be used during authorization, and the Reader shall provide their jwks keyset.

8.2.2.5.3 Data retrieval

The issuing authority OP may consent the user before sharing claims or scopes with the mDL Reader client. The mechanism for this consent is out of scope of this document. Additional data fields of the JWT as per OpenID Connect Core or regional profiles of OpenID Connect are permissible, and the mDL Reader should skip over JWT fields that it does not need in the Core, mDL, or domestic namespaces.

8.2.2.5.4 Closure

The issuing authority OpenID Provider should support OpenID Connect Back-channel Logout within a short time duration to prevent open sessions with the mDL Reader from being used for requesting additional attributes.

9 Security mechanisms

9.1 Overview

The security of mDL data exchanged with an mDL Reader should preserve the triad of confidentiality, integrity and authenticity by design and by default.

The security architecture aims to achieve four distinct goals:

a) Protection against forgery: Data elements are signed by the issuing authority (IA). The degree of protection against forgery depends on the degree to which the IA's keys are protected.

b) Protection against cloning: The mDL produces a signature or message authentication code over session data. The key used to authenticate the session data is stored only in the mDL and in turn is signed by the IA. The degree of protoection against cloning depends on the degree to which the mDL authentication key or the TLS server key is protected.

c) Protection against eavesdropping: Communications between mDL and mDL Reader are encrypted and authenticated. Device Engagement uses a separate communication channel to mitigate the risk of Man in the Middle (MITM) attacks. In addition, the reader can detect MITM attacks by validating the anti-cloning signature or message authentication code, which is created using a key for which the public part is signed by the IA in the MSO. If mDL Reader authentication is used, the mDL can detect MITM attacks before returning any data. Online retrieval uses TLS for encryption to further protect against eavesdropping and MITM attacks.

d) Protection against unauthorized access: The ability for the mDL Reader to send a request message is only authorized to the entity that participates in the device engagement. The encryption key for communications between the mDL and mDL Reader is derived from an ephemeral key pair from both the mDL and mDL Reader. The public key of the mDL is shared only through device engagement. Since device engagement is only short-range this protects against unauthorized access. Further access control methods can be used after receiving the request message from the mDL Reader.

Revocation of an mDL is out of scope for this document. However, the MSO includes update information and validity time frames which enable the mDL Reader to check the freshness of the data. The issuing authority shall define appropriate periods of validity that balance freshness with offline capability, taking into account that a shorter validity period mitigates certain security risks.

Table 20 describes the security mechanisms that can be implemented for each data retrieval interface. For offline retrieval, issuer data authentication, session encryption and mDL authentication shall be implemented. mDL Reader authentication is optional for both offline retrieval and online retrieval. For online retrieval, Transport Layer Security (TLS), 1.2 or higher, and JSON Web Signature (JWS) shall be implemented.

When online retrieval is used, it is recommended that one-time use tokens are used.

An mDL Reader needs access to the issuing authority certificate authority (IACA) root certificate to verify issuer data authentication, verify the JWT and perform TLS. One optional method to get access to these certificates is described in Annex C.

Table 20 — Data protection mechanisms

Data retrieval method	Security mechanisms	Reference
Offline retrieval	Session encryption	9.2.1
	Issuer data authentication	9.2.2
	mDL authentication	9.2.3
	mDL Reader authentication	9.2.4
Online retrieval	TLS	9.3.1
	JWS	9.3.2
	mDL authentication [a]	9.2.3
[a] only applicable when the online token is transferred using offline retrieval.		

Table 21 describes security functionalities for different data retrieval methods.

Table 21 — Security functionality

Functionality	Offline retrieval	Online retrieval
Protect against cloning of mDL/ binding mDL data to a specific device	mDL authentication	mDL authentication [a]
Authenticate the origin of mDL data	Issuer data authentication	JWS
Verify mDL data has not changed from issuing authority	Issuer data authentication	JWS
Preserve confidentiality of mDL data	Session encryption	TLS
Verify how up to date the mDL data is	Issuer data authentication	JWS
Prevent unnoticed alteration of communication	Session encryption mDL authentication	TLS
Prevent unauthorized access of mDL data	Close-range device engagement with session encryption	Close-range device engagement (with session encryption)[a]
[a] only applicable when the online token is transferred using offline retrieval.		

See Annex E for additional information on privacy and security.

9.2 Offline retrieval

9.2.1 Session encryption

9.2.1.1 Purpose

Encrypting with authentication of the mDL Reader Requests and mDL Responses with the session key protects mDL data from eavesdropping and alteration.

9.2.1.2 Applicability

This mechanism is applicable for an mDL using the offline retrieval.

9.2.1.3 Description

Session encryption uses standard ephemeral key ECDH to establish session keys for authenticated symmetric encryption.

9.2.1.4 Mechanism

The following steps are performed in session encryption:

— Step 1, Device Engagement. The mDL generates an ephemeral key pair (EDeviceKey.Priv, EDeviceKey.Pub), and includes the cipher suite identifier as part of the device engagement structure as defined in 8.1.1.

— Step 2, Session establishment. The mDL Reader generates its ephemeral key pair (EReaderKey.Priv, EReaderKey.Pub). Two session keys (SKReader, SKDevice) are derived independently by the mDL and the mDL Reader, and used to encrypt and decrypt messages during the remainder of the session. To compute the session keys (SKReader, SKDevice), the mDL uses KDF(ECDH(EDeviceKey.Priv, EReaderKey.Pub)) and the mDL Reader uses KDF(ECDH(EDeviceKey.Pub, EReaderKey.Priv)). The mDL Reader encrypts the first mDL Reader Request with SKReader and sends it to the mDL.

— Step 3...n, Session data. The mDL replies with the mDL Response encrypted with SKDevice. The mDL Reader and mDL may optionally exchange additional messages. Each message is encrypted by the mDL and mDL Reader using their respective session keys.

The mDL and mDL Reader ephemeral keys are encoded in COSE_Key structures according to as defined in RFC 8152. The uncompressed form shall be used, except for transfer in the Device Engagement structure, where both the uncompressed and the compressed form may be used.

The session establishment message uses the following CBOR message structure:

```
SessionEstablishment = {
     "eReaderKey" : EReaderKeyBytes, ; EReaderKey.Pub
     "data" : bstr                   ; Encrypt(SKReader, mDL Reader Request)
}

EReaderKeyBytes = #6.24(bstr .cbor COSE_Key) ; containing the EReaderKey.Pub
```

The session data messages use the following CDDL message structure:

```
SessionData = {
    "data" : bstr, //         ; Encrypt(SKReader or SKDevice, nonce, message data),
                              ; NOTE: // means or in CDDL
    "error" : uint            ; Error code
}
```

When there is an error in the session encryption, the error code (10) is returned without any data.

Table 22 shows the different methods used for the cryptographic operations. Currently only one set of cipher suite is defined, which is identified by the value '1':

— For ECDH, ECKA-DH (Elliptic Curve Key Agreement Algorithm – Diffie-Hellman) according to BSI TR-03111 shall be used. The output of this function is the shared secret value Zab.

— The key derivation shall use HKDF instantiated with SHA-256 as defined in RFC 5869. The info parameter shall be empty, the output key length is 256 bits. Two keys shall be derived, SKReader shall be derived using a salt of 0x00, SKDevice shall be derived using a salt of 0x01.

— For encryption AES-256-GCM (GCM: Galois Counter Mode) shall be used. The mDL Reader shall encrypt its messages with SKReader, the mDL shall encrypt its messages with SKDevice. Therefore, both the mDL and the mDL Reader need to generate both session keys in order to be able to also decrypt the messages they receive. The nonce used for encryption shall be built up according to the following structure: identifier | counter. The identifier is an 8 byte value. The mDL Reader shall use the following identifier: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00. The mDL shall use the following identifier: 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x01. Each session key has its own counter value. The counter value is an unsigned integer. The first encryption with a key shall use a counter value of 1. For each following encryption the counter value shall be increased by 1. The counter value shall be formatted as a 4 byte big endian value. A counter value shall never be reused in any future encryption using the same key. For the encryption, the IV is the nonce value and the AAD is an empty string. The format of the encrypted message is the ciphertext, followed by 16 bytes of the tag.

Table 22 — Cipher suite

Cipher suite value	Operation	Definition	Specification
1	ECDH	ECKA-DH	BSI TR-03111
	KDF	HKDF-SHA-256	RFC 5869
	Encrypt	AES-256-GCM	NIST SP 800-38D
	MAC	HMAC-SHA-256	RFC 2104

Table 23 shows the different curves used for the ephemeral key pair. Support for all of these curves is mandatory for an mDL Reader. For session encryption, only curves with the purpose ECDH shall be used.

Table 23 — Elliptic curves

Definition	Specification	Curve identifier	Purpose
Curve P-256	FIPS PUB 186-4	IANA COSE registry	ECDH/ECDSA
Curve P-384	FIPS PUB 186-4	IANA COSE registry	ECDH/ECDSA
Curve P-521	FIPS PUB 186-4	IANA COSE registry	ECDH/ECDSA
X25519	RFC 7748	IANA COSE registry	ECDH
X448	RFC 7748	IANA COSE registry	ECDH
Ed25519	RFC 8032	IANA COSE registry	EdDSA
Ed448	RFC 8032	IANA COSE registry	EdDSA
brainpoolP256r1	RFC 5639	-65537	ECDH/ECDSA
brainpoolP320r1	RFC 5639	-65538	ECDH/ECDSA
brainpoolP384r1	RFC 5639	-65539	ECDH/ECDSA
brainpoolP512r1	RFC 5639	-65540	ECDH/ECDSA

The informative example can be found in D.5.1.

NOTE A proposal will be made to the IANA registry for COSE curve identifiers for the curves not currently defined within the IANA registry, these identifiers are therefore subject to change in future versions of this document.

9.2.2 Issuer data authentication

9.2.2.1 Purpose

The purpose of issuer data authentication is to confirm that the mDL data is issued by the issuing authority and that it has not changed since issuance.

9.2.2.2 Applicability

This mechanism is applicable for an mDL supporting the offline retrieval.

NOTE Similar methods are described for the online retrieval (8.2.1.2).

9.2.2.3 Description

Issuer data authentication is implemented by way of a digital signature over mDL data, using a public-private (asymmetric) key pair.

A separate randomized message digest is calculated for each data element and included in the mobile security object (MSO), defined in 9.2.2.4. The MSO is then digitally signed using a private key that is kept secret by the IA and the digital signature is added to the mDL data.

NOTE A message digest has the following properties:

a) It is very small in size compared to the mDL data.

b) The probability of finding any two (different) mDL data sets that lead to the same message digest is negligible. This has the following implications:

1) The probability of finding an mDL data set A that produces the same message digest as a given mDL data set B is negligible.

2) The probability that a message digest (for the data on an mDL) remains the same upon a change in the data is negligible.

The public key belonging to the private key used for the digital signature is provided as part of the document signer (DS) certificate as defined in B.1.2.3. When the mDL is presented to an mDL Reader, the mDL Reader retrieves this DS certificate. The IACA root certificate as defined in B.1.2.1. is used to sign the document signer certificate. In order to verify the certificate chain of IACA and DS certificate, the following or equivalent checks from sections 6.1.1, 6.1.2 and 6.13 RFC 5280 should be performed:

From 6.1.1:

(a) Path length shall be 0

(b) Current date/time

(d) Take (1),(2),(3) and (4) from the IACA certificate

From 6.1.2:

(g) working_public_key_algorithm: from IACA certificate

(h) working_public_key: from IACA certificate

(i) working_public_key_parameters: from IACA certificate

(j) working_issuer_name: from IACA certificate

(k) max_path_length: 0

From 6.1.3:

(a) verify (1),(2),(3) and (4)

The following steps taken from ICAO 9303 part 12 should be performed:

— Assign the certificate subjectPublicKey to working_public_key.

— Assign the subjectPublicKeyInfo parameters, i.e. the namedCurve, to the working_public_key_parameters variable.

— Assign the certificate subjectPublicKey algorithm to the working_public_key_algorithm variable.

— Recognize and process any other critical extensions present in the certificate.

— Process any other recognized non-critical extensions present in the certificate. This includes the extended key usage extensions.

Furthermore, the following steps should also be performed:

a) Verify that the countryName element in the subject of the IACA certificate and countryName element in the DS certificate are the same.

b) Verify that the extended key usage in the DS certificate contains the identifier for use as a DS certificate.

After confirming the DS certificate is valid and is issued by a known and trusted IACA certificate, the mDL Reader uses the document signer public key to verify the digital signature in the COSE_Sign1 structure. The mDL Reader also computes the message digest of each of the received data elements and compares them with the corresponding message digest stored in the MSO. If the following conditions are met, the mDL Reader can consider that the received data elements are authentic:

a) The DS certificate is authenticated.

b) The digital signature verifies with the public key provided in the DS certificate.

c) The calculated message digests are the same as the message digests stored in the MSO.

d) If the mDL Reader retrieved the issuing_country element, it shall be verified that the value of that element matches the countryName element in the subject field within the DS certificate.

e) The DocType in the MSO matches the relevant DocType in the “Documents” structure.

f) The elements in the ‘ValidityInfo’ structure are verified against the current time stamp.

This document does not mandate methods to obtain and/or to establish trust in IACA certificates. It is the responsibility of the person or organization responsible for the mDL Reader to obtain and/or to establish trust in the IACA certificates used to verify a DS certificate. However, examples of methods and approaches to establish such trust in IACA certificates are provided in Annex C, which describes a method for distribution for IACA certificates. In any case, the issuing authority shall publicly publish its issuing authority certificate authority (IACA) root certificate.

This document does not prescribe methods for the generation, administration and safekeeping of key pairs. It is the responsibility of each issuing authority to ensure that keys are generated, administered and protected as necessary.

The certificate profiles for issuer data authentication specified in B.1.2 shall be used.

9.2.2.4 Signing method and structure

An mDL digital signature is generated over the mobile security object (MSO). The MSO shall be included in the mDL Response and has the following CDDL structure:

```
MobileSecurityObject = {
    "digestAlgorithm" : tstr,               ; Message digest algorithm used
    "valueDigests" : ValueDigests,          ; Array of digests of all data elements
    "deviceKey" : DeviceKey,
    "docType" : tstr,                       ; DocType as used in Documents
    "validityInfo" : ValidityInfo
}

DeviceKey = COSE_Key                        ; Device key in COSE_Key as defined in RFC
8152

ValueDigests = {
    "nameSpaces" : NameSpacesDigests
}

NameSpacesDigests = {
    + NameSpace => DigestIDs
}

DigestIDs = {
    + DigestID => Digest
}

ValidityInfo = {
    "signed" : tdate,
    "validFrom" : tdate,
    "validUntil" : tdate,
    ? "expectedUpdate" : tdate
}

NameSpace = tstr                            ; NameSpace as used in IssuerSigned
DigestID = uint                             ; DigestID as used in IssuerSignedItem
Digest = bstr                               ; digest(IssuerSignedItem)
```

The ‘digestAlgorithm’ and ‘valueDigests’ are the digest algorithm identifier and the digests of the data elements as further specified in 9.2.2.5. The ‘deviceKey’ is the public key pair used for mDL authentication. The ‘deviceKey’ element is encoded as an untagged COSE_Key element as specified in RFC 8152. Only curves specified in Table 23 are allowed. Curve identifiers from the COSE IANA registry shall be used where applicable. For other curves, Table 23 shall be used.

The DigestID is unsigned integer that is used to match the hashes in the MSO to the data elements in the mDL Response. The DigestID shall be unique within a NameSpace. It should be dynamic to prevent the MSO leaking on what data elements are present on a specific mdl.

The DocType is the document type of the document and shall be identical to DocType element in the mDL Response as defined in 8.2.1.1.2.2.

The ‘ValidityInfo’ structure contains information related to the validity of the MSO and its signature, and therefore the associated mDL. The signed element is the timestamp at which the MSO signature was created. The‘validFrom’ element contains the timestamp before which the mDL data is not yet valid. This can be used for when a change of mDL data is expected in the future, for example a change in the age data elements. The timestamp of ‘validFrom’ shall be equal or later than the ‘signed’ element.

The ‘validUntil’ element contains the timestamp after which the mDL data is no longer valid. The value of the timestamp shall be later than the ‘validFrom’ element. The optional ‘expectedUpdate’ element contains the timestamp at which the IA expects to resign the MSO (and potentially update data elements).

The ‘validFrom’ timestamp shall not be before the value of the ‘issue_date’ element from Table 3. The ‘validUntil’ timestamp shall not be beyond the value of the ‘expiry_date’ element from Table 3.

Since the elements in the ‘ValidityInfo’ structure can provide linkability clues, it is recommended to set these timestamps with a precision that limits the linkability information. This can be done for example by setting the hh,mm and ss information to the same value on each provisioned mDL.

The MSO is encapsulated and signed by the untagged COSE_Sign1 structure as defined in RFC 8152 and identified as "IssuerAuth" in 8.2.1.1.2.2. Within the COSE_Sign1 structure, the 'payload' shall be the 'MobileSecurityObject' structure. The 'external_aad' field used in the 'Sig_structure' shall be a bytestring of size zero.

The "alg" element (RFC 8152) shall be included as an element in the protected header. Other elements should not be present in the protected header. One of the signature algorithms "ES256" (ECDSA with SHA-256), "ES384" (ECDSA with SHA-384), "ES512" (ECDSA with SHA512) or "EdDSA" (EdDSA) as specified in RFC 8152 shall be used.

The DS certificate shall be included as a 'x5chain' element as described in "draft-ietf-cose-x509-04". It shall be included as an unprotected header element.

NOTE 1 this 'x5chain' element only consists of a single certificate.

NOTE 2 the x5chain element has the temporary identifer 33 registered in the IANA registry.

An informative example can be found in D.5.2.

9.2.2.5 Message digest function

The document signer shall use one of the following digest algorithms: SHA-256, SHA-384 or SHA-512 specified in ISO/IEC 10118-3. The use of SHA-256 is recommended. The algorithms shall be identified as defined in Table 24.

Table 24 — Digest algorithm identifiers

Digest algorithm	digestAlgorithm identifier
SHA-256	"SHA-256"
SHA-384	"SHA-384"
SHA-512	"SHA-512"

A digest is calculated separately for each data element present on the mDL and stored in the MSO. The same digest function shall be used for all data elements. Digests are identified by the combination of the nameSpace (7.3.3) and the DigestID (9.2.2.4). The input for the digest function is the binary data of the IssuerSignedItem (8.2.1.1.2.2). Each IssuerSignedItem also contains a random value. This value shall be different for each IssuerSignedItem and shall have a minimum length of 16 bytes. The purpose of the random value is to ensure that the digest value of the IssuerSignedItem by itself does not provide any information about it contents.

NOTE It is not necessary for the mDL Reader to retrieve all the data present on the mDL to verify the received mDL data; it can verify the signature on the whole MSO and use the DigestID and nameSpace for each received data element to find and verify those data elements.

9.2.3 mDL authentication

9.2.3.1 Purpose

The security objective of mDL authentication is to prevent cloning of the mDL and to mitigate man in the middle attacks.

9.2.3.2 Applicability

This mechanism is applicable for an mDL using offline retrieval.

9.2.3.3 Description

The mDL private key, which belongs to the mDL public key stored in the MSO, is used to authenticate the mDL. It is also used to authenticate the response data contained in the DeviceSignedItems structure.

The mDL public key is stored in the MSO, see 9.2.2.4. The mDL Reader assumes that the mDL is authentic only if the authentication signature or MAC is correct.

The mDL authentication key shall only be used with either mDL MAC Authentication or mDL DSA Authentication during its lifetime.

Security requirements regarding storage of credential information, including the mDL private key are out of scope for this document. It is the responsibility of the issuing authority to ensure that all data stored in the mDL is stored securely.

9.2.3.4 Mechanism

The mDL authentication key pair consists of a public and a private key (SDeviceKey.Priv, SDeviceKey.Pub). The public key is accessible through the DeviceKey element in the MSO. The DeviceKey shall use one of the algorithms from Table 23.

The mDL authentication key shall be used to authenticate the mDL in one of two ways: ECDH-agreed MAC or ECDSA / EdDSA signature. An mDL may choose either approach, but shall choose only one. An mDL authentication key shall not be used to produce both MACs and signatures. The MAC method is recommended.

The data that the mDL authenticates is the DeviceAuthentication structure as defined below. The mDL shall generate this structure and calculate either the MAC or signature. In order to verify the data, the mDL Reader shall generate the structure as well and validate the MAC or signature.

NOTE The DeviceAuthentication structure itself is not transferred as part of the mDL Response, only the resulting MAC or signature.

```
DeviceAuthentication = [
    "DeviceAuthentication",
    SessionTranscript,
    DocType,                    ; DocType as used in Documents structure in
OfflineResponse
    DeviceNameSpacesBytes       ; Same as in OfflineResponse
]

SessionTranscript = [
    DeviceEngagementBytes,
    EReaderKeyBytes             ; Same as in SessionEstablishment
]

DeviceEngagementBytes = #6.24(bstr .cbor DeviceEngagement) ; Containing DeviceEngagement
```

An informative example can be found in D.5.3.

9.2.3.5 mDL MAC Authentication

To authenticate the mDL with mDL MAC authentication, the mDL computes the MAC of the DeviceAuthentication structure with an ephemeral MAC key derived from the mDL authentication private key and the reader ephemeral public key. The mDL calculates this ephemeral MAC key (EMacKey) by performing KDF(ECDH(SDeviceKey.Priv, EReaderKey.Pub)) and the mDL Reader calculates this EMacKey by performing KDF(ECDH(SDeviceKey.Pub, EReaderKey.Priv)). The KDF and ECDH functions are described in 9.2.1 and shall be the same as the cipher suite indicated in the device engagement. The EMacKey shall be derived using a salt of 0x00

NOTE Deriving the key for the MAC calculation is very similar to the key derivation for the session keys as described in 9.2.1, with the difference that not the ephemeral mDL key pair is used, but the non-ephemeral key pair belonging to the DeviceKey element in the MSO.

The MAC value is contained in the MAC element within DeviceAuth in an untagged COSE_Mac0 structure as defined in RFC 8152. Within the COSE_Mac0 structure, the payload shall have a nil value. The

detached content is the DeviceAuthentication structure. The 'external_aad' fields shall be a bytestring of size zero.

The "alg" element (RFC 8152) shall be included as an element in the protected header. Other elements should not be present in the protected header. The MAC functions are described in Table 22 and shall be the same as the cipher suite indicated in the device engagement. RFC 8152 describes the algorithm identifiers that shall be used in the "alg" element. One of the following algorithms shall be used: "HMAC 256/256" (HMAC with SHA-256).

The MAC method is recommended because it does not require the mDL to produce a potentially non-repudiable signature over mDL Reader-provided data. The mDL can always deny the MAC value to a third party because the mDL Reader could have produced it by itself.

9.2.3.6 mDL DSA Authentication

To authenticate the mDL with mDL ECDSA / EdDSA authentication, the mDL signs the DeviceAuthentication structure with the mDL authentication private key. The signature is contained in the Signature element within DeviceAuth in an untagged COSE_Sign1 structure as defined in RFC 8152. Within the COSE_Sign1 structure, the payload shall have a nil value. The detached content is the DeviceAuthentication structure. The 'external_aad' fields shall be a bytestring of size zero.

The "alg" element (RFC 8152) shall be included as an element in the protected header. Other elements should not be present in the protected header. RFC 8152 describes hashing algorithms that can be used as signature algorithms. One of the following signature algorithms shall be used: "ES256" (ECDSA with SHA-256), "ES384" (ECDSA with SHA-384), "ES512" (ECDSA with SHA512) or "EdDSA" (EdDSA).

The mDL ECDSA / EdDSA authentication method is not recommended because it requires the mDL to produce a potentially non-repudiable signature over data provided by the mDL Reader.

9.2.4 mDL Reader authentication

9.2.4.1 Purpose

mDL Reader authentication uses information stored in the mDL Reader to confirm that the mDL Reader and the mDL Reader Request are authenticated. mDL Reader authentication is an optional mechanism.

9.2.4.2 Applicability

This mechanism is applicable for an mDL Reader using the offline retrieval.

9.2.4.3 Description

A private key stored in the mDL Reader is used to authenticate the mDL Reader, and used to authenticate the mDL Reader Request. The mDL Reader public key is stored in a certificate.

9.2.4.4 Mechanism

The mDL Reader authentication key pair consists of a public and a private key. The public key is accessible through a certificate provided with the mDL Reader Request.

The mDL Reader authentication key may be used to authenticate the mDL Reader by ECDSA / EdDSA signature.

The data that the mDL Reader authenticates is the ReaderAuthentication structure as defined below. The mDL Reader shall generate this structure and calculate the signature. In order to verify the data, the mDL shall generate the structure as well and validate the signature.

The signature is contained in an untagged COSE_Sign1 structure as defined in RFC 8152. Within the COSE_Sign1 structure, the payload shall have a nil value. The detached content is the ReaderAuthentication structure. The 'external_aad' fields shall be a bytestring of size zero.

The "alg" element (RFC 8152) shall be included as an element in the protected header. RFC 8152 describes hashing algorithms that can be used as signature algorithms. One of the following signature algorithms should be used: "ES256" (ECDSA with SHA-256), "ES384" (ECDSA with SHA-384), "ES512" (ECDSA with SHA512) or "EdDSA" (EdDSA).

NOTE The ReaderAuthentication structure itself is not transferred as part of the mDL Reader Request, only the resulting signature.

```
ReaderAuthentication = [
    "ReaderAuthentication",
    SessionTranscript,
    ItemsRequestBytes       ; Same as in OfflineRequest
]

SessionTranscript = [
    DeviceEngagementBytes,
    EreaderKeyBytes         ; Same as in SessionEstablishment
]

DeviceEngagementBytes = #6.24(bstr .cbor DeviceEngagement) ; Containing DeviceEngagement
```

The certificate containing the mDL Reader public key shall be included as a 'x5chain' element as described in "draft-ietf-cose-x509-04". It shall be included as an unprotected header element. The 'x5chain' shall include at least one certificate and may contain more.

Use of the mDL Reader authentication profile in B.1.4 is recommended.

9.3 Online retrieval

9.3.1 TLS

Communication between the mDL Reader and the issuing authority shall use Transport Layer Security (TLS) with server authentication. Therefore the mDL Reader and the issuing authority shall support TLS version 1.2 specified in RFC 5246 and may support TLS version 1.3 specified in RFC 8446, E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.3, August 2018. The mDL Reader shall act as TLS client and the issuing authority as TLS server. For the server authentication the issuing authority shall use an ECDSA key pair and the corresponding certificate shall be signed by the issuing authority's root key. The mDL Reader and the issuing authority may support TLS client authentication. Use of the TLS client authentication certificate profile in B.1.4 is recommended The key pairs used for the TLS authentication shall not be used for other purposes.

If the TLS server indicates in its TLS server certificate, see clause B.1.3.2, the support of the Online Certificate Status Protocol (OCSP), the TLS server shall support the TLS mechanisms to exchange the OCSP status information. More specifically for TLS version 1.2 the TLS server shall support the certificate status request for OCSP specified in RFC 6066 clause 8. If the server supports TLS version 1.3 according to RFC 8446 it shall support the TLS version 1.3 mechanisms to exchange the OCSP status information.

An mDL Reader requesting OCSP status information for the TLS server certificate shall use the corresponding TLS mechanisms to request this information. The OCSP signer certificate profile is defined in B.1.5.

The certificate profiles for the certificates used for TLS are defined in B.1.3

This TLS connection shall use one of the cipher suites listed in Table 25. The mDL Reader shall support all these cipher suites; the issuing authority shall support at least one of these cipher suites.

Table 25 — TLS cipher suites

Cipher suite	Reference
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	RFC 8422
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	RFC 8422
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256	RFC 7905

The key exchange shall make use of an elliptic curve listed in Table 26.

Table 26 — Elliptic curves for the TLS key exchange

Elliptic curve	Reference
P-256	FIPS PUB 186-4
P-384	FIPS PUB 186-4
P-521	FIPS PUB 186-4
x25519	RFC 7748
x448	RFC 7748

9.3.2 JWS

A JWT shall be protected using a JSON Web Signature (JWS). JWS is specified in RFC 7515 – JSON Web Signature. The JWS shall be signed with the JWS certificate, and this certificate shall be provided in the JWS header in the registered x5c attribute according to RFC 7515. The certificate profiles for the certificates used for JWS are defined in B.1.3. One of the following JSON Web Algorithm (JWA;RFC 7518) shall be used:

a) ES256: ECDSA using P-256 and SHA-256

b) ES384: ECDSA using P-384 and SHA-384

c) ES512: ECDSA using P-521 and SHA-512

An informative example can be found in D.5.4.

Annex A
(informative)

Mobile driving licence use cases

A.1 General

This Annex provides practical examples of use cases where a mobile driving licence is expected to be used. The primary purposes of a driving licence is to confirm identity and convey driving privileges. issuing authorities take their role as Identity Proofers seriously, resulting in trusted and widely used physical credential (card). The trusted identity attributes confirmed by these government entities (photo, address, date of birth/age, full name ...) are of value to establishments that need to verify a customer's age, identity, current contact information, or driving privileges.

A.2 Offline and online use cases

Driving licences are used every day in a myriad of use cases. AAMVA's Mobile Driver Licence Functional Needs Whitepaper describes the core of these use cases. The future of a driving licence can change dramatically as mDL and mDL Reader support these scenarios:

— The mDL can be connected to Internet infrastructure (online) or disconnected from it (offline);

— The mDL Reader can likewise be connected (online) or disconnected (offline);

— An attendant can host the mDL transaction and be trusted to verify the identity of the mDL Holder; or

— The transaction can be executed unattended where the mDL authentication and mDL verification is done remotely. Unattended use cases are not supported in this edition of the document.

There are two types of mDL use cases envisioned in this document and they are defined as follows:

— Offline: disconnected or offline mDLs can transmit groups of data attributes about the mDL Holder to a disconnected or offline mDL Reader over any communication channel supported by both devices. Data resides on the mobile device and it arrives intact with proof that there was no tampering.

— Online: connected or online mDL Readers can receive a "token" from the mDL that reveals nothing about the mDL Holder, but that the mDL Reader uses that "token" to request data from the issuing authority (directly or indirectly). mDL data is not retrieved from the mDL but from the issuing authority. The online use case may enable unattended remote transactions. Unattended use cases are not supported in this edition of the document.

A.3 mDL use cases

A.3.1 General

Following subclauses provide mDL use cases. This is not an exhaustive list of use cases but common uses of mDL. This document does not preclude other uses of mDL not defined in this subclause.

As the world transitions to digital, mDL has the opportunity to revolutionize workflows by providing multi-channel access to secure and privacy-protecting identifiers for users that support brand new workflows at any level of security.

A.3.2 Law enforcement roadside stop

Driving privileges may need to be provided to police or law enforcement official during a road or a traffic stop for a possible violation of law. During a road stop, the law enforcement officer conducting the road stop needs to identify and authenticate the driver of the vehicle and his / her driving privileges. The verification process is typically conducted in a non-controlled physical environment by a person who is trained in handling mDL. This operation can be either online or offline. Complete data is required / requested by the official in this use case. See Table A.1 for details.

Table A.1 — Use case: show driving privilege / roadside stop

Role	Law Enforcement, Vehicle Operating Authority
Data Needed	Complete driving licence data

A.3.3 Purchase age-restricted item

Oftentimes the purchase of certain commodities are generally restricted to persons beyond a certain age threshold – Alcohol and Tobacco Products are some examples. Similarly, there are age restricted venues, like bars and clubs, which sell items that are restricted to persons who are above a certain age. Establishments complying with such laws, will possibly verify the age using the mDL. The purchase could be made in-person.

The verification process could be conducted in a controlled physical environment by a person who is not trained in handling mDL. Only age proof and face image is required to conduct the transaction. See Table A.2 for details.

Table A.2 — Use case: purchase age-restricted item

Role	Alcohol Vendor, Tobacco Vendor, Lottery Vendor
Data Needed	Age Attestation or Date of Birth, Face Image

A.3.4 Enter a bar, club, or restaurant

Age proof may be required to enter a bar, a club, or a restaurant. A variant of this use case is entering a Casino, where additional legal requirements may require the mDL Holder to divulge his / her name, or may volunteer his / her name be matched to no-gamble lists.

The verification process is typically conducted in a controlled physical environment by a person who is not trained in handling mDL. This operation could be offline or online. At a minimum, age proof and faceimage is required to conduct the transaction. See Table A.3 for details.

Table A.3 — Use case: enter a bar, a club or a restaurant

Role	Bar, Club, Restaurant, Casino
Data Needed	Age Attestation or Date of Birth, Face Image, Name

A.3.5 Senior citizen benefits

Age proof may be required to get access to senior citizen benefits. The verification process is typically conducted in a controlled physical environment by a person who is not trained in handling mDL. This operation could be offline or online. At a minimum age proof and faceimage is required to conduct the transaction. See Table A.4 for details.

Table A.4 — Use case: senior citizen benefits

Role	Public transportation, Museum, Concert venue
Data Needed	Age Attestation or Date of Birth, Face Image

A.3.6 Car rental and car sharing

For renting an automobile, an mDL can identify the renter as well as to confirm driving privileges, and perhaps even up-to-date driving privileges. This can happen at rental counters or vehicle exit gates. The rental car experience is one where reimagining the user's experience from start to finish could make for an optimal experience. Smart rental companies will reimagine the whole process from booking the vehicle, to arriving, walking past the rental counter, having the reserved vehicle unlock automatically when it detects the proper mDL, and allowing the mDL Holder, after identity verification, to drive off the lot without stopping (potential unattended offline or online use case).

Car sharing programs have become popular in urban areas where program members schedule to pick up a car from wherever they are to get to a destination or for a round trip. In some areas and some programs, these are high-value automobiles. mDL can bring efficiencies and security to the car sharing programs.

The verification process is typically conducted in a non-controlled physical environment by a person who may or may not be trained in handling mDL. This operation can be either online or offline. See Table A.5 for details.

Table A.5 — Use case: car rental or car sharing

Role	Car Rental Company or Car Manufacturer
Data Needed	Issuing Authority, Driving Privileges, Issuance Date, Expiration Date, Full Name, Address, Face Image, Age Attestation

A.3.7 Check into a hotel

mDL can be required to confirm identity and contact information upon checking into a hotel.

The verification process is typically conducted in a controlled physical environment by a person who may not be trained in handling mDL. This operation can be either online or offline. See Table A.6 for details.

Table A.6 — Use case: check into a hotel

Role	Hotel Owner
Data Needed	Issuing Authority, Full Name, Address, Face Image

A.3.8 Secure building access

A driving licence is used to log and gain access to certain facilities or areas (e.g. federal facilities, schools, etc.). An mDL can be used to control access to certain facilities or areas, e.g. federal facilities.

The verification process is typically conducted in a controlled physical environment by a person who may be trained in handling mDL. This operation can be either online or offline. See Table A.7 for details.

Table A.7 — Use case: secure building access

Role	Facilities Owner
Data Needed	Issuing Authority, Full Name, Face Image

A.3.9 Airport security

In certain territories, any person wishing to enter the secure area of a commercial airport has to identify him or herself with government issued ID, and match to flight lists or boarding passes for that particular day. Identity Verification and flight checks are performed by the local authorities. Where DL is accepted as a means of identification, an mDL could be an alternate mechanism creating the opportunity for speed and efficiency gains in airports for mDL Holders.

The verification process is typically conducted in a controlled physical environment by a person who may be trained in handling mDL. This operation can be either online or offline. See Table A.8 for details.

Table A.8 — Use case: airport security

Role	Immigration / Airport Security Authority
Data Needed	Issuing Authority, Expiration Date, Full Name, Face Image

A.3.10 Vote or register to vote

Most jurisdictions accept government issued ID card as proof of identity for voting. Voting requires proof of identity and residency. mDL can be used to verify voter identity and jurisdiction of residency.

The verification process is typically conducted in a non-controlled physical environment by a person who may or may not be trained in handling mDL. This operation can be either online or offline. See Table A.9 for details.

Table A.9 — Use case: vote or register to vote

Role	Voting Site
Data Needed	Resident City and/or Resident Postal Code, Full Name, Face Image

Annex B
(normative)

Certificate profiles

B.1 Certificate profiles

B.1.1 Overview

The certificate profiles defined in this clause are mandatory. The IACA root certificate is the root certificate used to issue all end-entity certificates. Sub-CA's shall not be used. The usage of the end-entity certificates are defined using the extended key usage extension. All end-entity certificates shall contain the same country code as the IACA certificate. An IACA certificate may contain the stateOrProvinceName element to indicate that this IACA certificate only issues mDL's within that region. If the element is present in the IACA certificate, it shall also be present and hold the same value in the end-entity certificates.

All certificates shall be DER encoded.

For each certificate profile, the field type column indicates whether an element is mandatory (m), optional (o) or non-critical (mc).

The following extensions shall not be used:

- PolicyMappings
- NameConstraints
- PolicyConstraints
- InhibitAnyPolicy
- FreshestCRL

B.1.2 Issuer data authentication certificate profiles

B.1.2.1 IACA root certificate

This certificate profile **defines** the baseline for the IACA certificate, establishing the allowed security parameters for interoperability. The IACA certificate is used as the root for all certificates defined in this (Annex B) clause. One IACA certificate may be used by multiple Issuing Authorities within one country. See Table B.1 for details.

Issuing Authorities should define the validity period of the IACA root certificate as the sum of:

- The longest validity length of the end entity certificates issued (i.e. document signer certificates, JWS certificates, TLS server certificate and OCSP signer)
- Usage period: time during which end entity certificates will be issued
- Lead times: time required to create and disseminate the IACA root certificate, e.g. through Masterlists before and after its usage period

NOTE The longest validity length of the end entity certificates includes any document signer certificates for IDLs with a secure integrated circuit. IDLs with a secure integrated circuit and the corresponding document signer certificates typically have a longer maximum validity than document signer certificates for mDLs.

The private key usage period shall be carefully set, balancing the risk of having the too many documents issued under the same IACA root certificate against the efforts and lead time required to create new IACA root certificates too often. The recommended period is between 3 to 5 years."

Table B.1 — IACA root certificate

Certificate Component	Section in RFC 5280	Field Type	Description
Version	4.1.2.1	m	Shall be `v3`
Serial Number	4.1.2.2	m	Non-sequential positive, non-zero integer, minimum containing at least 63 bits of output from a CSPRNG, maximum 20 octets.
Signature	4.1.2.3	m	Value shall match the OID in Signature Algorithm (below)
Issuer	4.1.2.4	m	`countryName` is mandatory. The value shall be in upper case and contain the ISO 3166-1 two-letter code of the issuing country, exactly the same value as in the issuing country data element. `stateOrProvinceName` is optional. If it this element is present, the element shall also be present in the end-entity certificates and hold the same value. The value shall exactly match the value of the data element issuing_jurisdiction, if that element is present on the mDL. `organizationName` is mandatory. `commonName` shall be present. Its value is at the discretion of the IACA. `countryName` and `serialNumber`, if present, shall be `PrintableString`. Other attributes that have `DirectoryString` syntax shall be either `PrintableString` or `UTF8String`.
Validity	4.1.2.5	m	
Not Before		m	Date on which the certificate validity period begins.
Not After		m	Maximum of 15 years after “Not Before” date.
Subject	4.1.2.6	m	Same exact binary value as Issuer.
Subject Public Key Info	4.1.2.7	m	
algorithm		m	`1.2.840.10045.2.1` (Elliptic curve)
parameters		m	Implicitly specify curve parameters through an OID associated with a Brainpool curve specified in RFC 5639 or a NIST approved curve referenced in SP 800-56A rev 3 (`{ EC Parameters namedCurve }`): `1.3.36.3.3.2.8.1.1.7` (brainpoolP256r1) `1.3.36.3.3.2.8.1.1.9` (brainpoolP320r1) `1.3.36.3.3.2.8.1.1.11` (brainpoolP384r1) `1.3.36.3.3.2.8.1.1.13` (brainpoolP512r1) `1.2.840.10045.3.1.7` (Curve P-256) `1.3.132.0.34` (Curve P-384) `1.3.132.0.35` (Curve P-521)
subjectPublicKey		m	Public key shall be encoded in uncompressed form.
X.509v3 Extensions	4.2	m	Further extensions may be present if they are marked non critical.
Subject Key Identifier	4.2.1.2	m	SHA-1 hash of the subject public key `BIT STRING` value (excluding tag, length, and number of unused bits)
Key Usage	4.2.1.3	mc	
Digital Signature			`0`
Non Repudiation			`0`
Key Encipherment			`0`
Data Encipherment			`0`
Key Agreement			`0`
Key Certificate Signature			`1`

Table B.1 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
CRL Signature			1
Encipher Only			0
Decipher Only			0
IssuerAltName	4.2.1.7	m, nc	The issuer alternative name extension shall provide contact information for the issuer of the certificate. For that purpose the issuer alternative name shall include at least one of • rfc822Name or • uniformResourceIdentifier NOTE This contact information is intended to help establish trust in the certificate and the certified key by appropriate out of band mechanisms. Note that this information is only meant for contact information and does not in itself imply any level of trust in the certificate.
Basic Constraints	4.2.1.9	mc	
CA		m	TRUE
pathLenConstraint		m	0
CRLDistributionPoints	4.2.1.13	m	The 'reasons' and 'cRL Issuer' fields shall not be used
distributionPoint		m	URI HTTP for CRL Distribution Point
Signature Algorithm	4.1.1.2	m	Options: 1.2.840.10045.4.3.2 (ECDSA-with SHA256) 1.2.840.10045.4.3.3 (ECDSA-with SHA384) 1.2.840.10045.4.3.4 (ECDSA with SHA512)
Signature Value	4.1.1.3	m	Value according to Signature Algorithm. By imprinting this signature, the CA certifies the binding between the public key material and the subject of the certificate, i.e. the IACA.

B.1.2.2 IACA link certificate

This certificate profile defines the baseline for the IACA link certificate, establishing the allowed security parameters for interoperability. The IA should generate and distribute an IACA link certificate when doing an IACA re-key. The link certificate establish a trust path from the old IACA root certificate to the new one. See Table B.2 for details.

Table B.2 — IACA link certificate

Certificate Component	Section in RFC 5280	Field Type	Description
Version	4.1.2.1	m	Shall be v3
Serial Number	4.1.2.2	m	Non-sequential positive, non-zero integer, minimum containing at least 63 bits of output from a CSPRNG, maximum 20 octets.
Signature	4.1.2.3	m	Value shall match the OID in Signature Algorithm (below)
Issuer	4.1.2.4	m	Same exact binary value as Subject in the old IACA root certificate, for which the respective private key is signing this link certificate.
Validity	4.1.2.5	m	
Not Before			Date on which the link certificate validity period begins.
Not After			Date shall not be after the "Not After" date of the new IACA root certificate.
Subject	4.1.2.6	m	Same exact binary value as Issuer in the new IACA root certificate.
Subject Public Key Info	4.1.2.7	m	Same as Subject Public Key Info in the new IACA root certificate.

Table B.2 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
X.509v3 Extensions	4.2	m	Further extensions may be present if they are marked non critical.
Subject Key Identifier	4.2.1.2	m	SHA-1 hash of the subject public key `BIT STRING` value (excluding tag, length, and number of unused bits)
Key Usage	4.2.1.3	mc	
Digital Signature			`0`
Non Repudiation			`0`
Key Encipherment			`0`
Data Encipherment			`0`
Key Agreement			`0`
Key Certificate Signature			`1`
CRL Signature			`1`
Encipher Only			`0`
Decipher Only			`0`
Extended key usage	4.2.1.12	m	
Key usage		m	`1.0.18013.5.1.6` (ISO/IEC 18013-5 OID reserved for IACA link certificates)
Basic Constraints	4.2.1.9	mc	
CA		m	`TRUE`
pathLenConstraint		m	`0`
CRLDistributionPoints	4.2.1.13	m	The 'reasons' and 'cRL Issuer' fields shall not be used
distributionPoint		m	URI HTTP for CRL Distribution Point
Signature Algorithm	4.1.1.2	m	Options: `1.2.840.10045.4.3.2` (ECDSA-with SHA256) `1.2.840.10045.4.3.3` (ECDSA-with SHA384) `1.2.840.10045.4.3.4` (ECDSA with SHA512)
Signature Value	4.1.1.3	m	Value according to Signature Algorithm. By imprinting this signature, the CA certifies the binding between the public key material and the subject of the certificate, i.e. the Public Key and DN of new IACA root certificate.

B.1.2.3 mDL document signer certificate for issuer data authentication

This certificate is used to sign the mobile security object in the mDL Response.

The issuing authority should ensure that adequate security measures are used to ensure a high level of security for the use of the document signer key. See Table B.3 for details.

NOTE A method to enhance its security is for the issuing authority to use certificates that have a short lifespan (for example a few weeks), and if a compromise is detected, revoke the document signer certificate and reissue and disseminate to all concerned mDL updated data signed with a new document signer certificate.

Table B.3 — mDL document signer certificate

Certificate Component	Section in RFC 5280	Field Type	Description
Version	4.1.2.1	m	Shall be `v3`
Serial Number	4.1.2.2	m	Non-sequential positive, non-zero integer, minimum containing at least 63 bits of output from a CSPRNG, maximum 20 octets.
Signature	4.1.2.3	m	Value shall match the OID in Signature Algorithm (below)

Table B.3 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
Issuer	4.1.2.4	m	Same exact binary value as Subject of IACA certificate.
Validity	4.1.2.5	m	
Not Before		m	Date on which the certificate validity period begins.
Not After		m	Maximum of 15 months after "Not Before" date.
Subject	4.1.2.6	m	`countryName` is mandatory. The value shall be in upper case and contain the ISO 3166-1 two-letter code of the issuing country, exactly the same value as in the issuing country data element. `stateOrProvinceName` is optional. If it this element is present in the IACA root certificate, this element shall be present and hold the same value. The value shall exactly match the value of the data element issuing_jurisdiction, if that element is present on the mDL." `localityName` is optional. `organizationName` is mandatory. `commonName` shall be present. Its value is at the discretion of the IACA. `countryName` and `serialNumber`, if present, shall be `PrintableString`. Other attributes that have `DirectoryString` syntax shall be either `PrintableString` or `UTF8String`.
Subject Public Key Info	4.1.2.7	m	
algorithm		m	`1.2.840.10045.2.1` (Elliptic curve)
parameters		m	Implicitly specify curve parameters through an OID associated with a Brainpool curve specified in RFC 5639 or a NIST approved curve referenced in SP 800 56A rev 3 (`{ EC Parameters namedCurve }`): `1.3.36.3.3.2.8.1.1.7` (brainpoolP256r1) `1.3.36.3.3.2.8.1.1.9` (brainpoolP320r1) `1.3.36.3.3.2.8.1.1.11` (brainpoolP384r1) `1.3.36.3.3.2.8.1.1.13` (brainpoolP512r1) `1.2.840.10045.3.1.7` (Curve P-256) `1.3.132.0.34` (Curve P-384) `1.3.132.0.35` (Curve P-521)
subjectPublicKey		m	Public key shall be encoded in uncompressed form.
X.509v3 Extensions	4.2	m	Further extensions may be present if they are marked non critical.
Authority Key Identifier	4.2.1.1	m	
keyIdentifier		m	Same value as the Subject Key Identifier of the IACA certificate
Subject Key Identifier	4.2.1.2	m	SHA-1 hash of the subject public key `BIT STRING` value (excluding tag, length, and number of unused bits)
Key Usage	4.2.1.3	mc	
Digital Signature			`1`
Non Repudiation			`0`
Key Encipherment			`0`
Data Encipherment			`0`
Key Agreement			`0`
Key Certificate Signature			`0`
CRL Signature			`0`
Encipher Only			`0`
Decipher Only			`0`

Table B.3 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
IssuerAltName	4.2.1.7	m, nc	The issuer alternative name extension shall provide contact information for the issuer of the certificate. For that purpose the issuer alternative name shall include at least one of • `rfc822Name` or • `uniformResourceIdentifier` NOTE This contact information is intended to help establish trust in the certificate and the certified key by appropriate out of band mechanisms. Note that this information is only meant for contact information and does not in itself imply any level of trust in the certificate.
Extended key usage	4.2.1.12	m	
Key usage		m	`1.0.18013.5.1.2` (ISO/IEC 18013-5 OID reserved for mDL DS)
Subject Alternative Name	4.2.1.6	o	
Basic Constraints	4.2.1.9	mc	
CA		m	`FALSE`
CRLDistributionPoints	4.2.1.13	m	The 'reasons' and 'cRL Issuer' fields shall not be used
distributionPoint		m	URI HTTP for CRL Distribution Point
Signature Algorithm	4.1.1.2	m	Options: `1.2.840.10045.4.3.2` (ECDSA-with SHA256) `1.2.840.10045.4.3.3` (ECDSA-with SHA384) `1.2.840.10045.4.3.4` (ECDSA with SHA512)
Signature Value	4.1.1.3	m	Value according to Signature Algorithm. By imprinting this signature, the CA certifies the binding between the public key material and the subject of the certificate, i.e. the document signer.

B.1.3 Online services certificate profiles

B.1.3.1 JWS certificate

The JWS certificate is used to sign all data returned using the online data retrieval methods. See Table B.4 for details.

Table B.4 — JWS certificate

Certificate Component	Section in RFC 5280	Field Type	Description
Version	4.1.2.1	m	Shall be `v3`
Serial Number	4.1.2.2	m	Non-sequential positive, non-zero integer, minimum containing at least 63 bits of output from a CSPRNG, maximum 20 octets.
Signature	4.1.2.3	m	Value shall match the OID in Signature Algorithm (below)
Issuer	4.1.2.4	m	Same exact binary value as Subject of IACA certificate.
Validity	4.1.2.5	m	
Not Before		m	Date on which the certificate validity period begins..
Not After		m	Maximum of 15 months after "Not Before" date.

Table B.4 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
Subject	4.1.2.6	m	`countryName` is mandatory. The value shall be in upper case and contain the ISO 3166-1 two-letter code of the issuing country, exactly the same value as in the issuing country data element. `stateOrProvinceName` is optional. If it this element is present in the IACA root certificate, this element shall be present and hold the same value. The value shall exactly match the value of the data element issuing_jurisdiction, if that element is present on the mDL." `localityName` is optional. `organizationName` is mandatory. `commonName` shall be present. Its value is at the discretion of the IACA. `countryName` and `serialNumber`, if present, shall be `PrintableString`. Other attributes that have `DirectoryString` syntax shall be either `PrintableString` or `UTF8String`.
Subject Public Key Info	4.1.2.7	m	
algorithm		m	`1.2.840.10045.2.1` (Elliptic curve)
parameters		m	Implicitly specify parameters through an OID associated with a curve list in 9.3.2.
subjectPublicKey		m	Public key shall be encoded in uncompressed form.
X.509v3 Extensions	4.2	m	Further extensions may be present if they are marked non critical.
Authority Key Identifier	4.2.1.1	m	
keyIdentifier		m	Same value as the Subject Key Identifier of the IACA certificate
Subject Key Identifier	4.2.1.2	m	SHA-1 hash of the subject public key `BIT STRING` value (excluding tag, length, and number of unused bits)
Key Usage	4.2.1.3	mc	
Digital Signature			1
Non Repudiation			0
Key Encipherment			0
Data Encipherment			0
Key Agreement			0
Key Certificate Signature			0
CRL Signature			0
Encipher Only			0
Decipher Only			0
IssuerAltName	4.2.1.7	m, nc	The issuer alternative name extension shall provide contact information for the issuer of the certificate. For that purpose the issuer alternative name shall include at least one of • `rfc822Name` or • `uniformResourceIdentifier` NOTE This contact information is intended to help establish trust in the certificate and the certified key by appropriate out of band mechanisms. Note that this information is only meant for contact information and does not in itself imply any level of trust in the certificate.
Extended key usage	4.2.1.12	m	
Key usage		m	`1.0.18013.5.1.3` (ISO/IEC 18013-5 OID reserved for JWS)
Subject Alternative Name	4.2.1.6	o	

Table B.4 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
Basic Constraints	4.2.1.9	mc	
CA		m	FALSE
CRLDistributionPoints	4.2.1.13	m	The 'reasons' and 'cRL Issuer' fields shall not be used
distributionPoint		m	URI HTTP for CRL Distribution Point
Signature Algorithm	4.1.1.2	m	Options: 1.2.840.10045.4.3.2 (ECDSA-with SHA256) 1.2.840.10045.4.3.3 (ECDSA-with SHA384) 1.2.840.10045.4.3.4 (ECDSA with SHA512)
Signature Value	4.1.1.3	m	Value according to Signature Algorithm. By imprinting this signature, the CA certifies the binding between the public key material and the subject of the certificate, i.e. the issuing authority.

B.1.3.2 TLS server certificate – issuing authority

The TLS certificate is used to protect the online data retrieval methods using TLS. See Table B.5 for details.

Table B.5 — TLS server certificate: issuing authority

Certificate Component	Section in RFC 5280	Field Type	Description
Version	4.1.2.1	m	Shall be v3
Serial Number	4.1.2.2	m	Non-sequential positive, non-zero integer, minimum containing at least 63 bits of output from a CSPRNG, maximum 20 octets.
Signature	4.1.2.3	m	Value shall match the OID in Signature Algorithm (below)
Issuer	4.1.2.4	m	Same exact binary value as Subject of IACA certificate.
Validity	4.1.2.5	m	
Not Before		m	Date on which the certificate validity period begins.
Not After		m	Maximum of 27 months after "Not Before" date.
Subject	4.1.2.6	m	countryName is mandatory. The value shall be in upper case and contain the ISO 3166-1 two-letter code of the issuing country, exactly the same value as in the issuing country data element. stateOrProvinceName is optional. If it this element is present in the IACA root certificate, this element shall be present and hold the same value. The value shall exactly match the value of the data element issuing_jurisdiction, if that element is present on the mDL." localityName is optional. organizationName is mandatory. commonName shall be present. Its value is at the discretion of the IACA. countryName and serialNumber, if present, shall be PrintableString. Other attributes that have DirectoryString syntax shall be either PrintableString or UTF8String.
Subject Public Key Info	4.1.2.7	m	
algorithm		m	1.2.840.10045.2.1 (Elliptic curve)

Table B.5 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
parameters			Implicitly specify parameters through an OID associated with a Brainpool curve specified in RFC 5639 or a NIST approved curve referenced in SP 800 56A rev 3 (`{ EC Parameters namedCurve }`): `1.3.36.3.3.2.8.1.1.7` (brainpoolP256r1) `1.3.36.3.3.2.8.1.1.9` (brainpoolP320r1) `1.3.36.3.3.2.8.1.1.11` (brainpoolP384r1) `1.3.36.3.3.2.8.1.1.13` (brainpoolP512r1) `1.2.840.10045.3.1.7` (Curve P-256) `1.3.132.0.34` (Curve P-384) `1.3.132.0.35` (Curve P-521)
subjectPublicKey		m	Public key shall be encoded in uncompressed form.
X.509v3 Extensions	4.2	m	Further extensions may be present if they are marked non critical.
Authority Key Identifier	4.2.1.1	m	
keyIdentifier		m	Same value as the Subject Key Identifier of the IACA certificate
Subject Key Identifier	4.2.1.2	m	SHA-1 hash of the subject public key `BIT STRING` value (excluding tag, length, and number of unused bits)
Key Usage	4.2.1.3	mc	The exact combination of `keyUsage` bits shall be set in accordance with the particular cipher suite used.
Digital Signature			`1` (mandatory)
Non Repudiation			`0`
Key Encipherment			`1` (conditional to cipher suite used)
Data Encipherment			`0`
Key Agreement			`1` (conditional to cipher suite used)
Key Certificate Signature			`0`
CRL Signature			`0`
Encipher Only			`0`
Decipher Only			`0`
Subject Alternative Name	4.2.1.6	m	
dNSName		m	Internet domain name of the server. Can have more than one dNSName.
IssuerAltName	4.2.1.7	m, nc	The issuer alternative name extension shall provide contact information for the issuer of the certificate. For that purpose the issuer alternative name shall include at least one of • `rfc822Name` or • `uniformResourceIdentifier` NOTE This contact information is intended to help establish trust in the certificate and the certified key by appropriate out of band mechanisms. Note that this information is only meant for contact information and does not in itself imply any level of trust in the certificate.
Basic Constraints	4.2.1.9	mc	
CA		m	`FALSE`
Extended Key Usage	4.2.1.12	m	
id-kp-serverAuth		m	TLS server authentication
id-kp-clientAuth		o	TLS client authentication. Optional, to handle particular cases where the issuing authority service may need to act also as TLS client of third-party systems.

Table B.5 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
CRLDistributionPoints	4.2.1.13	m	The 'reasons' and 'cRL Issuer' fields shall not be used
distributionPoint		m	URI HTTP for CRL Distribution Point
Internet Certificate Extensions			
Authority Information Access	4.2.2.1	c	Conditional, if the IACA has an OCSP service.
Access Description OCSP			
accessMethod		m	`1.3.6.1.5.5.7.48.1` (OCSP)
accessLocation		m	URI HTTP pointing to corresponding OCSP service
Access Description CA Issuers			
accessMethod		m	`1.3.6.1.5.5.7.48.2` (CA Issuers)
accessLocation		m	URI HTTP pointing to corresponding issuing authority CA certificate
Signature Algorithm	4.1.1.2	m	Options: `1.2.840.10045.4.3.2` (ECDSA-with SHA256) `1.2.840.10045.4.3.3` (ECDSA-with SHA384) `1.2.840.10045.4.3.4` (ECDSA with SHA512)
Signature Value	4.1.1.3	m	Value according to Signature Algorithm. By imprinting this signature, the CA certifies the binding between the public key material and the subject of the certificate, i.e. the IACA online service.

B.1.4 mDL Reader authentication and TLS client authentication certificate

The certificate profile according to Table B.6 is recommended to be used for mDL Reader authentication (see 9.2.4) and TLS client authentication (see 9.3.1).

Table B.6 — mDL Reader authentication and TLS client authentication profile

Certificate Component	Section in RFC 5280	Field Type	Description
Version	4.1.2.1	m	Shall be `v3`
Serial Number	4.1.2.2	m	Non-sequential positive, non-zero integer, minimum containing at least 63 bits of output from a CSPRNG, maximum 20 octets.
Signature	4.1.2.3	m	Value shall match the OID in Signature Algorithm (below)
Issuer	4.1.2.4	m	A CA adopted by the IA, but not necessarily the IACA
Validity	4.1.2.5	m	
Not Before		m	Date on which the certificate validity period begins.
Not After		m	Maximum of 39 months after "Not Before" date.

Table B.6 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
Subject	4.1.2.6	m	Syntax and semantics of the Subject DN according to the CAB Forum Baseline Requirements with the exception of `commonName` which has the following requirements: `commonName` shall be present. This may be in the subject's preferred presentation format, or a format preferred by the IA, or some other format. Pseudonyms, nicknames, and names with spelling other than defined by the registered name may be used. This name needs not be an exact match of the fully registered organization name. However, this shall be unique within the Issuing Authority namespace. NOTE Personal certificates are not considered at this time.
Subject Public Key Info	4.1.2.7	m	
algorithm		m	`1.2.840.10045.2.1` (Elliptic curve)
parameters		m	Implicitly specify curve parameters through an OID associated with a Brainpool curve specified in RFC 5639 or a NIST approved curve referenced in SP 800 56A rev 3 (`{ EC Parameters namedCurve }`): `1.3.36.3.3.2.8.1.1.7` (brainpoolP256r1) `1.3.36.3.3.2.8.1.1.9` (brainpoolP320r1) `1.3.36.3.3.2.8.1.1.11` (brainpoolP384r1) `1.3.36.3.3.2.8.1.1.13` (brainpoolP512r1) `1.2.840.10045.3.1.7` (Curve P-256) `1.3.132.0.34` (Curve P-384) `1.3.132.0.35` (Curve P-521)
subjectPublicKey		m	Public key shall be encoded in uncompressed form.
X.509v3 Extensions	4.2	m	Further extensions may be present if they are marked non critical.
Authority Key Identifier	4.2.1.1	m	
keyIdentifier		m	Same value as the Subject Key Identifier of the Issuer CA certificate
Subject Key Identifier	4.2.1.2	m	SHA-1 hash of the subject public key `BIT STRING` value (excluding tag, length, and number of unused bits)
Key Usage	4.2.1.3	mc	
Digital Signature			1 (mandatory)
Non Repudiation			0
Key Encipherment			1 (conditional to cipher suite used)
Data Encipherment			0 or 1 Its value is at the discretion of the issuing authority
Key Agreement			1 (conditional to cipher suite used)
Key Certificate Signature			0
CRL Signature			0
Encipher Only			0
Decipher Only			0
Extended key usage	4.2.1.12	m	
id-kp-mDLReaderAuth		m	`1.0.18013.5.1.4` (ISO/IEC 18013-5 OID reserved for mDL Reader authentication)
id-kp-clientAuth		m	Allows the usage of TLS based client authentication

Table B.6 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
Basic Constraints	4.2.1.9	mc	
CA		m	FALSE
CRLDistributionPoints	4.2.1.13	m	The 'reasons' and 'cRL Issuer' fields shall not be used.
distributionPoint		m	URI HTTP for CRL Distribution Point
Internet Certificate Extensions			
Authority Information Access	4.2.2.1	m	
Access Description OCSP		c	Conditional, if the CA has an OCSP service.
accessMethod		m	1.3.6.1.5.5.7.48.1 (OCSP)
accessLocation		m	URI HTTP pointing to corresponding OCSP service
Access Description CA Issuers			
accessMethod		m	1.3.6.1.5.5.7.48.2 (CA Issuers)
accessLocation		m	URI HTTP pointing to corresponding issuing authority CA certificate
Signature Algorithm	4.1.1.2	m	Options: 1.2.840.10045.4.3.2 (ECDSA-with SHA256) 1.2.840.10045.4.3.3 (ECDSA-with SHA384) 1.2.840.10045.4.3.4 (ECDSA with SHA512)
Signature Value	4.1.1.3	m	Value according to Signature Algorithm. By imprinting this signature, the CA certifies the binding between the public key material and the subject of the certificate, i.e. the mDL Reader.

B.1.5 OCSP signer certificate profile

The OCSP signer certificate is used to sign OCSP messages. See Table B.7 —OCSP signer certificate for details.

Table B.7 — OCSP signer certificate

Certificate Component	Section in RFC 5280	Field Type	Description
Version	4.1.2.1	m	Shall be v3
Serial Number	4.1.2.2	m	Non-sequential positive, non-zero integer, minimum containing at least 63 bits of output from a CSPRNG, maximum 20 octets.
Signature	4.1.2.3	m	Value shall match the OID in Signature Algorithm (below)
Issuer	4.1.2.4	m	Same exact binary value as Subject of IACA certificate.
Validity	4.1.2.5	m	
Not Before		m	Date on which the certificate validity period begins.
Not After		m	If the OCSP signer certificate supports the CRLDistributionPoints extension: Maximum of 15 months after "Not Before" date. If the OCSP signer certificate supports the Revocation Checking of an Authorized Responder extension: Maximum of 90 days after "Not Before" date.

Table B.7 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
Subject	4.1.2.6	m	`countryName` is mandatory. The value shall be in upper case and contain the ISO 3166-1 two-letter code of the issuing country, exactly the same value as in the issuing country data element. `stateOrProvinceName` is optional. If it this element is present in the IACA root certificate, this element shall be present and hold the same value. The value shall exactly match the value of the data element issuing_jurisdiction, if that element is present on the mDL." `localityName` is optional. `organizationName` is mandatory. `commonName` shall be present. Its value is at the discretion of the IACA. `countryName` and `serialNumber`, if present, shall be `PrintableString`. Other attributes that have `DirectoryString` syntax shall be either `PrintableString` or `UTF8String`.
Subject Public Key Info	4.1.2.7	m	
algorithm		m	`1.2.840.10045.2.1` (Elliptic curve)
parameters			Implicitly specify parameters through an OID associated with a Brainpool curve specified in RFC 5639 or a NIST approved curve referenced in SP 800 56A rev 3 (`{ EC Parameters namedCurve }`): `1.3.36.3.3.2.8.1.1.7` (brainpoolP256r1) `1.3.36.3.3.2.8.1.1.9` (brainpoolP320r1) `1.3.36.3.3.2.8.1.1.11` (brainpoolP384r1) `1.3.36.3.3.2.8.1.1.13` (brainpoolP512r1) `1.2.840.10045.3.1.7` (Curve P-256) `1.3.132.0.34` (Curve P-384) `1.3.132.0.35` (Curve P-521)
subjectPublicKey		m	Public key shall be encoded in uncompressed form.
X.509v3 Extensions	4.2	m	Further extensions may be present if they are marked non critical.
Authority Key Identifier	4.2.1.1	m	
keyIdentifier		m	Same value as the Subject Key Identifier of the IACA certificate
Subject Key Identifier	4.2.1.2	m	SHA-1 hash of the subject public key `BIT STRING` value (excluding tag, length, and number of unused bits)
Key Usage	4.2.1.3	mc	
Digital Signature			`1` (mandatory)
Non Repudiation			`0`
Key Encipherment			`0`
Data Encipherment			`0`
Key Agreement			`0`
Key Certificate Signature			`0`
CRL Signature			`0`
Encipher Only			`0`
Decipher Only			`0`
Basic Constraints	4.2.1.9	mc	
CA		m	`FALSE`
Extended Key Usage	4.2.1.12	m	
id-kp-OCSPSigning		m	OCSP signing delegation, see RFC 6960

Table B.7 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Description
Revocation Checking of an Authorized Responder		c	See RFC 6960 clause 4.2.2.2.1. Either this extension or the CRLDistributionPoints extension shall be present.
id-pkix-ocsp-nocheck			
CRLDistributionPoints	4.2.1.13	c	Either this extension or the Recovation Checing of an Authorized Responder extension shall be present. The 'reasons' and 'cRL Issuer' fields shall not be used
distributionPoint		m	URI HTTP for CRL Distribution Point
Signature Algorithm	4.1.1.2	m	Options: `1.2.840.10045.4.3.2` (ECDSA-with SHA256) `1.2.840.10045.4.3.3` (ECDSA-with SHA384) `1.2.840.10045.4.3.4` (ECDSA with SHA512)
Signature Value	4.1.1.3	m	Value according to Signature Algorithm. By imprinting this signature, the CA certifies the binding between the public key material and the subject of the certificate.

B.2 CRL profile

An IACA shall generate certificate revocation information in accordance with the Certificate Revocation List (CRL) format specified in Table B.8. This CRL shall be a full and complete CRL, i.e. list all unexpired certificates issued by the IACA that have been revoked for any reason. This CRL may contain revocation information for IACA issued certificates that are not standardized in this document. An IACA shall not use an indirect or delta CRL.

If no certificates have been revoked since the last CRL was issued, an IACA shall issue a new CRL at least every 90 days. An IACA may issue CRLs more frequently than every 90 days.

If a certificate is revoked, the IACA shall issue a new CRL indicating this revocation within 48 hours.

Table B.8 — CRL profile

CRL Component	Section in RFC 5280	Field Type	Comments
Version	5.1.2.1	m	Shall be `v2`.
Signature	5.1.2.2	m	Value shall match the OID in Signature Algorithm (below).
Issuer Name	5.1.2.3	m	Same exact binary value as Subject of IACA certificate.
This Update	5.1.2.4	m	Issue date of this CRL.
Next Update	5.1.2.5	m	The next CRL will be issued no later than the Next Update date.
Revoked Certificates	5.1.2.6	c	If present, shall not be empty. Each CRL entry in the revoked certificates list shall contain the serial number of the revoked certificate and the revocation date. CRL entry extensions shall not be used.
CRL Extensions	5.2		Further extensions shall not be present
Authority Key Identifier	5.2.1	m	
keyIdentifier		m	Same value as the Subject Key Identifier of the IACA certificate
CRL Number	5.2.3	m	Sequential CRL number, increased monotonically at each new CRL issued.

Table B.8 *(continued)*

CRL Component	Section in RFC 5280	Field Type	Comments
Signature Algorithm	5.1.1.1	m	Options: `1.2.840.10045.4.3.2` (ECDSA-with SHA256) `1.2.840.10045.4.3.3` (ECDSA-with SHA384) `1.2.840.10045.4.3.4` (ECDSA with SHA512)
Signature Value	5.1.1.2	m	Value according to Signature Algorithm.

Annex C
(informative)

Master List Provider

C.1 mDL Master List Provider policy and security requirements

C.1.1 Introduction

C.1.1.1 Overview

The decentralized PKI trust model adopted by the mDL requires a mechanism to distribute and disseminate the set of Certification Authorities certificates from issuing authorities. Furthermore, the lack of a global organization with oversight of the mDL and willing to play an operational role (as is the case of ICAO for the electronic passport) limits the possibility of having a single central repository with all the IA CA certificates and working as the reference trust anchor for all mDL participants.

In this context, a mechanism referred to as "Master List" (ML) is hereby described whereby an entity (Provider) can compile, operate and provide such a trust anchor in the form of a service to mDL participants. As this service plays a critical role on the overall security and interoperability of the mDLs, a minimum set of security requirements are defined.

A ML Provider shall follow all requirements in this Annex. The ML Provider shall document the service it provides in a policy of technical and procedural controls. The policy shall follow the structure of this annex and be compliant with the respective requirements, which are mostly based on ETSI EN 319 411-1, *Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements, Version 1.2.0, August 2017* and FPKIPA, *X.509 Certificate Policy For The U.S. Federal PKI Common Policy Framework, Version 1.24, May 2015*. ML providers can further extend the policy with additional matters (e.g. business, legal, etc.). This policy should not be the only item used to assess the trustfulness of a ML Provider.

This annex does not prescribe the nature or governance of a ML Provider. In particular, it does not preclude a scenario where multiple ML Providers may coexist, from public and/or private entities, competing and/or collaborating. However, it is expected that the ML information provided is consistent amongst different ML Providers, i.e. for the set of overlapping issuing authorities, the corresponding CAs shall be the same (temporary differences may occur due to CA key rollovers or change of CA by an IA, for example).

Finally, the Master List is provided as one possible mechanism of setting a secure and interoperable Trust Model. It does not preclude other possible mechanisms, such as bilateral and/or regional agreements.

This Annex may be used for IDL's.

C.1.1.2 Document name and identification

This Policy is identified and can be referred to through the following OID:

```
id-idl-ml-policy OBJECT IDENTIFIER ::= {
    iso(1) standard(0) driving-licence (18013) part-5(5)
    master-list(3) 1}
```

Policies of Master List Providers shall be uniquely identifiable, including all published revisions.

C.1.1.3 Master list participants

This subclause provides an overview of the ML participants in the mDL.

C.1.1.3.1 Master List Providers

Master List Providers are organizations responsible for delivering the Master List, including activities such as:

— finding and regularly validating issuing authorities' Point of Contacts

— collecting information from issuing authorities through the respective Point of Contacts

— compiling the Master List

— creating and securing the Master List according to the defined format

— distributing the Master List amongst subscribers

— updating the Master List regularly

C.1.1.3.2 Issuing authorities

Issuing authorities are entities entitled to issue mDLs. Please refer to ISO/IEC 18013-1 for the standard definition.

C.1.1.3.3 Issuing authorities' Point of Contact

Each issuing authority shall designate a Point of Contact, i.e. the unique endpoint for the trusted communication channel handling all communication between the ML Provider and the issuing authority. The Point of Contact may be the same as the initial contact point (see C.1.3.2).

The Point of Contact can have multiple formats, including but not limited to a person, P.O. box, an email address, dedicated telephone line, telefax, etc. See C.1.3.2 for limitations on the initial point of contact.

C.1.1.3.4 Subscribers

Subscribers receive the ML from the ML Provider and may redistribute it amongst Relying Parties, according to the licensing terms of the ML, if any.

A subscriber can simultaneously play the role of a Relying Party and vice-versa.

C.1.1.3.5 Relying Parties

Relying Parties make use of the received ML to validate the mDLs presented for validation.

C.1.1.4 Master list usage

The Master List contains the CA certificates intended to be used by Relying Parties as the trust anchor for the verification of authenticity and integrity of mDLs.

In the absence of Master Lists that include all existing CA certificates from all Issuing Authorities it may happen that Relying Parties may have to use and combine several Master Lists to increase coverage.

C.1.1.5 Policy administration

The ML Provider shall publish under this subclause the official contacts of the person or department responsible for the Policy.

The contacts shall include name, mailing address, telephone number, and email address as minimum information.

C.1.2 Publication and repository responsibilities

The ML Provider shall make the ML available to the subscribers in a secure, mutually trusted and authenticated channel, with periodic updates. It is recommended to establish also procedures for extraordinary updates for emergency cases.

C.1.3 Identification and authentication

C.1.3.1 Naming

Issuing Authorities are uniquely identified by the 2-letter ISO 3166-1 country code. In the special cases of multiple different Issuing Authorities within a country, the country code is combined with the assigned ISO/IEC 7812 Issuer Identification Number (IIN) to the issuing authority. If no IIN is assigned, the country code shall be combined with the state or province name or code commonly used (for example in postal addresses).

The naming of Certification Authorities used by an issuing authority shall follow the guidelines of the certificate profiles defined and the format established in Master List syntax notation.

C.1.3.2 Initial identity validation

The Initial Identity Validation of the issuing authority is a critical step on the overall security of the Master List, as it is the first step to establishing a trusted communication channel between the Master List Provider and the Issuing Authorities.

After a trusted communication channel is established, the Master List Provider and the issuing authority can securely and reliably exchange information to be published in the Master List.

As a minimum, the Master List Provider is required to undertake the following steps:

a) Assess the legitimacy of and identify an initial contact point for the entity claiming to be an issuing authority. This can be achieved by a number of ways, including checking:

   - o against a governmental authoritative source,
   - o a regional/continental representative association (or similar body) of Issuing Authorities,
   - o official recognized directorate of Issuing Authorities,
   - o international conventions identifying Issuing Authorities,
   - o official reference by another issuing authority previously trusted
   - o other methods that clearly and undoubtedly identify an issuing authority as legitimate.

b) Initiate contact with the issuing authority's initial contact point. Use this initial contact point to engage with the issuing authority in setting up a trusted communication channel and the protocol for its use. Sensitive material used to set up the trusted communication channel shall be exchanged using out-of-band communication. The protocol shall guarantee the integrity, authenticity, non-repudiation and confidentiality of the information to be exchanged.

After the trusted communication channel is established, the Initial Identity Validation is concluded and the parties can then start exchanging information required for the Master List.

The Master List Provider shall regularly check that each assigned Point of Contact is still valid and the established communication channel provides the security guarantees, and perform the necessary updates if any required.

C.1.3.3 Identification and authentication for re-key requests

In the context of the ML, a re-key request is understood as the process to add a new CA to the records of the issuing authority.

This process shall use the trusted communication channel established at the initial identity validation (C.1.3.2). If the trusted communication channel is not considered secure (for example, weak keys or algorithm in use, lost keys, key renewal, etc), a new trusted communication channel shall be established following the same requirements.

C.1.3.4 Identification and authentication of suspension requests

Suspension requests from issuing authorities shall be communicated through the trusted communication channels established at the initial identity validation process.(C.1.3.2).

The ML Provider is also reserved the right to proceed unilaterally with the suspension of a CA on the grounds of a documented process and criteria.

C.1.4 Master list life-cycle operational requirements

C.1.4.1 Certification authority application

The IA may apply to the ML Provider for inclusion of its CA into the ML.

The respective IA Point of Contact shall provide all requested information by the ML Provider, according to the established processes and through the trusted communication channel. The provided information shall be authentic, correct, complete and truthful.

The Terms and Conditions of the ML Service, if any, shall be made available to the applicant by the ML Provider.

C.1.4.2 Certification authority application processing

The ML Provider shall validate the information submitted by the IAPC. Should any of the acceptance checks and conditions fail, the ML Provider is reserved the right to terminate the application process.

The ML Provider shall keep records (see C.1.5.5) of the application analysis, processing, internal and external checks and results.

C.1.4.3 Certification authority application acceptance

If the application processing by the ML Provider approves the CA, it can be included into the ML for the corresponding issuing authority and published according to the timeline set by the ML Provider and agreed by the applicant.

C.1.4.4 Certification authority renewal

A CA renewal is understood as a request to include a new CA certificate with the same name, public key, and other information as the old one, but with a new, extended validity period and a new serial number.

CA renewals shall not be used.

C.1.4.5 Certification authority application re-key

A CA re-key is understood as a request to include a new CA certificate with a different public key (and serial number) while retaining the remaining contents of the old CA certificate. The new certificate may be assigned a different validity period, key identifiers, specify a different CRL distribution point, and/or be signed with a different key.

The old certificate may or may not be revoked, but shall not be further re-keyed, renewed, or modified.

ML Providers shall process CA re-keys as new CA applications and thus follow the definitions laid down on subclauses C.1.4.1 to C.1.4.3.

C.1.4.6 Certification authority application modification

Modifying a CA certificate is understood as a request to include a new certificate that has the same or a different key and a different serial number, and that differs in one or more other fields from the old certificate.

The old certificate may or may not be revoked, but shall not be further re-keyed, renewed, or modified.

ML Providers shall process CA modifications as new CA applications and thus follow the definitions laid down on subclauses C.1.4.1 to C.1.4.3.

C.1.4.7 Certification authority application suspension

The ML Provider shall suspend CA certificates in a timely manner based on authorized and validated requests from the issuing authority Point of Contact.

The ML Provider shall keep records (see C.1.5.5) of the analysis, processing, internal and external checks and results.

C.1.4.8 End of subscription

This policy does not define nor limits any particular form or nature of relation between ML Providers and issuing authorities, and ML Providers and subscribers. However, it assumes the existence of these Participants (C.1.1.3) and some form of relation is established between them.

In the event of termination of the relation between the ML Provider and the issuing authority, the ML Provider is allowed to remove/revoke/suspend or maintain unchanged the corresponding IACAs, according to the terms and conditions agreed between the parties, or at its sole discretion in its absence.

On the other side, in the event of termination of the relation between the ML Provider and a subscriber, the contents of the ML shall not be affected.

C.1.5 Facility, management and operational controls

C.1.5.1 Physical security controls

ML Provider's equipment shall be protected from unauthorized access while the cryptographic module (see C.1.6.2) is installed and activated. The ML Provider shall implement physical access controls to reduce the risk of equipment tampering even when the cryptographic module is not installed and activated. ML Provider cryptographic tokens shall be protected against theft, loss, and unauthorized use.

The following controls are required to be fulfilled:

a) Physical access to components of the ML Provider's system whose security is critical to the provision of its ML services shall be limited to authorized individuals.

b) Controls shall be implemented to avoid loss, damage or compromise of assets and interruption to business activities.

c) Controls shall be implemented to avoid compromise or theft of information and information processing facilities.

d) Components that are critical for the secure operation of the ML service shall be located in a protected security perimeter with physical protection against intrusion, controls on access through the security perimeter and alarms to detect intrusion.

e) The facilities concerned with ML generation and management (i.e. CAs status lifecycle management) shall be operated in an environment which physically protects the services from compromise through unauthorized access to systems or data.

f) Every entry to the physically secure area shall be subject to independent oversight and non-authorized person shall be accompanied by an authorized person whilst in the secure area.

g) Every entry and exit shall be logged and such access log shall be inspected periodically.

h) Physical protection shall be achieved through the creation of clearly defined security perimeters (i.e. physical barriers) around the ML generation and management services.

i) Any parts of the premises shared with other organizations shall be outside the perimeter of the ML generation management services.

j) Physical and environmental security controls shall be implemented to protect the facility housing system resources, the system resources themselves, and the facilities used to support their operation.

k) The ML Provider's physical and environmental security policy for systems concerned with ML generation and management services shall address the physical access control, natural disaster protection, fire safety factors, failure of supporting utilities (e.g. power, telecommunications), structure collapse, plumbing leaks, protection against theft, breaking and entering, and disaster recovery.

l) Controls shall be implemented to protect against equipment, information, media and software relating to the ML Provider's services being taken off-site without authorization.

m) Other functions relating to ML Provider's operations may be supported within the same secured area provided that the access is limited to authorized personnel.

C.1.5.2 Procedural controls

ML Providers shall implement security measures in order to protect the authenticity, integrity and confidentiality of their data and the accurate functionality of their IT systems.

The following controls are required to be fulfilled:

a) The ML Provider shall administer user access of operators, administrators and system auditors.

b) The administration shall include user account management and timely modification or removal of access.

c) Access to information and application system functions shall be restricted in accordance with the access control policy.

d) The ML Provider's system shall provide sufficient computer security controls for the separation of trusted roles identified in ML Provider's practices, including the separation of security administration and operation functions. Particularly, use of system utility programs shall be restricted and controlled.

e) ML Provider's personnel shall be identified and authenticated before using critical applications related to the service.

f) ML Provider's personnel shall be accountable for their activities.

g) Activation of the ML signing key shall be under at least dual control by authorized, trusted personnel such that one person alone cannot activate the ML creation system on his/her own.

C.1.5.3 Personnel controls

The ML Provider shall ensure that employees and contractors support the trustworthiness of the ML Provider's operations.

The following controls are required to be fulfilled:

a) The ML Provider shall employ staff and, if applicable, subcontractors, who possess the necessary expertise, reliability, experience, and qualifications and who have received training regarding security and personal data protection rules as appropriate for the offered services and the job function.

b) ML Provider's personnel should be able to fulfil the requirement of "expert knowledge, experience and qualifications" through formal training and credentials, or actual experience, or a combination of the two.

c) This should include regular (at least every 12 months) updates on new threats and current security practices.

d) Appropriate disciplinary sanctions shall be applied to personnel violating ML Provider's policies or procedures.

e) Security roles and responsibilities, as specified in the ML Provider's information security policy, shall be documented in job descriptions or in documents available to all concerned personnel.

f) Trusted roles, on which the security of the ML Provider's operation is dependent, shall be clearly identified.

g) Trusted roles shall be named by the management.

h) Trusted roles shall be accepted by the management and the person to fulfil the role.

i) ML Provider's personnel (both temporary and permanent) shall have job descriptions defined from the view point of roles fulfilled with segregation of duties and least privilege, determining position sensitivity based on the duties and access levels, background screening and employee training and awareness.

j) Where appropriate, job descriptions shall differentiate between general functions and ML Provider's specific functions. These should include skills and experience requirements.

k) Personnel shall exercise administrative and management procedures and processes that are in line with the ML Provider's information security management procedures.

l) Managerial personnel shall possess experience or training with respect to the ML service that is provided, familiarity with security procedures for personnel with security responsibilities and experience with information security and risk assessment sufficient to carry out management functions.

m) All ML Provider's personnel in trusted roles shall be free from conflict of interest that might prejudice the impartiality of the ML Provider's operations.

n) Trusted roles shall include roles that involve the following responsibilities:

    a) Security Officers: Overall responsibility for administering the implementation of the security practices.

    b) System Administrators: Authorized to install, configure, maintain and recover the ML Provider's trustworthy systems for service management.

    c) System Operators: Responsible for operating the ML Provider's trustworthy systems on a day-to-day basis. Authorized to perform system backup.

d) System Auditors: Authorized to view archives and audit logs of the ML Provider's trustworthy systems.

o) ML Provider's personnel shall be formally appointed to trusted roles by senior management responsible for security requiring the principle of "least privilege" when accessing or when configuring access privileges.

p) Personnel shall not have access to the trusted functions until the necessary checks are completed.

C.1.5.4 Audit logging procedures

The ML Provider shall record and keep accessible for an appropriate period of time, including after the activities of the ML Provider have ceased, all relevant information concerning data issued and received by the ML Provider, in particular, for the purpose of providing evidence in legal proceedings and for the purpose of ensuring continuity of the service.

The following controls are required to be fulfilled:

c) The confidentiality and integrity of current and archived records concerning operation of services shall be maintained.

d) Records concerning the operation of services shall be completely and confidentially archived in accordance with disclosed business practices.

e) Records concerning the operation of services shall be made available if required for the purposes of providing evidence of the correct operation of the services for the purpose of legal proceedings.

f) The precise time of significant ML Provider's environmental, key management and clock synchronization events shall be recorded.

g) The time used to record events as required in the audit log shall be synchronized with UTC at least once a day.

h) Records concerning services shall be held for a period of time as appropriate for providing necessary legal evidence and as notified in the ML Provider's terms and conditions.

i) The events shall be logged in a way that they cannot be easily deleted or destroyed (except if reliably transferred to long-term media) within the period of time that they are required to be held. Where possible, the security audit logs shall be automatically collected. Where this is not possible, a logbook, paper form, or other physical mechanism shall be used. All security audit logs, both electronic and non-electronic, shall be retained and made available during compliance audits.

C.1.5.4.1 Types of events recorded

The ML Provider shall record details of the actions taken to process a request and to issue a ML, including all information generated and documentation received in connection with the request; the time and date; and the personnel involved. The ML Provider shall make these records available to Auditors as proof of the ML Provider’s compliance with these requirements.

The ML Provider shall record at least the following events:

a) ML signing key lifecycle management events, including:

   a. Key generation, backup, storage, recovery, archival, and destruction; and

   b. Cryptographic device lifecycle management events.

b) ML and IA lifecycle management events, including:

   c. CA application, re-key requests, and suspension;

   d. All verification activities stipulated in these Requirements;

e. Date, time, phone number used, persons spoken to, and end results of verification telephone calls;

f. Acceptance and rejection of CA applications; and

g. Issuance of MLs.

c) Security events, including:

h. Successful and unsuccessful ML Provider's system access attempts;

i. ML Provider's and security system actions performed;

j. Security profile changes;

k. System crashes, hardware failures, and other anomalies;

l. Firewall and router activities; and

m. Entries to and exits from the ML Provider facility.

Log entries shall include the following elements:

d) Date and time of entry;

e) Identity of the person making the journal entry; and

f) Description of the entry.

C.1.5.5 Records archival

The ML Provider shall retain the following for at least seven years after any CA (accepted or not) based on these records ceases to be valid:

a) log of all events relating to the life cycle of keys managed by the ML Provider's system

b) documentation and other evidence as referred in C.1.4.2.

C.1.5.6 Key changeover

To minimize risk from compromise of a ML Provider's private signing key, that key may be changed often. From that time on, only the new key should be used to sign MLs. If the old private key is necessary during a limited period of time to keep signing MLs and allow the migration for Relying Parties with legacy systems, the old key shall be retained and protected. Once the old private signing key is not needed anymore, it may be destroyed.

The ML Provider's signing key shall have a validity period as described in C.1.7.3.

When a ML Provider updates its private signature key and thus generates a new public key, the ML Provider shall notify all subscribers that rely on its respective MLs that it has been changed. The ML Provider shall provide the new public key through secure means (e.g. trusted communication channel established with subscribers/Relying Parties, provision of key rollover certificates – new public key is signed by the old private key, and vice versa).

C.1.5.7 Compromise and disaster recovery

C.1.5.7.1 Incident and compromise handling procedures

System activities concerning access to IT systems, use of IT systems, and service requests shall be monitored.

The following controls are required to be fulfilled:

a) Monitoring activities should take account of the sensitivity of any information collected or analysed.

b) Abnormal system activities that indicate a potential security violation, including intrusion into the ML Provider's network, shall be detected and reported as alarms.

c) The ML Provider shall monitor the following events:

   a. start-up and shutdown of the logging functions; and

   b. availability and utilization of needed services with the ML Provider's network

d) The ML Provider shall act in a timely and co-ordinated manner in order to respond quickly to incidents and to limit the impact of breaches of security.

e) The ML Provider shall appoint trusted role personnel to follow up on alerts of potentially critical security events and ensure that relevant incidents are reported in line with the ML Provider's procedures.

f) The ML Provider shall establish procedures to notify the appropriate parties in line with the applicable regulatory rules of any breach of security or loss of integrity that has a significant impact on the ML service provided and on the personal data maintained therein within 24 hours of the breach being identified.

g) Where the breach of security or loss of integrity is likely to adversely affect a natural or legal person to whom the ML service has been provided, the ML Provider shall also notify the natural or legal person of the breach of security or loss of integrity without undue delay.

h) The ML Provider's systems shall be monitored including the monitoring or regular review of audit logs to identify evidence of malicious activity implementing automatic mechanisms to process the audit logs and alert personnel of possible critical security events.

i) The ML Provider shall address any critical vulnerability not previously addressed by the ML Provider, within a period of 48 hours after its discovery.

j) For any vulnerability, given the potential impact, the ML Provider may either choose to:

   c. create and implement a plan to mitigate the vulnerability; or

   d. document the factual basis for the ML Provider's determination that the vulnerability does not require remediation.

k) Incident reporting and response procedures shall be employed in such a way that damage from security incidents and malfunctions are minimized.

C.1.5.7.2 Computing resources, software, and/or data are corrupted

The following controls are required to be fulfilled:

a) ML Provider's systems data necessary to resume CA operations shall be backed up and stored in safe places, preferably also remote, suitable to allow the ML Provider to timely go back to operations in case of incident/disasters.

b) Back-up copies of essential information and software should be taken regularly.

c) Adequate back-up facilities should be provided to ensure that all essential information and software can be recovered following a disaster or media failure.

d) Back-up arrangements should be regularly tested to ensure that they meet the requirements of business continuity plans.

e) Backup and restore functions shall be performed by the relevant trusted roles specified in C.1.5.3.

f) For information requiring dual control for management, for example keys, dual control shall be applied to recovery.

C.1.5.7.3 ML Provider private key compromise procedures

The following controls are required to be fulfilled in case of a private key compromise:

a) The ML Provider's business continuity plan (or disaster recovery plan) shall address the compromise, loss or suspected compromise of a ML Provider's private key as a disaster.

b) The processes planned as per the previous requirement shall be in place.

c) Following a disaster, the ML Provider shall, where practical, take steps to avoid repetition of a disaster.

d) In the case of compromise as a minimum:

   a. The ML Provider shall inform the following of the compromise: all issuing authorities Point of Contacts and subscribers and other entities with which the ML Provider has agreements or other form of direct established relations, among which Relying Parties and ML Providers; and

   b. The ML Provider shall indicate that MLs issued using this private key may no longer be valid.

Furthermore, the following controls are required to be fulfilled in case of an algorithm compromise:

a) Should any of the algorithms, or associated parameters, used by the ML Provider or its issuing authorities Point of Contacts and/or subscribers become insufficient for its remaining intended usage, then the ML Provider shall inform all of them with whom it has agreement or other form of established relations.

b) Should any of the algorithms, or associated parameters, used by the ML Provider or its subscribers become insufficient for its remaining intended usage, then the ML Provider shall plan the transition to a new stronger algorithm and execute it the earliest possible time.

C.1.5.7.4 Business continuity capabilities after a disaster

The following controls are required to be fulfilled:

a) The ML Provider shall define and maintain a continuity plan to enact in case of a disaster.

b) In the event of a disaster, including compromise of a private signing key or compromise of some other credential of the ML Provider, operations shall be restored within the delay established in the continuity plan, having addressed any cause for the disaster which may recur (e.g. a security vulnerability) with appropriate remediation measures.

C.1.5.8 Master list termination

Potential disruptions to subscribers and Relying Parties shall be minimized as a result of the cessation of the ML Provider's services, and in particular continued maintenance of information required to verify the correctness of ML services shall be provided.

Furthermore, the following controls are required to be fulfilled:

a) The ML Provider shall have an up-to-date termination plan.

b) Before the ML Provider terminates its services, at least the following procedures apply:

   a. Before the ML Provider terminates its services, the ML Provider shall inform the following of the termination: all issuing authorities Point of Contacts and subscribers and other entities with which the ML Provider has agreements or other form of established relations, among which Relying Parties and ML Providers.

b. Before the ML Provider terminates its services, the ML Provider shall terminate authorization of all subcontractors, if any, to act on behalf of the ML Provider in carrying out any functions relating to the processing and/or dissemination of the ML.

c. Before the ML Provider terminates its services, the ML Provider shall transfer obligations to a reliable party for maintaining all information necessary to provide evidence of the operation of the ML Provider for a reasonable period, unless it can be demonstrated that the ML Provider does not hold any such information. The minimum information set is composed of:

   i. registration information;

   ii. event log archives.

d. Before the ML Provider terminates its services, the ML Provider's private keys, including backup copies, shall be destroyed, or withdrawn from use, in a manner such that the private keys cannot be retrieved.

e. Before the ML Provider terminates its services, where possible ML Provider should make arrangements to transfer provision of ML services for its existing subscribers and Issuing Authorities Point of Contacts to another ML Provider.

c) The ML Provider shall have an arrangement to cover the costs to fulfil these minimum requirements in case the ML Provider becomes bankrupt or for other reasons is unable to cover the costs by itself, as far as possible within the constraints of applicable legislation regarding bankruptcy.

d) The ML Provider shall state in its practices the provisions made for termination of service. This shall include:

   f. notification of affected entities; and

   g. where applicable, transferring the ML Provider's obligations to other parties.

e) The ML Provider shall maintain or transfer to a reliable party its obligations to make available its public key or its history of MLs to subscribers and Relying Parties for a reasonable period.

C.1.6 Technical security controls

C.1.6.1 Key pair generation and installation

Appropriate security controls shall be in place for the management of any cryptographic keys and any cryptographic devices throughout their lifecycle.

The following controls are required to be fulfilled:

a) The ML signing key pair generation shall be undertaken in a physically secured environment (see C.1.5.1) by personnel in trusted roles (see C.1.5.3).

b) The ML Provider key pair used for signing MLs shall be created under, at least, dual control.

c) The number of personnel authorized to carry out ML signing key pair generation shall be kept to a minimum and be consistent with the ML Provider's practices.

d) ML Signing key pair generation shall be performed using an algorithm as specified in C.1.7.3.

e) The selected key length and algorithm for ML signing key are specified in C.1.7.3.

f) Before expiration of its ML certificate which is used for signing MLs, in case of continuing with the service, the ML Provider shall generate a new key pair and obtain a corresponding certificate for signing subject key pairs, and shall apply all necessary actions to avoid disruption to the operations of any entity that may rely on the ML certificate.

g) Before expiration of its ML certificate which is used for signing MLs, in case of continuing with the service, the new ML certificate shall also be issued and distributed in accordance with this document.

h) The operations described in clauses f) and g) above should be performed with a suitable interval between certificate expiry date and the last certificate signed to allow all parties that have relationships with the ML Provider (subscribers, Relying Parties, Issuing Authorities, etc.) to be aware of this key changeover and to implement the required operations to avoid inconveniences and malfunctions. This does not apply to a ML Provider which will cease its operations before its own certificate-signing certificate expiration date.

i) The ML Provider shall have a documented procedure for conducting generation of ML signing key pairs. Such procedure shall indicate, at least, the following:

   a. roles participating in the ceremony (internal and external from the organization);

   b. functions to be performed by every role and in which phases;

   c. responsibilities during and after the ceremony; and

   d. requirements of evidence to be collected of the ceremony.

j) The ML Provider shall produce a report proving that the ceremony, as in i) above, was carried out in accordance with the stated procedure and that the integrity and confidentiality of the key pair was ensured.

k) CA signature verification (public) keys of the ML signing certificate shall be available to subscribers and Relying Parties in a manner that assures the integrity of the CA public key and authenticates its origin.

C.1.6.2 Private key protection and cryptographic module engineering controls

The following controls are required to be fulfilled:

a) ML Provider's signing key pair generation shall be carried out within a secure cryptographic device which is a trustworthy system which:

   a. is assured to EAL 4 or higher in accordance with ISO/IEC 15408, or equivalent national or internationally recognized evaluation criteria for IT security provided this is a security target or protection profile which meets the requirements of this document, based on a risk analysis and taking into account physical and other non-technical security measures; or

   b. meets the requirements identified in ISO/IEC 19790 or FIPS PUB 140-2 level 3.

b) The secure cryptographic device shall be operated in its configuration as described in the appropriate certification guidance documentation or in an equivalent configuration which achieves the same security objective.

c) The ML private signing key shall be held and used within a secure cryptographic device meeting the requirements of clauses 1) and 2) above.

d) If and when outside the secure cryptographic device, the ML signing private key shall be protected in a way that ensures the same level of protection as provided by the secure cryptographic device.

e) The ML signing private key may be backed up, stored and recovered only by personnel in trusted roles (see C.1.5.3) using, at least, dual control in a physically secured environment (see C.1.5.1)

f) The number of personnel authorized to carry out the ML signing private key back up, storage and recovery shall be kept to a minimum and be consistent with the ML Provider's practices.

g) Copies of the ML private signing keys shall be subject to the same or greater level of security controls as keys currently in use.

h) Where the ML signing private keys and any copies are stored in a dedicated secure cryptographic device, access controls shall be in place to ensure that the keys are not accessible outside this device.

i) The secure cryptographic device shall not be tampered with during shipment.

j) The secure cryptographic device shall not be tampered with while stored.

k) The secure cryptographic device shall be functioning correctly

l) The CA private signing keys stored on the CA's secure cryptographic device shall be destroyed upon device retirement.

C.1.6.3 Other aspects of key pair management

The ML Provider shall use appropriately the ML private signing keys.

The following controls are required to be fulfilled:

a) The ML Provider shall not use the ML signing private keys beyond the end of their life cycle.

b) ML signing key(s) used for generating MLs as defined in C.1.7.1 shall not be used for any other purpose.

c) The ML signing keys shall only be used within physically secure premises.

d) The use of the ML's private key shall be compatible with the hash algorithm, the signature algorithm and signature key length used for generating MLs (defined in C.1.7.3).

e) All copies, if any, of the ML signing private keys shall be destroyed at the end of their life cycle.

C.1.6.4 Activation data

The installation, activation and recovery of the ML signing key pairs in a secure cryptographic device shall require simultaneous control of at least two trusted employees, for example, using m-of-n authentication mechanisms.

Data used to unlock private keys shall be protected from disclosure by a combination of cryptographic and physical access control mechanisms. Activation data shall be either:

— memorized; or

— biometric in nature; or

— recorded and secured at the level of assurance associated with the activation of the cryptographic module, and shall not be stored with the cryptographic module.

C.1.6.5 Computer security controls

The ML Provider's system access shall be limited to authorized individuals.

The following controls are required to be fulfilled:

a) Controls (e.g. firewalls) shall protect the ML Provider's internal network domains from unauthorized access including access by subscribers and third parties.

b) Firewalls should also be configured to prevent all protocols and accesses not required for the operation of the ML Provider.

c) Sensitive data shall be protected against being revealed through re-used storage objects (e.g. deleted files) being accessible to unauthorized users.

d) Local network components (e.g. routers) shall be kept in a physically and logically secure environment.

e) Local network components (e.g. routers) configurations shall be periodically checked for compliance with the requirements specified by the ML Provider.

f) The ML Provider shall enforce multi-factor authentication for all accounts capable of directly causing ML issuance.

g) Dissemination application shall enforce access control on attempts to add or delete MLs and modify other associated information.

h) Continuous monitoring and alarm facilities shall be provided to enable the ML Provider to detect, register and react in a timely manner upon any unauthorized and/or irregular attempts to access its resources.

C.1.6.6 Life cycle security controls

The ML Provider shall use trustworthy systems and products that are protected against modification and ensure the technical security and reliability of the processes supported by them.

The following controls are required to be fulfilled:

a) An analysis of security requirements shall be carried out at the design and requirements specification stage of any systems development project undertaken by the ML Provider or on behalf of the ML Provider to ensure that security is built into IT systems.

b) Change control procedures shall be applied for releases, modifications and emergency software fixes of any operational software and changes to the configuration which applies the ML Provider's security policy.

c) The procedures shall include documentation of the changes.

d) The integrity of ML Provider's systems and information shall be protected against viruses, malicious and unauthorized software.

e) Media used within the ML Provider's systems shall be securely handled to protect media from damage, theft, unauthorized access and obsolescence.

f) Media management procedures shall protect against obsolescence and deterioration of media within the period of time that records are required to be retained.

g) Procedures shall be established and implemented for all trusted and administrative roles that impact on the provision of services.

h) The ML Provider shall specify and apply procedures for ensuring that:

   a. security patches are applied within a reasonable time after they come available;

   b. security patches are not applied if they introduce additional vulnerabilities or instabilities that outweigh the benefits of applying them; and

   c. the reasons for not applying any security patches are documented.

i) Capacity demands shall be monitored and projections of future capacity requirements shall be made to ensure that adequate processing power and storage are available.

C.1.6.7 Network security controls

The ML Provider shall protect its network and systems from attack.

The following controls are required to be fulfilled:

a) The ML Provider shall segment its systems into networks or zones based on risk assessment considering functional, logical, and physical (including location) relationship between trustworthy systems and services.

b) The ML Provider shall apply the same security controls to all systems co-located in the same zone.

c) The ML Provider shall restrict access and communications between zones to those necessary for the operation of the ML Provider.

d) The ML Provider shall explicitly forbid or deactivate not needed connections and services.

e) The ML Provider shall review the established rule set on a regular basis.

f) The ML Provider shall keep all systems that are critical to the ML Provider's operation in one or more secured zone(s).

g) The ML Provider shall separate dedicated network for administration of IT systems and ML Provider 's operational network.

h) The ML Provider shall not use systems used for administration of the security policy implementation for other purposes.

i) The ML Provider shall separate the production systems for the ML Provider's services from systems used in development and testing (e.g. development, test and staging systems).

j) The ML Provider shall establish communication between distinct trustworthy systems only through trusted channels that are logically distinct from other communication channels and provide assured identification of its end points and protection of the channel data from modification or disclosure.

k) If a high level of availability of external access to the ML service is required, the external network connection should be redundant to ensure availability of the services in case of a single failure.

l) The ML Provider shall undergo or perform a regular vulnerability scan on public and private IP addresses identified by the ML Provider and record evidence that each vulnerability scan was performed by a person or entity with the skills, tools, proficiency, code of ethics, and independence necessary to provide a reliable report.

m) The ML Provider shall undergo a penetration test on the ML Provider's systems at set up and after infrastructure or application upgrades or modifications that the ML Provider determines are significant.

n) The ML Provider shall record evidence that each penetration test was performed by a person or entity with the skills, tools, proficiency, code of ethics, and independence necessary to provide a reliable report.

o) The ML Provider shall maintain and protect all ML systems in at least a secure zone and shall implement and configure a security procedure that protects systems and communications between systems inside secure zones and high security zones.

p) The ML Provider shall configure all ML systems by removing or disabling all accounts, applications, services, protocols, and ports that are not used in the ML's operations.

q) The ML Provider shall grant access to secure zones and high security zones to only trusted roles.

r) The ML issuing system shall be in a high security zone.

C.1.6.8 Timestamping

The following controls are required to be fulfilled:

a) Asserted times shall be accurate to within three minutes.

b) Electronic or manual procedures may be used to maintain system time.

c) Clock adjustments are auditable events.

C.1.7 Master list and master list signer certificate profiles

C.1.7.1 Master list CDDL profile

The Master Lift profile uses a COSE_Sign structure with the X509 (chain) element from draft-iets-cose-x509-04: *CBOR Object Signing and Encryption (COSE): Headers for carrying and referencing X.509 certificates* to the master certificate.

The payload shall use the following CDDL structure:

```
MasterList = {
   "type" : tstr           ; currently "1.0"
   "version" : tstr        : currenlty "1.0"
   "date" : tdate          ; date-time according to RFC 7049
   ? "nextUpdate" : tdate  ; date-time according to RFC 7049
   "certificateInfo : [+ CertificateInfo]
}

CertificateInfo = {
   "issuingCountry": tstr  ; ISO3166-1 or ISO3166-2 depending on the issuer
   ? "issuingAuthority": tstr
   ? "stateOrProvinceName": tstr
   ? "docType": tstr
   "certStructure" : CertStructure
}

CertStructure = {
   ? "DN":  bstr
   "ski": bstr
   "certificate" : bstr
}
```

C.1.7.2 Master list CMS profile

Master Lists may be implemented as instances of the `ContentInfo` type specified in RFC 5652. The `ContentInfo` shall contain exactly one instance of the `SignedData` type as specified in Table C.1 and shall not contain any other values. These master lists shall be encoded using the Distinguished Encoding Rules (DER) as specified in RFC 5652.

Table C.1 — The `SignedData` type for master lists

Value	Presence	Comments
`SignedData`	M	
`version`	M	Shall be set to 3
`digestAlgorithms`	M	Should include the `digestAlgorithm` used in the `SignerInfo`
`encapContentInfo`	M	
`eContentType`	M	Shall be set to the OID `id-idl-ml`
Presence: M/R/O/X: The presence of the value is mandatory (M), recommended (R), optional (O) or the value shall not be present (X)		

Table C.1 *(continued)*

Value	Presence	Comments
eContent	M	Shall contain the encoded contents of an IacaMasterList
certificates	M	Shall contain the master list signer certificate and should contain the certificate of the CA that has issued the master list signer certificate
crls	X	
signerInfos	M	This field shall contain exactly 1 signerinfo.
SignerInfo	M	
version	M	Depends on the choice in the sid field. For details see RFC 5652.
sid	M	It is recommended that this field makes use of the master list signer certificate's subjectKeyIdentifier instead of issuerandSerialNumber.
digestAlgorithm	M	One of the following algorithms shall be used: SHA256, SHA384 or SHA512 The value shall be encoded according to RFC 5754.
signedAttrs	M	Shall contain the signingTime attribute, see RFC 5652. May contain the nextUpdate attribute.
signatureAlgorithm	M	One of the following OIDs shall be used: 1.2.840.10045.4.3.2 (ECDSA-with SHA256) 1.2.840.10045.4.3.3 (ECDSA-with SHA384) 1.2.840.10045.4.3.4 (ECDSA with SHA512)
signature	M	The result of the signature generation process
unsignedAttrs	X	
Presence: M/R/O/X: The presence of the value is mandatory (M), recommended (R), optional (O) or the value shall not be present (X)		

The IacaMasterList is defined as follows:

```
IacaMasterList
{ iso(1) standard(0) driving-licence (18013) part-5(5) master-list(3)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
IMPORTS
-- Imports from RFC 5280 Appendix A.1
      Certificate
      FROM PKIX1Explicit88
         { iso(1) identified-organization(3) dod(6) internet(1) security(5)
         mechanisms(5) pkix(7) mod(0) pkix1-explicit(18) };
-- IACA Master List
IacaMasterListVersion ::= INTEGER {v1(0)}
IacaMasterList ::= SEQUENCE {
      version IacaMasterListVersion,
      certList SET OF Certificate }
-- Object Identifiers
id-idl-ml OBJECT IDENTIFIER ::= { iso(1) standard(0) driving-licence (18013)
part-5(5) 3}
END
```

The next update attribute encodes the timestamp at which the Master List Provider expects to resign (and potentially update) the master list. The following OID identifies the next update attribute:

```
id-idl-ml-nextUpdate OBJECT IDENTIFIER ::= { iso(1) standard(0) driving-licence
(18013) part-5(5) master-list(3) 2}
```

The next update attribute values have ASN.1 type `NextUpdate`:

```
NextUpdate ::= Time
Time ::= CHOICE {
      utcTime UTCTime,
      generalizedTime GeneralizedTime }
```

C.1.7.3 Master list signer certificate profile

This certificate profile defines the baseline for the ML signer certificates, establishing the minimum security parameters (key lengths, algorithms, policy IDs, etc). CAs issuing certificates to ML Providers may choose to use equivalent or higher parameters, as well as other certificate fields and extensions that do not limit or reduce the overall security level. See Table C.2 for details.

The field type column indicates whether an element is mandatory (m), optional (o) or non-critical (mc).

Table C.2 — Master list signer certificate profile

Certificate Component	Section in RFC 5280	Field Type	Comments
Version	4.1.2.1	m	Shall be `v3`
Serial Number	4.1.2.2	m	Non-sequential positive, non-zero integer, maximum 20 octets.
Signature	4.1.2.3	m	Value shall match the OID in Signature Algorithm (below)
Issuer	4.1.2.4	m	According to the Certification Authority issuing the ML signer certificate.
Validity	4.1.2.5	m	
Not Before		m	Date/time when the certificate was issued.
Not After		m	Maximum of 39 months after "Not Before" date
Subject	4.1.2.6	m	Minimum required fields. Other may be present (e.g. Serial Number, State, Organization Unit, etc.)
Country (C)		m	Country code of jurisdiction of ML Provider. Encoded as `PrintableString`.
Organization (O)		m	Full registered name of the ML Provider. Encoded as `UTF8String`.
Common Name (CN)		m	Name under which ML Provider operates ML service and is commonly known. Encoded as `UTF8String`.
Subject Public Key Info	4.1.2.7	m	
algorithm		m	`1.2.840.10045.2.1` (Elliptic curve)
parameters		m	Implicitly specify parameters through an OID associated with a Brainpool curve specified in RFC 5639 or a NIST approved curve referenced in 800-78-4 (`{ EC Parameters namedCurve }`): `1.3.36.3.3.2.8.1.1.7` (brainpoolP256r1) `1.3.36.3.3.2.8.1.1.9` (brainpoolP320r1) `1.3.36.3.3.2.8.1.1.11` (brainpoolP384r1) `1.3.36.3.3.2.8.1.1.13` (brainpoolP512r1) `1.2.840.10045.3.1.7` (Curve P-256) `1.3.132.0.34` (Curve P-384) `1.3.132.0.35` (Curve P-521)
subjectPublicKey		m	The public key shall be encoded in uncompressed form.
X.509v3 Extensions	4.1.2.9	m	
Authority Key Identifier	4.2.1.1	m	
keyIdentifier		m	Same value as the Subject Key Identifier of the issuer's certificate
Subject Key Identifier	4.2.1.2	m	SHA-1 hash of the subject public key `BIT STRING` value (excluding tag, length, and number of unused bits)
Key Usage	4.2.1.3	mc	
Digital Signature			`0`

Table C.2 *(continued)*

Certificate Component	Section in RFC 5280	Field Type	Comments
Non Repudiation			1
Key Encipherment			0
Data Encipherment			0
Key Agreement			0
Key Certificate Signature			0
CRL Signature			0
Encipher Only			0
Decipher Only			0
Certificate Policies	4.2.1.4	m	The CA may add other Policies.
policyIdentifier		m	1.0.18013.5.3.1 (ISO/IEC 18013-5 ML Policy)
policyQualifiers		m	policyQualifierID: 1.3.6.1.5.5.7.2.2 (userNotice) Explicit text encoded as UTF8String: Certificate issued in accordance with the Certificate Policy for ISO-compliant Driving Licence.
Basic Constraints	4.2.1.10	mc	
CA		m	FALSE
CRLDistributionPoints	4.2.1.13	m	The 'reasons' and 'cRL Issuer' fields shall not be used
distributionPoint		m	URI HTTP for CRL Distribution Point
Internet Certificate Extensions			
Authority Information Access	4.2.2.1	m	
Access Description OCSP			
accessMethod		m	1.3.6.1.5.5.7.48.1 (OCSP)
accessLocation		m	URI HTTP pointing to corresponding OCSP service
Access Description CA Issuers			
accessMethod		m	1.3.6.1.5.5.7.48.2 (CA Issuers)
accessLocation		m	URI HTTP pointing to corresponding issuing authority CA certificate
Signature Algorithm	4.1.1.2	m	Options: 1.2.840.10045.4.3.2 (ECDSA-with SHA256) 1.2.840.10045.4.3.3 (ECDSA-with SHA384) 1.2.840.10045.4.3.4 (ECDSA-with SHA512)
Signature Value	4.1.1.3	m	Value according to Signature Algorithm. By imprinting this signature, the CA certifies the binding between the public key material and the subject of the certificate, i.e. the ML Provider.

C.1.8 Compliance audit and other assessment

It is beyond the scope of this Policy to establish an auditing scheme for ML Providers. Nevertheless, it is understood that ML Providers should be able to publicly demonstrate compliance to this Policy and the security requirements.

As a minimum, ML Providers shall conduct self-audits on a periodic basis (at least yearly) to assess compliance to this Policy.

An independent third party assessment can be achieved by an ML Provider based on the following principles:

— Auditor qualification: ML Provider shall select an independently acting and accredited company/ organisation ("Auditing Body") or certified auditors to audit the ML Service according to this Policy. The Auditing Body shall either be accredited for this purpose by its national accreditation body or authorized by a responsible government office.

— Audit basis: The Audit shall be based on ISO/IEC 27001 and 27002.

— Checking requirement realisation: The audit and control shall not only check that procedural security controls are specified but also that they are adhered to in practice. This also includes the initial identity validation, the receipt of IACA applications and the suspension/removal procedure for IACAs.

— Iteration of audits and controls: Audits and controls shall be performed at least every three years. The Auditing Body and the ML Provider shall carry out a review at least once a year by a team of one or more auditors to ensure on going compliance with this Policy.

— Being not conformant: In the event that an audit indicates that the ML Provider is not conformant to this Policy, or its certification becomes invalid or expires, the ML Provider shall notify its Point of Contacts of Issuing Authorities and subscribers.

— Availability of audit results: The certificate of conformity may be made available to Issuing Authorities, subscribers, Relying Parties and other possible stakeholders.

It is recommended that a ML Provider implements an Information Security Management System (ISMS) for its ML Service in accordance to ISO/IEC 27001. The ISMS is based on an ISMS policy of which its scope is defined by this Policy and, if applicable, the associated Practice Statement.

Annex D
(informative)

Data structure examples

D.1 Introduction

This annex contains examples for different structures used in the document. Since CBOR results in binary structures, a diagnostic notation will be used together with the binary encoding, whenever CBOR examples are made in this annex.

D.2 Driving privileges

An example of the driving privileges structure, where the mDL Holder has driving privileges for three vehicle categories (A,B,C). Only C has an expiration date and the mDL Holder has no restrictions or conditions.

```
[
   {
      "vehicle_category_code": "A", "issue_date": 18013("2017-01-01")
   },
   {
      "vehicle_category_code": "B", "issue_date": 18013("2017-01-01")
   },
   {
      "vehicle_category_code": "C", "issue_date": 18013("2017-01-01"), "expiry_date":
   18013("2020-01-01")
   }
]
```

This example is represented as the following byte string when encoded with CBOR:

```
83                                      # array(3)
   A2                                   # map(2)
      75                                # text(21)
         76656869636C655F63617465676F72795F636F6465 # "vehicle_category_code"
      61                                # text(1)
         41                             # "A"
      6A                                # text(10)
         69737375655F64617465           # "issue_date"
      D9 465D                           # tag(18013)
         6A                             # text(10)
            323031372D30312D3031        # "2017-01-01"
   A2                                   # map(2)
      75                                # text(21)
         76656869636C655F63617465676F72795F636F6465 # "vehicle_category_code"
      61                                # text(1)
         42                             # "B"
      6A                                # text(10)
         69737375655F64617465           # "issue_date"
      D9 465D                           # tag(18013)
         6A                             # text(10)
            323031372D30312D3031        # "2017-01-01"
   A3                                   # map(3)
      75                                # text(21)
         76656869636C655F63617465676F72795F636F6465 # "vehicle_category_code"
      61                                # text(1)
         43                             # "C"
      6A                                # text(10)
         69737375655F64617465           # "issue_date"
      D9 465D                           # tag(18013)
```

```
         6A                              # text(10)
            323031372D30312D3031         # "2017-01-01"
      6B                                 # text(11)
         6578706972795F64617465          # "expiry_date"
      D9 465D                            # tag(18013)
         6A                              # text(10)
            323032302D30312D3031         # "2020-01-01"
```

D.3 Device engagement

An example of the device engagement structure is as follows:

```
[
  "1.0",
  [
    1,
    24(

h'A401022001215820E4706DE318A40D0BD8648B7907B0283F7445370241FF1FDD77F08D6A598BE90222582079
767FBE391223F61DCD5E980133A035B4918F6F9DE41B3CEB8D801860DF8859'
    )
  ],
  [
    [
      2,
      1,
      {
        1: true,
        11: h'00000005000010008000008 05F9B34FF'
      }
    ]
  ],
  {

  },
  [

  ],
  {
    "value": "testValue"
  }
]
```

The same CBOR encoded example is represented as the following byte string:

```
8663312e308201d818584ba401022001215820e4706de318a40d0bd8648b7907b0283f7445370241ff1fdd77f0
8d6a598be90222582079767fbe391223f61dcd5e980133a035b4918f6f9de41b3ceb8d801860df885981830201
a201f50b5000000005000010008000008 05f9b34ffa080a16576616c7565697465737456616c7565
```

D.4 Data retrieval

D.4.1 Offline retrieval

D.4.1.1 mDL Reader Request

An example of the mDL Reader Request is as follows:

```
{
  "version": "1.0",
  "docRequests": [
    {
      "itemsRequest": 24(

h'A267646F6354797065756F72672E69736F2E31383031332E352E312E6D444C6A6E616D65537061636573A271
6F72672E69736F2E31383031332E352E31A46A676976656E5F6E616D65F56A62697274685F64617465F56A6973
7375655F64617465F568706F727472616974F478186F72672E69736F2E31383031332E352E312E55746F706961
```

```
A16855746F7069614944F5'
      )
    }
  ]
}

Where the itemsRequest translates to:

{
   "docType": "org.iso.18013.5.1.mDL",
   "nameSpaces":
   {
      "org.iso.18013.5.1":
      {
         "given_name": true,
         "birth_date": true,
         "issue_date": true,
         "portrait": false
      },
      "org.iso.18013.5.1.Utopia":
      {
         "UtopiaID": true
      }
   }
}
```

The same example data encoded as CBOR bytes:

```
a26776657273696f6e63312e306b646f6352657175657373747381a16c6974656d7352657175657374d8185891a2
67646f6354797065756f72672e69736f2e31383031332e352e312e6d444c6a6e616d65537061636573a2716f72
672e69736f2e31383031332e352e31a46a676976656e5f6e616d65f56a62697274685f64617465f56a69737375
655f64617465f568706f727472616974f478186f72672e69736f2e31383031332e352e312e55746f706961a168
55746f7069614944f5
```

D.4.1.2 mDL Response

An example of the mDL Response is as follows:

```
{
  "version": "1.0",
  "documents": [
    {
      "org.iso.18013.5.1.mDL": {
        "issuerSigned": {
          "nameSpaces": {
            "org.iso.18013.5.1": [
              24(

h'A468646967657374494402667261 6E646F6D58207EF5D6FC314C8E6B91343741B2A5CA6835BF52728ED64BF7
73D020075C5556C771656C656D656E744964656E7469666965726A676976656E5F6E616D656C656C656D656E74
56616C756563546F6D'
              ),
              24(

h'A468646967657374494403667261 6E646F6D5004FA3F77BC7D3F65AD8E71E8BCA98D7B71656C656D656E744
964656E7469666965726A62697274685F646174656C656C656D656E7456616C7565D9465D6A313939392D303
32D3134'
              ),
              24(

h'A468646967657374494404667261 6E646F6D50635DCE207F4BA4FFF087AB1D427B663071656C656D656E7449
64656E7469666965726A69737375655F646174656C656C656D656E7456616C7565C074323031372D30372D3134
5430303A30303A30305A'
              ),
              24(

h'A468646967657374494413667261 6E646F6D5820E8FC51B0AE7A73D23D91B5333512BAEAC318B3EDF5975A88
8890126181FAC4F771656C656D656E744964656E7469666965726870 6F7274726169746C656C656D656E745661
6C7565590412FFD8FFE000104A4649460001010100900090000FFDB004300130D0E110E0C13110F1115141317
1D301F1D1A1A1D3A2A2C2330453D4947443D43414C566D5D4C51685241435F82606871757B7C7B4A5C86908577
```

```
8F6D787B76FFDB0043011415151D191D381F1F38764F434F7676767676767676767676767676767676767676767676-
76767676767676767676767676767676767676767676767676767676767676FFC000110800180064030122000 2
1101031101FFC4001B000003010003010000000000000000000000050604010203 07FFC400321000010303
030205020309000000000000010203040005110612211331141551617122410781A1163542527391B2C1F1
FFC400150101010000000000000000000000000000000001FFC4001A1101010100030100000000000000000000
014111213161FFDA000C03010002110311003F00A5BBDE22DA2329C7D692BC7D0D03F52CFB0FF75E7A7EF3E770
9723A1D0DAE146DDFBB3C039CE07AD2BD47A7E32DBB8DD1D52D6EF4B284F64A480067DFB51F87FFB95FF00EB9F
F14D215DE66AF089CE44B7DBDE9CB6890A2838EDDF18078F7ADD62D411EF4DB9B10A65D6B95A147381EA0D495B
933275FE6BBA75C114104A8BA410413E983DFF004F5AF5D34B4B4CDE632D0BF1FD1592BDD91C6411F3934C2FA6
AF6B54975D106DCF4A65AE56E856001EBC03C7CE29DD9EEF1EF10FC447DC9DA76AD2AEE93537A1BA7E4F70DD8E
FF0057C6DFFB5E1A19854A83758E54528750946EC6704850CD037BCEB08B6D7D2CC76D3317FC7B5CC04FB67072
69C5C6E0C5B60AE549242123B0E493F602A075559E359970D98DB89525456B51C951C8AFA13EA8E98E3C596836
783D5C63F5A61A99FDB7290875DB4BE88AB384BBBBBFC7183FDEAA633E8951DB7DA396DC48524FB1A8BD611A5A
A2A2432F30AB420A7A6D3240C718CF031FA9EF4C9AD550205AA02951DF4A1D6C8421B015B769DB8C9229837EA2
BE8B1B0D39D0EBA9C51484EFDB8C0EFD8D258DAF3C449699F2EDBD4584E7AF9C64E3F96B9BEB28D4AC40931E64
78C8E76A24A825449501D867D2B1DCDEBAE99B9C752AE4ECD6DDE4A179C1C1E460938F9149EF655E515C03919A
289CB3DCA278FB7BF177F4FAA829DD8CE3F2AC9A7ECDE490971FAFD7DCE15EED9B71C018C64FA514514B24E8E4
F8C5C9B75C1E82579DC1233DFEC08238F6ADD62D391ACC1C5256A79E706D52D431C7A0145140B9FD149EB3A60D
C5E88CBBC2DA092411E9DC71F39A7766B447B344E847DCAC9DCB5ABBA8D145061D43A6FCF1E65CF15D0E90231D
3DD9CFE62995C6DCC5CA12A2C904A15F71DD27D451453E09D1A21450961CBB3EA8A956433B781F1CE33DFED54F
0E2B50A2B71D84ED6DB18028A28175F74FC6BDA105C529A791C25C4F3C7A11F71586268F4A66B726E33DE9EA6F
1B52B181C760724E47B514520A5A28A283FFD9'
            )
          ],
          "org.iso.18013.5.1.Utopia": [
            24(

h'A4686469676573744944418346672616E646F6D582055039196F846FDA152830317DBFDC9A5AAAC9238B648A2
1BA56AC7ACE71809BA71656C656D656E744964656E7469666965726855746F70696149446C656C656D656E7456
616C7565F5'
            )
          ]
        },
        "issuerAuth": [
          h'A10126',
          {
            33:
h'308201D230820178A00302010202104C7FF615C64916807A58CBE67BA1D0CE300A06082A8648CE3D04030230
34310B300906035504061302553310F300D060355040A0C0655746F706961311430120603550403 0C0B55746F
706961204941434 1301E170D313931303031303030303030305A170D323231303031303030303030305A3032310B3
009060355040613025553310F300D060355040A0C0655746F706961311230100603550403 0C0955746F706961
204453305930 1306072A8648CE3D020106082A8648CE3D03010703420004F2736B046885DF28F1D7E5115E448
9DC85CCA129BAF16D5DDA4E8F5DACC122E7D860FEBBCE074F18F8B3B2DDEF2D4AC201C8A60E48FC89DF413F52F
4761DE0BFA36E306C301D0603551D0E041604144FC593C52459ADFAF156949B969C3208C841A02D301F0603551
D23041830168014C757E4317E6908CABF0F0FD7566E438B2C027DE130090603551D1304023000300B0603551D
0F0404030205A030120603551D25040B3009060728818C5D050102300A06082A8648CE3D040302034800304502
21008C13A970FCE5E50DE2D364158167EF17BF350D6D9D574968BC053953EE895BE4022005C506A1426BD3332C
E6B69FDC571F7B3724E4AC5501FCCA61036BE12AD1502E'
          },

h'A56F6469676573744 16C676F726974686D675348412D3235366C76616C7565446967657374 73A16A6E616D65
537061636573A2716F72672E69736F2E31383031332E352E31AD015820AF307AD59C28C5032C1A49394E0BCE7
CB673A533D9AD12BD4CA980F6F12DB5BE02582076AAC7B52F938A4AEAE31D796BA5AEB2DBE0C9D96AD75B18D7
5AFAE875A79C76035820E2503B905A1754EF483BF5DD429C3A9BC66BDF7B3D1C080CFDEB4EC27AADEE6304582
0A963382AA8BA4C52DA522BC31D41E2564915F57CB144CB7C69F47A85B44A7E5905582020EE714191901E777A
E7F45B2EB6554ACA8DBB10F3093BE267DCE2C7F204D8E1065820B53392E547BA0A2369CBA5AC6D91F430FA8FB
76F0C4DF90FCB5791B6FF038EB60758204E0AD333B3BFD242DF9552E295A4D85EFD82E349456EA10FA80EEDD6D
C3195C0085820467AC91E0ACDEAC85D8272401F8749BC2E1B3F46385E5B91DBB71D8B9A8261EE0A5820BFAD1F0
BA42F3FE57068A2AFD4F2051EA60BB91CB0813FE98B323D969E218B450B5820541E4DE68CDD216DA9AEC22C912
2E1D3C25D38CFA26455DFF5882CC2C371F51413582041314B909ED36016CF9042E7B94AAC41955D459AB258A18
D4D87AD5817D6C68B175820D536408A50EC2818E426727078BD4429A2D9A0E061FECD46F5CFDAA77ADA95C0181
85820580BAE0ECF9A1C4D162586D7520615E2A7AC14584967FC9BBE8373E59302CA9878186F72672E69736F2E3
1383031332E352E312E55746F706961A1183458208D5FFF80E966D52C123C34B07BD6546909AD7AE6BF48D736E
0865A8B99853B7F696465766963654B6579A401022001215820855 3FBB8C982DB3F5A45D6FB12DFA04C0CF7A48
F43657F203B3DD05BA8D62392225820334E2EDC3DDC0549F8C8DC79C10FA3BFE26B389F6AAD3E0603A6FCAE1BB
FEA4567646F6354797065756F72672E69736F2E31383031332E352E312E6D444C6C76616C6964697479496E666
FA3667369676E6564C074323031392D30382D30385430303A30303A30305A6976616C696446726F6DC07432303
1392D30382D32345430303A30303A30305A6A76616C6964556E74696CC074323032392D30322D32385430303A3
0303A30305A',

h'D115FF9FBF1CF89E5B8DF4A0E98EA8337C6C8C5BC81E2EE98C43526DC47E7D9696BA7F45F011B2FBAA8FA33B
```

```
003F22A2A5B11FF896006878E6A17B4E9B918772'
          ]
        },
        "deviceSigned": {
          "nameSpaces": 24(
            h'A0'
          ),
          "deviceAuth": {
            "deviceMac": [
              h'A10105',
              {

              },
              null,

h'8d244815695c2467a59c2c2bbabf611c489a46fd7e758b3e3a6a00e33a5ed470'
            ]
          }
        },
        "errors": {
          "org.iso.18013.5.1": [
            {
              "document_number": 3
            }
          ]
        }
      }
    }
  ],
  "status": 0
}
```

The same example data encoded in CBOR bytes:

```
a36776657273696f6e63312e3069646f63756d656e747381a1756f72672e69736f2e31383031332e352e312e6d
444ca36c697373756572653696676e6564a26a6e616d65537061636573a2716f72672e69736f2e31383031332e3
52e3184d8185862a468646967657374494402667261 6e646f6d58207ef5d6fc314c8e6b91343741b2a5ca6835
bf52728ed64bf773d020075c5556c771656c656d656e744964656e7469666965726a676976656e5f6e616d656
c656c656d656e7456616c756563546f6dd818585ba4686469676573744944036672616e646f6d5004fa3f77bc
7d3f65ad8e71e8bca98d7b71656c656d656e744964656e7469666965726a62697274685f646174656c656c656
d656e7456616c7565d9465d6a313939392d30332d3134d8185863a4686469676573744944046672616e646f6d5
0635dce207f4ba4fff087ab1d427b663071656c656d656e744964656e7469666965726a69737375655f6461746
56c656c656d656e7456616c7565c074323031372d30372d31345430303a30303a30305ad818590471a46864696
765737449441366726 16e646f6d5820e8fc51b0ae7a73d23d91b5333512baeac318b3edf5975a8888901261 81f
ac4f771656c656d656e744964656e74696696572687 06f7274726169746c656c656d656e7456616c756559041
2ffd8ffe000104a4649460001010100900090000 0ffdb004300130d0e110e0c13110f11151413171d301f1d1a1
a1d3a2a2c2330453d4947443d43414c566d5d4c51685241435f82606871757b7c7b4a5c869085778f6d787b76f
fdb0043011415151d191d381f1f38764f434f7676767676767676767676767676767676767676767676767676
7676767676767676767676767676767676767676767676ffc0001108001800640301220002110103110 1ffc40
01b0000030100030100000000000000000000000005060401020307ffc4003210000103030302050203090000000
0000001020304000511061221133114155161712241078 1a1163542527391b2c1f1ffc4001501010100000000000
0000000000000000000001ffc4001a1101010100030100000000000000000000000014111213161ffda000c030
100021103110 03f00a5bbde22da2329c7d692bc7d0d03f52cfb0ff75e7a7ef3e7709723a1d0dae146ddfbb3c03
9ce07ad2bd47a7e32dbb8dd1d52d6ef4b284f64a480067dfb51f87ffb95ff00eb9ff14d215de66af089ce44b7d
bde9cb6890a2838eddf18078f7add62d411ef4db9b10a65d6b95a147381ea0d495b933275fe6bba75c114104a
8ba410413e983dff004f5af5d34b4b4cde632d0bf1fd1592bdd91c6411f3934c2fa6af6b54975d106dcf4a65a
e56e856001ebc03c7ce29dd9eef1ef10fc447dc9da76ad2aee93537a1ba7e4f70dd8eff0057c6dffb5e1a1985
4a83758e54528750946ec6704850cd037bceb08b6d7d2cc76d3317fc7b5cc04fb6707269c5c6e0c5b60ae5492
42123b0e493f602a075559e359970d98db89525456b51c951c8afa13ea8e98e3c596836783d5c63f5a61a99fd
b7290875db4be88ab384bbbbbfc7183fdeaa633e8951db7da396dc48524fb1a8bd611a5aa2a2432f30ab420a7
a6d3240c718cf031fa9ef4c9ad550205aa02951df4a1d6c8421b015b769db8c9229837ea2be8b1b0d39d0eba9
c51484efdb8c0efd8d258daf3c449699f2edbd4584e7af9c64e3f96b9beb28d4ac40931e6478c8e76a24a8254
49501d867d2b1dcdebae99b9c752ae4ecd6dde4a179c1c1e460938f9149ef655e515c03919a289cb3dca278fb
7bf177f4faa829dd8ce3f2ac9a7ecde490971fafd7dce15eed9b71c018c64fa514514b24e8e4f8c5c9b75c1e8
2579dc1233dfec08238f6add62d391acc1c5256a79e706d52d431c7a0145140b9fd149eb3a60dc5e88cbbc2da
092411e9dc71f39a7766b447b344e847dcac9dcb5abba8d145061d43a6fcf1e65cf15d0e90231d3dd9cfe6299
5c6dcc5ca12a2c904a15f71dd27d451453e09d1a21450961cbb3ea8a956433b781f1ce33dfed54f0e2b50a2b7
1d84ed6db18028a28175f74fc6bda105c529a791c25c4f3c7a11f71586268f4a66b726e33de9ea6f1b52b181c
760724e47b514520a5a28a283ffd978186f72672e69736f2e31383031332e352e312e55746f70696181d81858
5ea4686469676573744944183466726 16e646f6d582055039196f846fda152830317dbfdc9a5aaac9238b648a2
1ba56ac7ace71809ba71656c656d656e744964656e7469666965726855746f70696149446c656c656d656e7456
```

```
616c7565f56a69737375657241757468844 3a10126a118215901d6308201d230820178a00302010202104c7ff6
15c64916807a58cbe67ba1d0ce300a06082a8648ce3d0403023034310b3009060355040613025553310f300d06
0355040a0c0655746f706961311430120603550403 0c0b55746f706961204941434 1301e170d31393130303130
30303030305a170d323231303031303030303030305a3032310b3009060355040613025553310f300d060355040a
0c0655746f706961311230100603550403 0c0955746f706961204453305930 1306072a8648ce3d020106082a86
48ce3d03010703420004f2736b046885df28f1d7e5115e4489dc85cca129baf16d5dda4e8f5dacc122e7d860fe
bbce074f18f8b3b2ddef2d4ac201c8a60e48fc89df413f52f4761de0bfa36e306c301d0603551d0e041604144f
c593c52459adfaf156949b969c3208c841a02d301f0603551d23041830168014c757e4317e6908cabf0f0fd756
6e438b2c027de130090603551d1304023000300b0603551d0f0404030205a030120603551d25040b3009060728
818c5d050102300a06082a8648ce3d040302034800304502210 08c13a970fce5e50de2d364158167ef17bf350d
6d9d574968bc053953ee895be4022005c506a1426bd3332ce6b69fdc571f7b3724e4ac5501fcca61036be12ad1
502e59032ca56f646967657374416c676f726974686d675348412d3235366c76616c756544696765737473a16a
6e616d65537061636573a2716f72672e69736f2e31383031332e352e31ad015820af307ad59c28c5032c1a4939
4e0bce7cb673a533d9ad12bd4ca980f6f12db5be02582076aac7b52f938a4aeae31d796ba5aeb2dbe0c9d96ad7
5b18d75afae875a79c76035820e2503b905a1754ef483bf5dd429c3a9bc66bdf7b3d1c080cfdeb4ec27aadee63
045820a963382aa8ba4c52da522bc31d41e2564915f57cb144cb7c69f47a85b44a7e5905582020ee714191901e
777ae7f45b2eb6554aca8dbb10f3093be267dce2c7f204d8e1065820b53392e547ba0a2369cba5ac6d91f430fa
8fb76f0c4df90fcb5791b6ff038eb60758204e0ad333b3bfd242df9552e295a4d85efd82e349456ea10fa80eed
d6dc3195c0085820467ac91e0acdeac85d8272401f8749bc2e1b3f46385e5b91dbb71d8b9a8261ee0a5820bfad
1f0ba42f3fe57068a2afd4f2051ea60bb91cb0813fe98b323d969e218b450b5820541e4de68cdd216da9aec22c
9122e1d3c25d38cfa26455dff5882cc2c371f51413582041314b909ed36016cf9042e7b94aac41955d459ab258
a18d4d87ad5817d6c68b175820d536408a50ec2818e426727078bd4429a2d9a0e061fecd46f5cfdaa77ada95c0
18185820580bae0ecf9a1c4d162586d7520615e2a7ac14584967fc9bbe8373e59302ca9878186f72672e69736f
2e31383031332e352e312e55746f706961a1183458208d5fff80e966d52c123c34b07bd6546909ad7ae6bf48d7
36e0865a8b99853b7f696465766963654b6579a401022001215820 8553fbb8c982db3f5a45d6fb12dfa04c0cf7
a48f43657f203b3dd05ba8d62392225820334e2edc3ddc0549f8c8dc79c10fa3bfe26b389f6aad3e0603a6fcae
1bbfea4567646f6354797065756f72672e69736f2e31383031332e352e312e6d444c6c76616c6964697479496e
666fa3667369676e6564c074323031392d30382d30385430303a30303a30305a6976616c696446726f6dc07432
3031392d30382d32345430303a30303a30305a6a76616c6964556e74696cc074323032392d30322d3238543030
3a30303a30305a5840d115ff9fbf1cf89e5b8df4a0e98ea8337c6c8c5bc81e2ee98c43526dc47e7d9696ba7f45
f011b2fbaa8fa33b003f22a2a5b11ff896006878e6a17b4e9b9187726c6465766963655369676e6564a26a6e61
6d65537061636573d81841a06a6465766963654 1757468a1696465766963654d61638443a10105a0f658208d24
4815695c2467a59c2c2bbabf611c489a46fd7e758b3e3a6a00e33a5ed470666572726f7273a1716f72672e6973
6f2e31383031332e352e3181a16f646f63756d656e745f6e756d6265720366737461746174757300
```


D.4.2 Online retrieval

D.4.2.1 WebAPI

D.4.2.1.1 Request

An example of an OnlineRequest is as follows:

```
{
  "version": "1.0",
  "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJUb2tlbklEIjoiM0tQMkhSMzJWTkQiLCJUb2tlbiI6IjEyMzQ1
NiIsIkRldmljZUlEIjoieTlrdzd6d3h5YjhjZDY2aDl2azciLCJEb2N1bWVudElEIjoiNjdwZnZwdDZmaGZ5Ymt3cG
1uOGMifQ.OqlNGh_W7f7h9o0nqE2IneOyXc3zOH6RzzUvs2dnJGw",
  "docType": "org.iso.18013.5.1.mDL",
  "nameSpaces": {
       "org.iso.18013.5.1": {
              "family_name": false,
              "given_name": false,
              "birth_date": false,
              "issue_date": false,
              "expiry_date": false,
              "issuing_country": false,
              "issuing_authority": false,
              "document_number": false,
              "driving_privileges": false,
              "un_distinguishing_sign": false,
              "portrait": false
       }
   }
}
```

D.4.2.1.2 Response

An example of an OnlineResponse is as follows:

JWT Payload:


```
{
  "version": "1.0",
  "docType": "iso.org.18013.5.1.mDL",
  "nameSpaces": {
    "org.iso.18013.5.1": {
      "family_name": "TURNER",
      "given_name": "SUSAN",
      "birth_date": "1998-08-28",
      "issue_date": "2018-01-15T10:00:00.0000000-07:00",
      "expiry_date": "2022-08-27T12:00:00.0000000-06:00",
      "issuing_country": "US",
      "issuing_authority": "CO",
      "document_number": "542426814",
      "driving_privileges": [
        {
          "codes": [
            {
              "code": "D"
            }
          ],
          "vehicle_category_code": "D",
          "issue_date": "2019-01-01",
          "expiry_date": "2027-01-01"
        },
        {
          "codes": [
            {
              "code": "C"
            }
          ],
          "vehicle_category_code": "C",
          "issue_date": "2019-01-01",
          "expiry_date": "2017-01-01"
        }
      ],
      "un_distinguishing_sign": "USA",
      "portrait":
"/9j/4AAQSkZJRgABAQEAkACQAAD/2wBDABMNDhEODBMRDxEVFBMXHTAfHRoaHToqLCMwRT1JR0Q9Q0FMVm1dTFF
oUkFDX4JgaHF1e3x7SlyGkIV3j214e3b/2wBDARQVFR0ZHTgfHzh2T0NPdnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dn
Z2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnZ2dnb/wAARCAAYAGQDASIAAhEBAxEB/8QAGwAAAwEAAwEAAAAAAAAAAAA
AAAAUGBAECAwf/xAAyEAABAwMDAgUCAwkAAAAAAAABAgMEAAURBhIhEzEUFVFhcSJBB4GhFjVCUnORssHx/8QAF
QEBAQAAAAAAAAAAAAAAAAAAAAH/xAAaEQEBAQADAQAAAAAAAAAAAAAAAUERITFh/9oADAMBAAIRAxEAPwClu94i
2iMpx9aSvH0NA/Us+w/3Xnp+8+dwlyOh0NrhRt37s8A5zgetK9R6fjLbuN0dUtbvSyhPZKSABn37Ufh/+5X/AOu
f8U0hXeZq8InORLfb3py2iQooO3fGAePet1i1BHvTbmxCmXWuVoUc4HqDUlbkzJ1/mu6dcEUEEqLpBBBPpg9/
wBPWvXTS0tM3mMtC/H9FZK92RxkEfOTTC+mr2tUl10Qbc9KZa5W6FYAHrwDx84p3Z7vHvEPxEfcnadq0q7pNTeh
un5PcN2O/wBXxt/7XhoZhUqDdY5UUodQlG7GcEhQzQN7zrCLbX0sx20/x7XMBPtnByacXG4MW20zF/x7XMBPtnByacXG4MW2CuVJJCEjsOS
T9gKgdVWeNZlw2Y24l/WmGpn9tykIdtL6Iqzh/LuLu7v8cYP96qYz6JUdt9tZl2nbAxACDpoAGQBQRCHgVIVNktxpqyVjHXvPmeCThXJpXMFkqmRQEIbAVt2nbjJImrHVWMq9tMtLvSk8zIcqbSstKsaJiJVZKUhVjtAGiiv2QqCTxLsRHJlthn9ckDJnRI8WlrmCs8KtqAQptnbQKSCvjmHelO5/XWvCrpdGk+q1SVStsQyTgNk9Qx1dA8aqsxmJTPWLoZhPcrvQwX/rSCQqxA/jPjgPtGrbMuMjFcsEvrkA/xRyNiJaXnxeTx/v7Uo4QSNYy+O8T+5ZxKvrqvUfDzgu/0U7adDTHNRSa7rU8/7qsyv5TRnAGFaSkvRvKodRp/jULUrQ8HVfbH
/2Q=="
    }
  },
  "iat": 1571085297,
  "exp": 1571085417
}
```


D.4.2.2 OIDC

An example of an OIDC workflow is as follows:

Step 1 - Get OpenID configuration (/.well-known/openid-configuration)

Request:

```
GET /.well-known/openid-configuration HTTP/1.1
Host: {BASE_URL}
Accept: */*
```

Response:

```
{
    "issuer": "https://{BASE_URL}",
    "jwks_uri": "https://{BASE_URL}/.well-known/openid-configuration/jwks", //Step 5 -
Validate Certificate Chain
    "authorization_endpoint": "https://{BASE_URL}/connect/authorize", //Step 3 -
Authorization based on OAuth 2.0 Authorization Code Flow Grant
    "token_endpoint": "https://{BASE_URL}/connect/token", //Step 4 - Get Id Token
    "userinfo_endpoint": "https://{BASE_URL}/connect/userinfo",
    "end_session_endpoint": "https://{BASE_URL}/connect/endsession",
    "revocation_endpoint": "https://{BASE_URL}/connect/revocation",
    "introspection_endpoint": "https://{BASE_URL}/connect/introspect",
    "device_authorization_endpoint": "https://{BASE_URL}/connect/deviceauthorization",
    "registration_endpoint": "https://{BASE_URL}/connect/register", //Step 2 - Client
Registration
    "frontchannel_logout_supported": true,
    "frontchannel_logout_session_supported": true,
    "backchannel_logout_supported": true,
    "backchannel_logout_session_supported": true,
    "scopes_supported": [
        "org.iso.18013.5.1:resident_address",
        "org.iso.18013.5.1:portrait",
        "org.iso.18013.5.1:portrait_capture_date",
        "org.iso.18013.5.1:age_in_years",
        "org.iso.18013.5.1:age_birth_year",
        "org.iso.18013.5.1:age_over_18",
        "org.iso.18013.5.1:birthplace",
        "org.iso.18013.5.1:age_over_21",
        "org.iso.18013.5.1:nationality",
        "org.iso.18013.5.1:resident_city",
        "org.iso.18013.5.1:resident_state",
        "org.iso.18013.5.1:resident_postal_code",
        "org.iso.18013.5.1:biometric_template_face",
        "org.iso.18013.5.1:biometric_template_signature_sign",
        "org.iso.18013.5.1:issuing_jurisdiction",
        "org.iso.18013.5.1:name_nat_char",
        "org.iso.18013.5.1:hair_color",
        "org.iso.18013.5.1:weight",
        "org.iso.18013.5.1:given_name",
        "org.iso.18013.5.1:family_name",
        "org.iso.18013.5.1:birthdate",
        "org.iso.18013.5.1:issue_date",
        "org.iso.18013.5.1:eye_color",
        "org.iso.18013.5.1:expiry_date",
        "org.iso.18013.5.1:issuing_authority",
        "org.iso.18013.5.1:document_number",
        "org.iso.18013.5.1:administrative_number",
        "org.iso.18013.5.1:driving_privileges",
        "org.iso.18013.5.1:gender",
        "org.iso.18013.5.1:height",
        "org.iso.18013.5.1:issuing_country",
        "openid",
    ],
    "claims_supported": [
        "org.iso.18013.5.1:resident_address",
        "org.iso.18013.5.1:portrait",
        "org.iso.18013.5.1:portrait_capture_date",
        "org.iso.18013.5.1:age_in_years",
        "org.iso.18013.5.1:age_birth_year",
        "org.iso.18013.5.1:age_over_18",
        "org.iso.18013.5.1:birthplace",
        "org.iso.18013.5.1:age_over_21",
```

```
        "org.iso.18013.5.1:nationality",
        "org.iso.18013.5.1:resident_city",
        "org.iso.18013.5.1:resident_state",
        "org.iso.18013.5.1:resident_postal_code",
        "org.iso.18013.5.1:biometric_template_face",
        "org.iso.18013.5.1:biometric_template_signature_sign",
        "org.iso.18013.5.1:issuing_jurisdiction",
        "org.iso.18013.5.1:hair_color",
        "org.iso.18013.5.1:weight",
        "org.iso.18013.5.1:eye_color",
        "org.iso.18013.5.1:height",
        "org.iso.18013.5.1:gender",
        "org.iso.18013.5.1:driving_privileges",
        "org.iso.18013.5.1:administrative_number",
        "org.iso.18013.5.1:document_number",
        "org.iso.18013.5.1:issuing_authority",
        "org.iso.18013.5.1:issuing_country",
        "org.iso.18013.5.1:expiry_date",
        "org.iso.18013.5.1:issue_date",
        "org.iso.18013.5.1:birthdate",
        "org.iso.18013.5.1:family_name",
        "org.iso.18013.5.1:given_name",
        "docType",
        "org.iso.18013.5.1:name_nat_char",
        "sub"
    ],
    "grant_types_supported": [
        "authorization_code",
        "client_credentials",
        "refresh_token",
        "implicit",
        "urn:ietf:params:oauth:grant-type:device_code"
    ],
    "response_types_supported": [
        "code",
        "token",
        "id_token",
        "id_token token",
        "code id_token",
        "code token",
        "code id_token token"
    ],
    "response_modes_supported": [
        "form_post",
        "query",
        "fragment"
    ],
    "token_endpoint_auth_methods_supported": [
        "client_secret_basic",
        "client_secret_post"
    ],
    "subject_types_supported": [
        "public"
    ],
    "id_token_signing_alg_values_supported": [
        "ES256"
    ],
    "code_challenge_methods_supported": [
        "plain",
        "S256"
    ]
}
```

Step 2 – Client Registration

The mDL Reader can pre-register with the issuing authority through their client registration process to obtain a client_id OR the mDL Reader can use dynamic client registration as specified in OpenID Connect Dynamic Registration, *N. Sakimura et. al., Defines how clients/readers dynamically register with OpenID Providers, November 2014*.

The mDL Reader in order to utilize dynamic client registration will find the “registration_endpoint” from the response from /.well-known/openid-configuration from the previous step. If the “registration_endpoint” does not exist then mDL Reader have to pre-register with the issuing authority through their client registration process to obtain a client_id in an offline manner.

In case the “registration_endpoint” value does exist and the mDL Reader had not already pre-register, mDL Reader should proceed with the following request:

Request:

```
POST /connect/register HTTP/1.1
Host: {BASE_URL}
Content-Type: application/json

{
    "redirect_uris": [
        "YOUR REDIRECT URI",
        "com.company.isomdlreader://login"
    ],
    "scope": "openid org.iso.18013.5.1:given_name org.iso.18013.5.1:family_name org.
iso.18013.5.1:birthdate org.iso.18013.5.1:portrait"
}
```

Response

```
{
    "client_id": "ac55a761d4e74b35a9f2de06c3ea4f63",
    "client_id_issued_at": 1569657361,
    "client_secret": "OoZlNbaKSPQKWSLSc6AHTUFmpPOssio1",
    "client_secret_expires_at": 0,
    "grant_types": [
        "authorization_code"
    ],
    "client_name": "ac55a761d4e74b35a9f2de06c3ea4f63",
    "client_uri": null,
    "logo_uri": null,
    "redirect_uris": [
        "com.company.isomdlreader://login"
    ],
    "Scope": "openid org.iso.18013.5.1:given_name org.iso.18013.5.1:family_name org.
iso.18013.5.1:birthdate"
}
```

Step 3 – Authorization based on OAuth 2.0 Authorization Code Flow Grant

The mDL Reader shall use the Authorization Code Flow Grant using the client_id from the previous step and the full token as input to the “login_hint” parameter as specified in OpenID Connect Core 1.0.

In this example the request asks for the following scopes:

- org.iso.18013.5.1:given_name
- org.iso.18013.5.1:family_name
- org.iso.18013.5.1:birthdate
- org.iso.18013.5.1:portrait

Apart from that, OpenID Connect requests contain the openid scope value.

The next step is the redirection back to mDL Holder native app based on “OAuth 2.0 for Native Apps” RFC 8252

Request:

```
GET /connect/authorize?client_id=ac55a761d4e74b35a9f2de06c3ea4f63&amp; scope=openid org.
iso.18013.5.1:given_name org.iso.18013.5.1:family_name org.iso.18013.5.1:birthdate org.
iso.18013.5.1:portrait&amp; redirect_uri=com.company.isomdlreader://login&amp; response_
```

```
type=code&amp; login_hint=TOKEN-FROM-MDL-HOLDER HTTP/1.1
Host: {BASE_URL}
```

Response:

```
{
    "Query": {
        "code": [
            "1ad358e1fbc55e7cce0e515966f7b72699b9d47ce7fd6a1a63423820c9ed8f42"
        ],
        "scope": [
            "openid       org.iso.18013.5.1:given_name       org.iso.18013.5.1:family_name
org.iso.18013.5.1:birthdate org.iso.18013.5.1:portrait"
        ]
    },
    "Headers": {
        "Cache-Control": [
            "no-cache"
        ],
        "Connection": [
            "Keep-Alive"
        ],
        "Accept": [
            "*/*"
        ],
        "Accept-Encoding": [
            "gzip, deflate"
        ],
        "Cookie": [
            "idsrv=……..."
        ],
        "Host": [
            "{BASE_URL}"
        ],
        "Referer": [
            "https://{BASE_URL}/connect/authorize/callback?client_id= ac55a761d4e74b35a
9f2de06c3ea4f63&scope=openid%20org.iso.18013.5.1:given_name%20org.iso.18013.5.1:family_
name%20org.iso.18013.5.1:birthdate%20org.iso.18013.5.1:portrait&redirect_uri=com.company.
isomdlreader://login&response_type=code&login_hint=VIENNAS-TOKEN"
        ],
        "X-Original-Proto": [
            "http"
        ],
        "X-Original-For": [
            "127.0.0.1:58873"
        ]
    }
}
```

Step 4 – Get Id Token

Get the Id Token using the code value from the previous step.

Request:

```
POST /connect/token HTTP/1.1
Content-Type:application/x-www-form-urlencoded
Host: {BASE_URL}

redirect_uri: com.company.isomdlreader://login
grant_type:authorization_code
code:385b19cf37ddb9bd35a29e6db85d2d52a178c96009fd7a14d43ccc4b324f6322
client_id: ac55a761d4e74b35a9f2de06c3ea4f63
client_secret: OoZlNbaKSPQKWSLSc6AHTUFmpPOssio1
```

Response:

```
{
    "id_token":
"eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYmYiOjE1NzEzMjMzMDYsImV4cCI6MTU3MTMyMzYwNiwiaXNz
```

```
IjoiaHR0cHM6Ly9sb2NhbGhvc3Q6NTAwMSIsImF1ZCI6IjdiMGQyZGU0OTVmOTQ5ZWY5YmIxMzRlMDA1NWMxYjZjI
iwiaWF0IjoxNTcxMzIzMDM1LCJhdF9oYXNoIjoiSWluQTE2X3oxUVFXd2k1aE1LbFdCZyIsInN1YiI6IjE0Njg3OD
UyMiIsImF1dGhfdGltZSI6MTU3MTMyMjkzNiwiaWRwIjoibG9jYWwiLCJkb2NUeXBlIjoib3JnLmlzby4xODAxMy4
1LjEubURMIiwib3JnLmlzby4xODAxMy41LjE6cG9ydHJhaXQiOiJfOWpfNEFBUVNrWkpSZ0FCQVFFQThBRHdBQURf
NFFDS1JYaHBaZ0FBVFUwQUtnQUFBQWdBQndFYUFBVUFBQUFCQUFBQVlnRWJBQVVBQUFBQkFBQUFhZ0VvQUFNQUFBQ
UJBQUlBQUFFeEFBSUFBQUFRQUFBQWNsRVFBQUVBQUFBQkFRQUFBRkVSQUFRQUFBQUJBQUFBQUZFU0FBUUFBQUFCQUF
BQUFBQUFBQUFBQUFEd0FBQUFBUUFBQVBBQUFBQUJjR0ZwYm5RdWJtVjBJRFF1TVM0MkFQX2JBRU1BS2gwZ0pTQWFLa
VVpSlM4dEtqSV9hVVFfT2pvX2dWeGhUR21aaHFDZWxvYVRrYWk5OHMyb3MtVzFrWlBTXzlYbC12X19fXy1qeV9fX19
fX184dl9fX19fYkFFTUJMUzh2UHpjX2ZFUkVmUC11azY3X19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX
19fX19fX19fX19fX19fX19fX19fX19fX19fX19BQUJFSUFDZ0FLQU1CSWdBQ0VRRURFUUhfeEFBZkFBQUJ
CUUVCQVFFQkFRQUFBQUFBQUFBQUFRSURCQVVHQndnSkNndl94QUMxRUFBQ0FRTURBZ1FEQlFVRUJBQUFBWDBCQWdNQ
UJCRUZFaUV4UVFZVFVXRUhJbkVVTW9HUm9RZ2pRckhCRlZMUjhDUXpZbktDQ1FvV0Z4Z1pHaVVtSnlncEtqUTFOamM
0T1RwRFJFVkdSMGhKU2xOVVZWWlhXRmxhWTJSbFptZG9hV3B6ZEhWMmQzaDVlb09FaFlhSGlJbUtrcE9VbFphWG1Kb
WFvcU9rcGFhbnFLbXFzck8wdGJhM3VMbTZ3c1BFeGNiSHlNbkswdFBVMWRiWDJObmE0ZUxqNU9YbTUtanA2dkh5OF9
UMTl2ZjQtZnJfeEFBZkFRQURBUUVCQVFFQkFRRUJBQUFBQUFBQUFRSURCQVVHQndnSkNndl94QUMxRVFBQ0FRSUVCQ
U1FQndVRUJBQUJBbmNBQVFJREVRUUZJVEVHRWtGUkIyRnhFeUl5Z1FnVVFwR2hzY0VKSXpOUzhCVmljdEVLRmlRMDR
TWHhGeGdaR2lZbktDa3FOVFkzT0RrNlEwUkZSa2RJU1VwVFZGVldWMWhaV21Oa1pXWm5hR2xxYzNSMWRuZDRlWHFD
ZzRTRmhvZUlpWXFTazVTVmxwZVltWnFpbzZTbHBxZW9xYXF5czdTMXRyZTR1YnJDdzhURnhzZkl5Y3JTMDlUVjF0Z
lkyZHJpNC1UbDV1Zm82ZXJ5OF9UMTl2ZjQtZnJfMmdBTUF3RUFBaEVERVFBX0FOU2lpb3BUeUJRQV9ldnJUcXJWSk
VlU0tBSktLUTBVQU9wa2k1NUZQcUZuSlBIQW9BWlVpTGprMHplM3JRSElQUElvQWxORk56UlFBOV91R29LS0tBRXB
wb29vQWxYN29vb29vQV9fMlEiLCJvcmcuaXNvLjE4MDEzLjUuMTpnaXZlbl9uYW1lIjoiSk9ITiIsIm9yZy5pc28u
MTgwMTMuNS4xOmZhbWlseV9uYW1lIjoiRE9FIiwib3JnLmlzby4xODAxMy41LjE6YmlydGhkYXRlIjoiMTk4NS0wM
y0zMCIsImFtciI6WyJwd2QiXX0.LpgOQfGycgEXRGFV-vLUEup9oxKfTzDyU179mtTFTodXBwx0aWaZAvslWH2rt-
ET1RQG2Pa3ccNE4anfqPHT9Q",
    "access_token":
"eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYmYiOjE1NzEzMjMwMzUsImV4cCI6MTU3MTMyMzMzNSwiaX
NzIjoiaHR0cHM6Ly9sb2NhbGhvc3Q6NTAwMSIsImF1ZCI6Imh0dHBzOi8vbG9jYWxob3N0OjUwMDEvcmVzb3VyY
2VzIiwiY2xpZW50X2lkIjoiN2IwZDJkZTQ5NWY5NDllZjliYjEzNGUwMDU1YzFiNmMiLCJzdWIiOiIxNDY4Nzg1M
jIiLCJhdXRoX3RpbWUiOjE1NzEzMjI5MzYsImlkcCI6ImxvY2FsIiwic2NvcGUiOlsib3JnLmlzby4xODAxMy41L
jE6cG9ydHJhaXQiLCJvcmcuaXNvLjE4MDEzLjUuMTpnaXZlbl9uYW1lIiwib3JnLmlzby4xODAxMy41LjE6ZmFt
aWx5X25hbWUiLCJvcmcuaXNvLjE4MDEzLjUuMTpiaXJ0aGRhdGUiLCJvcGVuaWQiXSwiYW1yIjpbInB3ZCJdfQ.
NjjUPRIgZ2opFeEJdJTV2_Hy2JSH_15ntoQho54SKSn1ZAtqXw6q-Yiulmo3L8ZAUa6_ARAA8VdNr219upULmA",
    "expires_in": 300,
    "token_type": "Bearer"
}
```

Step 5 – Validate Certificate Chain

Request:

```
GET /well-known/openid-configuration/jwks HTTP/1.1
Host: {BASE_URL}
```

Response:

```
{
    "keys": [
        {
            "kty": "EC",
            "use": "sig",
            "x5c": [

"MIIB7TCCAZKgAwIBAgIId9jYlj3/sxowCgYIKoZIzj0EAwIwKzELMAkGA1UEBhMCR1IxDTALBgNVBAoTBElBQ0ExD
TALBgNVBAMTBElBQ0EwHhcNMTkxMDE2MTIyNjAwWhcNMjAxMDE1MTIyNjAwWjA7MQswCQYDVQQGEwJHUjENMAsGA1U
EChMESUFDQTEdMBsGA1UEAxMUSUFDQSBKV1MgQ2VydGlmaWNhdGUwWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAASQ
hNR/MjfoON0y1Mz1p9Kiik6detBenFETTwS60C/789y21J0KxEeD6FqFApFII0wsDL5TQ4DECt9tW3oRC53ko4GPMI
GMMB0GA1UdDgQWBBSxUNshdoNVSQA0n4acab8FAZJwADAfBgNVHSMEGDAWgBSJLDEn8Ig7L8sbyskqsshX4gG2eTAL
BgNVHQ8EBAMCB4AwEgYDVR0lBAswCQYHKIGMXQUBAzApBgNVHR8EIjAgMB6gHKAahhhodHRwczovL2lhY2EuY29tL2
dldC5jcmwwCgYIKoZIzj0EAwIDSQAwRgIhAIWTdEcTRFpkqzwRGjmq0T+Hr8CT3f7ASRCwR6TJSmOYAiEAr9bn9jTb
mrGpHRZccKSezL7BSUNPfbjbUtNAs1d2v9I=",

"MIIBvzCCAWSgAwIBAgIQAIyBH6jN6S2j1FBLxxMtdzAKBggqhkjOPQQDAjArMQswCQYDVQQGEwJHUjENMAsGA1UEC
hMESUFDQTENMAsGA1UEAxMESUFDQTAeFw0xOTEwMTYxMjE4MDBaFw0yOTEwMTYxMjE4MDBaMCsxCzAJBgNVBAYTAk
dSMQ0wCwYDVQQKEwRJQUNBMQ0wCwYDVQQDEwRJQUNBMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEVI4S4wJiErN
CyaXZujeh7HFPRB8MxI8pKxJnJk9MHlDIhsdjt2tjcaqz7nMyyi6WMHpVO27NHzQ8Pu1xipWj7KNqMGgwDwYDVR0T
AQH/BAUwAwEB/zAdBgNVHQ4EFgQUiSwxJ/CIOy/LG8rJKrLIV+IBtnkwCwYDVR0PBAQDAgEGMCkGA1UdHwQiMCAwHq
AcoBqGGGh0dHBzOi8vaWFjYS5jb20vZ2V0LmNybDAKBggqhkjOPQQDAgNJADBGAiEAhVLbpC5Bjt2PDtgzau5XwobE
wj7Cdb46k6X8344O2RECIQDDX4A67pHrz8HiR1nYU0gXy9u3bJVIRccZi2wkrn/DtQ=="
            ],
            "alg": "ES256"
        }
```

```
    ]
}
```

D.5 Security mechanisms

D.5.1 Session encryption

This example uses the example data for DeviceEngagement, OfflineRequest and OfflineResponse.

MDL private COSE key :

```
{
  1: 2,
  -1: 1,
  -2: h'E4706DE318A40D0BD8648B7907B0283F7445370241FF1FDD77F08D6A598BE902',
  -3: h'79767FBE391223F61DCD5E980133A035B4918F6F9DE41B3CEB8D801860DF8859',
  -4: h'90AE586A95B305C7354AEFA49BB267EE4EC74816B7494BDBD0B0E6E98F6718A5'
}
```

Reader private COSE key :

```
{
  1: 2,
  -1: 1,
  -2: h'18CD31CC15E50D45C3597E2B9ECAB5771A5F61FC4290819415006251EF180C8E',
  -3: h'FC159B5D687BDD4580124480C3A7474ECE63234405F6126DC65653C25D573CD2',
  -4: h'BDBA90434FEB6113986A7A4AB214432933423F842FD2F0F889398C3798B3DAFA'
}
```

ECKA-DH result (Zab):

```
0bb2778ca8a43e580f1c992c1b9c2ea47d9ef3e53b477ef62d4a0dfe8b58bded
```

HKDF with input null and salt 01 result (SKDevice):

```
e4b68931b7ec6defecd9f2c9f9f60268e3f770c71b7795350112fd3fba4fb6c2
```

HKDF with input null and salt 00 result (SKReader):

```
4c0e999158099212366fbe740f3c822f9090463182c24465b0e660a018fb1f62
```

Encrypted message with SKReader and IV '`000000000000000000000001`':

```
{
  "eReaderKey": 24(
h'A40102200121582018CD31CC15E50D45C3597E2B9ECAB5771A5F61FC4290819415006251EF180C8E225820FC
159B5D687BDD4580124480C3A7474ECE63234405F6126DC65653C25D573CD2'
  ),
  "data": h'
d306a1a31a1cd24c364aaea6b1d3e8dd239b23565f3dda697a9d9dc1f6c21c981db8a6c99706b627889c513b6a
e3f6e422e708c030929fc9cb34cab7543c1f54a2cf2986b4ffd25ea5913b0e9359e079d66e803d64fce22e5328
338186642c1637bfe0cc27d782843c6088355aa37049250d104701b115e1bef1ec285ed397a76d4c98fdbf8ec5
50138f9ea1989226ab1ce165e2dc6cc9c8a2dc129629678cc7924439d55ea06fa80695c01e58a87b580e22bc2c
3f71c568c781c019ce9cfcb1352e49a70b006d90d9965da295'
}
```

CBOR bytes:

```
a26a655265616465724b6579d818584ba40102200121582018cd31cc15e50d45c3597e2b9ecab5771a5f61fc4
290819415006251ef180c8e225820fc159b5d687bdd4580124480c3a7474ece63234405f6126dc65653c25d57
3cd2646461746158cdd306a1a31a1cd24c364aaea6b1d3e8dd239b23565f3dda697a9d9dc1f6c21c981db8a6
c99706b627889c513b6ae3f6e422e708c030929fc9cb34cab7543c1f54a2cf2986b4ffd25ea5913b0e9359e07
9d66e803d64fce22e5328338186642c1637bfe0cc27d782843c6088355aa37049250d104701b115e1bef1ec28
5ed397a76d4c98fdbf8ec550138f9ea1989226ab1ce165e2dc6cc9c8a2dc129629678cc7924439d55ea06fa806
95c01e58a87b580e22bc2c3f71c568c781c019ce9cfcb1352e49a70b006d90d9965da295
```

Encrypted message with SKReader and IV '`000000000000000100000001`':

```
{
  "data": h'
0bc798e4d869d17c321e24c3111d11fd2fc955dad221bb803e08ebf309bd7b94d57f3162d5370f09415145645d
785fdb0d3da84201227d6feb7f0bea059e954d6f74d94c223e589ae44edfff85b9bc51887d71eb07ac40ce57d
a3bb010aa23de8a6c46998a024e1a1f183add692b572bdd25f7d30f96427cbfeb117112f97037d002893c45d0
49eaa56413cf23bf54f519ea3ca5ae984917b11b21cd20a199998c93f55fd271ad442231fd0bc720394a756c6
9b6e86785ed09ea5866b48a2567edd91ab74d4234e6a12b4adf831146e0a73907a7555621032b8c5d548075b1
6a1b5ade26eaa97c77866460a8067b4fbf58ff887fb6d33572122c70d724636fae4fd57f17ad388326474cf69
d7f6183a6894552267ec7c9aa3b5e251ac8afb5900b02353e591edf245c403b9411514e60fdfeb6d5bcb907be
2eae078999994a1726ab516613b1a4abd8e99d0dc3fefedd72ee754a20b8e387ad1be08ff9fce055a832f0857
65ed9b6e276c65c440411971185c2a30b1028a37cf629259875e7e1f9eb5f936b1ee4b10c3ff5c3d874dd660e
3e14294bfdd0f3ef374c0e09cbf776ee4ad357e877c316930b301805c6fef34069a42eb2bc2b53fde1d67b2e0
20a666303dcc55e0e4d280b0d6bf7f12f693bba482c4c45567bc18fc8b678cb11a837a6f4a440379afd5528b82
00b9c2e2e0f8e640a1d7509ba15a5e6c5fcc91e73d05e6bdba459d6f4a9ba7684352c7c16760153394e5afe7af
e64273fddc2c74208e3e14909eb0fc098e95ec330ba3209fd2af52a93fbc8a20622b32601d02043ef6e12ffe37
9999089db2989668bd52a5933f336a5d305cae11d5320ef2ab093633d32811e6f276bbf93ee9f2e4db0899a94c
0d2bbeeb9288ee446da3c703b60d269ebada8463244b5a208f8e2fde7a47d4c545646b15709b57e44395e6a227
f2391c559ea5343bdc1c86245def3dc50a18462227aec2d12058e8f726564f8f5caf78e7de2be563a3d87f569c
da1056b867238e8d636141ae969747e309e942f94e14763bf51f723b573efeac7562d4904e22f7884881b1ad6b
d1842c21960c211847b0dd72b5291dec1f4287bf0b3fefec7c7b7819ee583520ce5f71b3c275d771afcbab9319
0838bd17224d70039a15859ed6e349745034a4cd1f3a57dcf355fcb875c239e6ed1ca60a802ae13a4e42da6411
2bc222dc12e686a38924d1a8e7a4ff9defb66c3aa80ac991596f59e78aecda545016aea95fa76d328608c5a8b5
9ef286493ab2f50b54b8d6f36b92294f530c3e1ae36c97645affd863f1e631059b936991844b32b1918c286b45
b9af1ed915176aec31daa9789ce2af1da9a6ab30e1d2b46bbe601788b35e2d1aebf5b066e01ef1ce03451e5e1a
4fa7685d244ba9b7cbcdc55e0b2d8420b9180f0ab9b4ce7df937461041d27627817f6d3a3da2b95fa374e30506
673a46507fe7315be113ec2a3cae3f04dbe15786b974be26972f5072c04059d92819a09603fed57bbfdf3ac8fb
419198fad70d2af3065cb5901722f728f41f9cf7257cb66c1aff614966b1791393b53af240684fb2e1933eb52c
05ccb8cd9e971f68135c9cec921e7bd24d1756b2691b9aa4345279cfdf4813ae3c20048bcd3727f9f53de55017
a4ab1e10de347d996df7710177b5d587da932e32348ed97ace34007479cb1ef6eb08fe2ef1c795b173a3d5cbeb
384b80ce75db24ca2127dc89c3d868a3e646a1ef028d853ff37cc5b2c4cbf2be0c853bd4ab58fb5ea5e050c0db
13e9569682fdae28133cde1534933ab81fcfe5626d19f536e0299b40cbb5973b2b49a81716ecc1bb3a1047f14c
ce3428c1a81fb2a870562850f63e31c0e330ede29c0c4428d11d7b2a805fe115ef7b12ee7f9106afbb5c83b28f
2ee7e36469b1aefa39a63fdc0a10a46698135725bc13ecdb6ff967456eabe8bfbdef687feb023acc348f737b57
793d1976cba0bf1d57a441a732d0c9ac4dd41f9d3015a533fee25f4b515f22672611f139ebb8fa110ef1a30934
0e478e8bcff3cc520a8a01f0521cb27af8ec3a03b5f132e0e7ba5bc9e60ad506b943167a23371a50802881c916
79209104bdbbea08c14f9de58710f7576ed5326df09a547f168bab84319b5c45765577f1f1971e3b99037197fe
f36adc16b23cdf3fd4ec39a464983ab3a95868d194d9f5d9165baf860e280ce72b5d2c2471da1643aad321b74d
39c328fe86452effb672209b7cc75e262545b5c60c214742ba18dc66d90d9f25c6108d0b8037a6fc264e83664a
23893041877d528cbc703b8e11ee060d5429df8901bd60c3cd36c7fb07625e97d4990bf32ece1d7afb095c0286
5f0929b418fec1983b5ea7ddfd3d87073fbf7989c8433aaa1971f8f88d3c367c53a864bd433f61787647dd1f06
d4508ddfa3d6a39ee3cf677e67ffb6e9ff66d5f2abb440520ccf4ed2bd8ca212da0828ecd779bb083f726356df
4d4df7156786d7f88d740d62ea20ad4a4b6db85f479cf5852cb940b287f84c573c495322ec7148b7fd59429fcd
407de81f542e1cc077ef58c7eb1e299b34e59f636a02290c78c277faf1c1db96d62ab3f883bf46289e34254e2c
748ff2a9057caf3acd23f5d896d9e443584ffcae98cbe14d0fae6a7fee492223d712ce5455c0226dba9901d1ac
f8b71a67d4c45a24f72a85e8c66c32b2d92fb83dcb077606bab5ecc86858d4b27eaf4ed3933d3bb2790d5b3111
5417ee7b9476f09f21e65bb8cba20ef7701a1db227b482826d6f7cac909cb5ebfd9680ae92485da66c14d8b5b2
dd11302137ba18748f8a0441070e685f14202019d22bbc7f478f8bdd3ba6a36a7af6cc74d64ed374b69b33bfce
f4475440e9f5478169e6a22061a5db5951d122849241f3812b7e3e47c2e7fa46d649b56ca9b4c1997ce56f6c62
63229aeb5e511d9c0af161b4df5eeb921524dd65af599bb0e10692bc0e9b3a0055e3c86b5b92a43ff7513200f0
175841c71138fb0bad07b48e03dbc06c4ebbc180fa5bac461a7596257da52bea2492336fa5d8a2966197346a56
6e4846387f274c98c9a87ed44db1eaa99437b2aedbb134452fef808ce6e83d997659b40b868b132a2402ce0279
fe6eff3c3e587edd67ebbd9665945beddc0f81d20199339390b909a34b4a11f922e3e7146d729f4a89be45db0f
e0e8e62ade4dda25cfe6c6e5402ae82635a1c122582d81664a13cb9803f7635fc436dd93fa8e8153ee5f2433d9
d3b12f047d60333c06a0a989de3bb39d5f83003e8fd64c3b2ecb76060e2e51c292810af4320e7ccebffb757e1e
9b12b36860a36f676ce1139ef6e5d14518cad58189ee8fbfe11e9cdd1017e985578ba3e9c26c3bf2e56b81a725
758a38af5f8e015573a109c0ebe38d428bd39c3256135a5562143249ffefe4a51797edbc8025ff77e4c4c150e2
47ec959d303c44896633df36c794e540cc3558ed6043ac00e79ffdef495132a4cfe7e8d14cc4b240e1a29d2590
46e7b5ce9f1a646c82468d65cd5c34091057212b45e3065806f081a7922234d00f25df8d1727fe11e6b16658e4
3fd26ea64481e75ff995090be522c71405c9b3e436d0793357e0c058c459f4f6f3bc59c9ae5c6c4dc20e703885
f58d21a1febf414dc3ac40cf5dd3f4cc0287c392abacc6fc4ebebad9615e431ccf3781186ff4d012561a59635c
36de6c98d02a7473ebd0c4ce80266dedfd60d8101e38410ddf0a80ca4f4fa295a0022d58a700ae5f6fafcda733
81391ea83413a7c10823a3cfb0784cc47ba7e4cb9d8bfc83cad29d30c79c3cbcf2b849c00d4ccbcb4e09daa97c
96863634fb330c24bcb558dfa2f0bac03cfbc3a0f33ec0d47bfd71b14f493c0f377e57844bf36e9e9b8fc9bdd4
312f8c49f8d78f8eee2b735ee927f53fc1c1d45a8caa3194cb8f949e37e731acb2a13926f7164d61aa05bca7b7
7bd2f0b08973b44627d97db66ac34e0dde4687aa43a4ed88624a351baa0166b6665f76fa57fe735c46a95a3164
e7b8369513121e820b0cf4efea3c1cd1561bfd5c2e73c104b53434a8d3e7a1c2b5bc95f5556fdd5f7d8f563c6b
cafc0ef6b6f675aceed7dc7d56d8c8eb033d0b97dee22951b706bd240812271d8942be0dad42cc814850f5aad7
583b4b7f6085e070256882ff1f482f0527b77368a9a61dbc51866ef608f5e79e4437f3fe16359aed66111b9cf0
71ece0e6c34ad76775cda075054aada979bc3bda3892d0832ceb1c49ac62759a10bcf1e585e05f359aa3fa4adb
f6402d05d34fb05da32eb4b4ac3b9207def36f8f04ace7dde98ca85be96fc595afdb309b80e3542f65f987cf13
0e2c1f49a9a0871b8f246267bd49d4911e89f7c2db8d7a2c450a569436f455fd1786e11f8554f13a8e29a1a4c8
```

```
dd888a69a9042b891ca38e2a4b95db9afb165f4b4bcab4ae27b9560b3fbc44d71e4680e20d0db8bc239363b0dd
f3845cc3709f9bdf6f2d901c99d4ba68f5ae38fa07c215b875f6da7507d419bb21be431ce7e7c23fd6dd03e3e2
a84ba27'
}
```

CBOR bytes:

```
a16464617461590c7a0bc798e4d869d17c321e24c3111d11fd2fc955dad221bb803e08ebf309bd7b94d57f3162
d5370f09415145645d785fdb0d3da84201227d6feb7f0bea059e954d6f74d94c223e589ae44edfff85b9bc5188
7d71eb07ac40ce57da3bb010aa23de8a6c46998a024e1a1f183add692b572bdd25f7d30f96427cbfeb117112f9
7037d002893c45d049eaa56413cf23bf54f519ea3ca5ae984917b11b21cd20a199998c93f55fd271ad442231fd
0bc720394a756c69b6e86785ed09ea5866b48a2567edd91ab74d4234e6a12b4adf831146e0a73907a755562103-
2b8c5d548075b16a1b5ade26eaa97c77866460a8067b4fbf58ff887fb6d33572122c70d724636fae4fd57f17ad
388326474cf69d7f6183a6894552267ec7c9aa3b5e251ac8afb5900b02353e591edf245c403b9411514e60fdfe
b6d5bcb907be2eae078999994a1726ab516613b1a4abd8e99d0dc3fefedd72ee754a20b8e387ad1be08ff9fce
055a832f085765ed9b6e276c65c440411971185c2a30b1028a37cf629259875e7e1f9eb5f936b1ee4b10c3ff5
c3d874dd660e3e14294bfdd0f3ef374c0e09cbf776ee4ad357e877c316930b301805c6fef34069a42eb2bc2b5
3fde1d67b2e020a666303dcc55e0e4d280b0d6bf7f12f693bba482c4c45567bc18fc8b678cb11a837a6f4a4403
79afd5528b8200b9c2e2e0f8e640a1d7509ba15a5e6c5fcc91e73d05e6bdba459d6f4a9ba7684352c7c1676015
3394e5afe7afe64273fddc2c74208e3e14909eb0fc098e95ec330ba3209fd2af52a93fbc8a20622b32601d0204
3ef6e12ffe379999089db2989668bd52a5933f336a5d305cae11d5320ef2ab093633d32811e6f276bbf93ee9f
2e4db0899a94c0d2bbeeb9288ee446da3c703b60d269ebada8463244b5a208f8e2fde7a47d4c545646b15709b
57e44395e6a227f2391c559ea5343bdc1c86245def3dc50a18462227aec2d12058e8f726564f8f5caf78e7de2b
e563a3d87f569cda1056b867238e8d636141ae969747e309e942f94e14763bf51f723b573efeac7562d4904e22
f7884881b1ad6bd1842c21960c211847b0dd72b5291dec1f4287bf0b3fefec7c7b7819ee583520ce5f71b3c275
d771afcbab93190838bd17224d70039a15859ed6e349745034a4cd1f3a57dcf355fcb875c239e6ed1ca60a802a
e13a4e42da64112bc222dc12e686a38924d1a8e7a4ff9defb66c3aa80ac991596f59e78aecda545016aea95fa
76d328608c5a8b59ef286493ab2f50b54b8d6f36b92294f530c3e1ae36c97645affd863f1e631059b936991844
b32b1918c286b45b9af1ed915176aec31daa9789ce2af1da9a6ab30e1d2b46bbe601788b35e2d1aebf5b066e01
ef1ce03451e5e1a4fa7685d244ba9b7cbcdc55e0b2d8420b9180f0ab9b4ce7df937461041d27627817f6d3a3da
2b95fa374e30506673a46507fe7315be113ec2a3cae3f04dbe15786b974be26972f5072c04059d92819a09603f
ed57bbfdf3ac8fb419198fad70d2af3065cb5901722f728f41f9cf7257cb66c1aff614966b1791393b53af2406
84fb2e1933eb52c05ccb8cd9e971f68135c9cec921e7bd24d1756b2691b9aa4345279cfdf4813ae3c20048bcd3
727f9f53de55017a4ab1e10de347d996df7710177b5d587da932e32348ed97ace34007479cb1ef6eb08fe2ef1c
795b173a3d5cbeb384b80ce75db24ca2127dc89c3d868a3e646a1ef028d853ff37cc5b2c4cbf2be0c853bd4ab5
8fb5ea5e050c0db13e9569682fdae28133cde1534933ab81fcfe5626d19f536e0299b40cbb5973b2b49a81716e
cc1bb3a1047f14cce3428c1a81fb2a870562850f63e31c0e330ede29c0c4428d11d7b2a805fe115ef7b12ee7f9
106afbb5c83b28f2ee7e36469b1aefa39a63fdc0a10a46698135725bc13ecdb6ff967456eabe8bfbdef687feb0
23acc348f737b57793d1976cba0bf1d57a441a732d0c9ac4dd41f9d3015a533fee25f4b515f22672611f139ebb
8fa110ef1a309340e478e8bcff3cc520a8a01f0521cb27af8ec3a03b5f132e0e7ba5bc9e60ad506b943167a233
71a50802881c91679209104bdbbea08c14f9de58710f7576ed5326df09a547f168bab84319b5c45765577f1f19
71e3b99037197fef36adc16b23cdf3fd4ec39a464983ab3a95868d194d9f5d9165baf860e280ce72b5d2c2471d
a1643aad321b74d39c328fe86452effb672209b7cc75e262545b5c60c214742ba18dc66d90d9f25c6108d0b803
7a6fc264e83664a23893041877d528cbc703b8e11ee060d5429df8901bd60c3cd36c7fb07625e97d4990bf32ec
e1d7afb095c02865f0929b418fec1983b5ea7ddfd3d87073fbf7989c8433aaa1971f8f88d3c367c53a864bd433
f61787647dd1f06d4508ddfa3d6a39ee3cf677e67ffb6e9ff66d5f2abb440520ccf4ed2bd8ca212da0828ecd77
9bb083f726356df4d4df7156786d7f88d740d62ea20ad4a4b6db85f479cf5852cb940b287f84c573c495322ec7
148b7fd59429fcd407de81f542e1cc077ef58c7eb1e299b34e59f636a02290c78c277faf1c1db96d62ab3f883b
f46289e34254e2c748ff2a9057caf3acd23f5d896d9e443584ffcae98cbe14d0fae6a7fee492223d712ce5455c
0226dba9901d1acf8b71a67d4c45a24f72a85e8c66c32b2d92fb83dcb077606bab5ecc86858d4b27eaf4ed3933
d3bb2790d5b31115417ee7b9476f09f21e65bb8cba20ef7701a1db227b482826d6f7cac909cb5ebfd9680ae924
85da66c14d8b5b2dd11302137ba18748f8a0441070e685f14202019d22bbc7f478f8bdd3ba6a36a7af6cc74d64
ed374b69b33bfcef4475440e9f5478169e6a22061a5db5951d122849241f3812b7e3e47c2e7fa46d649b56ca9b
4c1997ce56f6c6263229aeb5e511d9c0af161b4df5eeb921524dd65af599bb0e10692bc0e9b3a0055e3c86b5b9
2a43ff7513200f0175841c71138fb0bad07b48e03dbc06c4ebbc180fa5bac461a7596257da52bea2492336fa5d
8a2966197346a566e4846387f274c98c9a87ed44db1eaa99437b2aedbb134452fef808ce6e83d997659b40b868
b132a2402ce0279fe6eff3c3e587edd67ebbd9665945beddc0f81d20199339390b909a34b4a11f922e3e7146d7
29f4a89be45db0fe0e8e62ade4dda25cfe6c6e5402ae82635a1c122582d81664a13cb9803f7635fc436dd93fa8
e8153ee5f2433d9d3b12f047d60333c06a0a989de3bb39d5f83003e8fd64c3b2ecb76060e2e51c292810af4320
e7ccebffb757e1e9b12b36860a36f676ce1139ef6e5d14518cad58189ee8fbfe11e9cdd1017e985578ba3e9c26
c3bf2e56b81a725758a38af5f8e015573a109c0ebe38d428bd39c3256135a5562143249ffefe4a51797edbc802
5ff77e4c4c150e247ec959d303c44896633df36c794e540cc3558ed6043ac00e79ffdef495132a4cfe7e8d14cc
4b240e1a29d259046e7b5ce9f1a646c82468d65cd5c34091057212b45e3065806f081a7922234d00f25df8d172
7fe11e6b16658e43fd26ea64481e75ff995090be522c71405c9b3e436d0793357e0c058c459f4f6f3bc59c9ae5
c6c4dc20e703885f58d21a1febf414dc3ac40cf5dd3f4cc0287c392abacc6fc4ebebad9615e431ccf3781186ff
4d012561a59635c36de6c98d02a7473ebd0c4ce80266dedfd60d8101e38410ddf0a80ca4f4fa295a0022d58a70
0ae5f6fafcda73381391ea83413a7c10823a3cfb0784cc47ba7e4cb9d8bfc83cad29d30c79c3cbcf2b849c00d4
ccbcb4e09daa97c96863634fb330c24bcb558dfa2f0bac03cfbc3a0f33ec0d47bfd71b14f493c0f377e57844bf
36e9e9b8fc9bdd4312f8c49f8d78f8eee2b735ee927f53fc1c1d45a8caa3194cb8f949e37e731acb2a13926f71
64d61aa05bca7b77bd2f0b08973b44627d97db66ac34e0dde4687aa43a4ed88624a351baa0166b6665f76fa57f
e735c46a95a3164e7b8369513121e820b0cf4efea3c1cd1561bfd5c2e73c104b53434a8d3e7a1c2b5bc95f5556
```

```
fdd5f7d8f563c6bcafc0ef6b6f675aceed7dc7d56d8c8eb033d0b97dee22951b706bd240812271d8942be0dad4
2cc814850f5aad7583b4b7f6085e070256882ff1f482f0527b77368a9a61dbc51866ef608f5e79e4437f3fe163
59aed66111b9cf071ece0e6c34ad76775cda075054aada979bc3bda3892d0832ceb1c49ac62759a10bcf1e585e
05f359aa3fa4adbf6402d05d34fb05da32eb4b4ac3b9207def36f8f04ace7dde98ca85be96fc595afdb309b80e
3542f65f987cf130e2c1f49a9a0871b8f246267bd49d4911e89f7c2db8d7a2c450a569436f455fd1786e11f855
4f13a8e29a1a4c8dd888a69a9042b891ca38e2a4b95db9afb165f4b4bcab4ae27b9560b3fbc44d71e4680e20d0
db8bc239363b0ddf3845cc3709f9bdf6f2d901c99d4ba68f5ae38fa07c215b875f6da7507d419bb21be431ce7e
7c23fd6dd03e3e2a84ba27
```

The following are the private keys stated earlier but in pkcs8 format:

Key agreement privkey mdl pkcs8:

```
304d020100301306072a8648ce3d020106082a8648ce3d030107043330310201010420 90ae586a95b305c7354a
efa49bb267ee4ec74816b7494bdbd0b0e6e98f6718a5a00a06082a8648ce3d030107
```

Key agreement pubkey reader pkcs8:

```
3059301306072a8648ce3d020106082a8648ce3d0301070342000418cd31cc15e50d45c3597e2b9ecab5771a5
f61fc4290819415006251ef180c8efc159b5d687bdd4580124480c3a7474ece63234405f6126dc65653c25d57
3cd2
```

Key agreement privkey reader pkcs8:

```
304d020100301306072a8648ce3d020106082a8648ce3d030107043330310201010420bdba90434feb6113986a
7a4ab214432933423f842fd2f0f889398c3798b3dafaa00a06082a8648ce3d030107
```

Key agreement pubkey mdl pkcs8:

```
3059301306072a8648ce3d020106082a8648ce3d03010703420004e4706de318a40d0bd8648b7907b0283f74
45370241ff1fdd77f08d6a598be90279767fbe391223f61dcd5e980133a035b4918f6f9de41b3ceb8d80186
0df8859
```

D.5.2 Issuer data authentication

The following is an example of an MSO, as also included in D.4.1.2:

```
{
  "digestAlgorithm": "SHA-256",
  "valueDigests": {
    "nameSpaces": {
      "org.iso.18013.5.1": {
        1: h'AF307AD59C28C5032C1A49394E0BCE7CB673A533D9AD12BD4CA980F6F12DB5BE',
        2: h'76AAC7B52F938A4AEAE31D796BA5AEB2DBE0C9D96AD75B18D75AFAE875A79C76',
        3: h'E2503B905A1754EF483BF5DD429C3A9BC66BDF7B3D1C080CFDEB4EC27AADEE63',
        4: h'A963382AA8BA4C52DA522BC31D41E2564915F57CB144CB7C69F47A85B44A7E59',
        5: h'20EE714191901E777AE7F45B2EB6554ACA8DBB10F3093BE267DCE2C7F204D8E1',
        6: h'B53392E547BA0A2369CBA5AC6D91F430FA8FB76F0C4DF90FCB5791B6FF038EB6',
        7: h'4E0AD333B3BFD242DF9552E295A4D85EFD82E349456EA10FA80EEDD6DC3195C0',
        8: h'467AC91E0ACDEAC85D8272401F8749BC2E1B3F46385E5B91DBB71D8B9A8261EE',
        10: h'BFAD1F0BA42F3FE57068A2AFD4F2051EA60BB91CB0813FE98B323D969E218B45',
        11: h'541E4DE68CDD216DA9AEC22C9122E1D3C25D38CFA26455DFF5882CC2C371F514',
        19: h'41314B909ED36016CF9042E7B94AAC41955D459AB258A18D4D87AD5817D6C68B',
        23: h'D536408A50EC2818E426727078BD4429A2D9A0E061FECD46F5CFDAA77ADA95C0',
        24: h'580BAE0ECF9A1C4D162586D7520615E2A7AC14584967FC9BBE8373E59302CA98'
      },
      "org.iso.18013.5.1.Utopia": {
        52: h'8D5FFF80E966D52C123C34B07BD6546909AD7AE6BF48D736E0865A8B99853B7F'
      }
    }
  },
  "deviceKey": {
    1: 2,
    -1: 1,
    -2: h'8553FBB8C982DB3F5A45D6FB12DFA04C0CF7A48F43657F203B3DD05BA8D62392',
    -3: h'334E2EDC3DDC0549F8C8DC79C10FA3BFE26B389F6AAD3E0603A6FCAE1BBFEA45'
  },
  "docType": "org.iso.18013.5.1.mDL",
  "validityInfo": {
    "signed": 0(
```

```
      "2019-08-08T00:00:00Z"
    ),
    "validFrom": 0(
      "2019-08-24T00:00:00Z"
    ),
    "validUntil": 0(
      "2029-02-28T00:00:00Z"
    )
  }
}
```

The following is an example of an IssuerAuth structure, as also included in D.4.1.2:

```
[
  h'A10126',
  {
    33:
h'308201D230820178A00302010202104C7FF615C64916807A58CBE67BA1D0CE300A06082A8648CE3D04030230
34310B300906035504061302553310F300D060355040A0C0655746F706961311430120603550403 0C0B55746F
706961204941434130 1E170D3139313030313030303030305A170D3232313030313030303030305A3032310B3
009060355040613025553310F300D060355040A0C0655746F706961311230100603550403 0C0955746F706961
204453305930130 6072A8648CE3D020106082A8648CE3D030107034200 04F2736B046885DF28F1D7E5115E448
9DC85CCA129BAF16D5DDA4E8F5DACC122E7D860FEBBCE074F18F8B3B2DDEF2D4AC201C8A60E48FC89DF413F52F
4761DE0BFA36E306C301D0603551D0E041604144FC593C52459ADFAF156949B969C3208C841A02D301F0603551
D230418301680 14C757E4317E6908CABF0F0FD7566E438B2C027DE130090603551D1304023000300B0603551D
0F0404030205A030120603551D25040B3009060728818C5D050102300A06082A8648CE3D040302034800304502
21008C13A970FCE5E50DE2D364158167EF17BF350D6D9D574968BC053953EE895BE4022005C506A1426BD3332C
E6B69FDC571F7B3724E4AC5501FCCA61036BE12AD1502E'
  },

h'A56F646967657374416C676F726974686D675348412D3235366C76616C7565446967657374 73A16A6E616D65
537061636573A2716F72672E69736F2E31383031332E352E31AD015820AF307AD59C28C5032C1A49394E0BCE7
CB673A533D9AD12BD4CA980F6F12DB5BE02582076AAC7B52F938A4AEAE31D796BA5AEB2DBE0C9D96AD75B18D7
5AFAE875A79C76035820E2503B905A1754EF483BF5DD429C3A9BC66BDF7B3D1C080CFDEB4EC27AADEE6304582
0A963382AA8BA4C52DA522BC31D41E2564915F57CB144CB7C69F47A85B44A7E5905582020EE714191901E777A
E7F45B2EB6554ACA8DBB10F3093BE267DCE2C7F204D8E1065820B53392E547BA0A2369CBA5AC6D91F430FA8FB
76F0C4DF90FCB5791B6FF038EB60758204E0AD333B3BFD242DF9552E295A4D85EFD82E349456EA10FA80EEDD6D
C3195C0085820467AC91E0ACDEAC85D8272401F8749BC2E1B3F46385E5B91DBB71D8B9A8261EE0A5820BFAD1F0
BA42F3FE57068A2AFD4F2051EA60BB91CB0813FE98B323D969E218B450B5820541E4DE68CDD216DA9AEC22C912
2E1D3C25D38CFA26455DFF5882CC2C371F51413582041314B909ED36016CF9042E7B94AAC41955D459AB258A18
D4D87AD5817D6C68B175820D536408A50EC2818E426727078BD4429A2D9A0E061FECD46F5CFDAA77ADA95C0181
85820580BAE0ECF9A1C4D162586D7520615E2A7AC14584967FC9BBE8373E59302CA9878186F72672E69736F2E3
1383031332E352E312E55746F706961A1183458208D5FFF80E966D52C123C34B07BD6546909AD7AE6BF48D736E
0865A8B99853B7F696465766963654B6579A401022001215820 8553FBB8C982DB3F5A45D6FB12DFA04C0CF7A48
F43657F203B3DD05BA8D62392225820334E2EDC3DDC0549F8C8DC79C10FA3BFE26B389F6AAD3E0603A6FCAE1BB
FEA4567646F6354797065756F72672E69736F2E31383031332E352E312E6D444C6C76616C6964697479496E666
FA3667369676E6564C074323031392D30382D30385430303A30303A30305A6976616C696446726F6DC07432303
1392D30382D32345430303A30303A30305A6A76616C6964556E74696CC074323032392D30322D32385430303A3
0303A30305A',

h'D115FF9FBF1CF89E5B8DF4A0E98EA8337C6C8C5BC81E2EE98C43526DC47E7D9696BA7F45F011B2FBAA8FA33B
003F22A2A5B11FF896006878E6A17B4E9B918772'
]
```

D.5.3 mDL Authentication

The following is an example of the DeviceAuthentication structure, based on D.4.1.2

```
[
  "DeviceAuthentication",
  [
    24(

h'8663312e308201d818584ba401022001215820e4706de318a40d0bd8648b7907b0283f7445370241ff1fdd77
f08d6a598be90222582079767fbe391223f61dcd5e980133a035b4918f6f9de41b3ceb8d801860df8859818302
01a201f50b5000000005000010008000008 05f9b34ffa080a16576616c7565697465737456616c7565'
    ),
    24(

h'A401022001215820 18CD31CC15E50D45C3597E2B9ECAB5771A5F61FC4290819415006251EF180C8E225820FC
```

```
159B5D687BDD4580124480C3A7474ECE63234405F6126DC65653C25D573CD2'
    )
  ],
  "org.iso.18013.5.1.mDL",
  24(
    h'A0'
  )
]
```

CBOR bytes:

```
847444657669636541757468656e7469636174696f6e82d81858828663312e308201d818584ba4010220012158
20e4706de318a40d0bd8648b7907b0283f7445370241ff1fdd77f08d6a598be90222582079767fbe391223f61d
cd5e980133a035b4918f6f9de41b3ceb8d801860df885981830201a201f50b5000000005000010008000008 05f
9b34ffa080a16576616c756569746573745646 16c7565d818584ba40102200121582018cd31cc15e50d45c3597e
2b9ecab5771a5f61fc4290819415006251ef180c8e225820fc159b5d687bdd4580124480c3a7474ece63234405-
f6126dc65653c25d573cd2756f72672e69736f2e31383031332e352e312e6d444cd81841a0
```

The EMacKey used to calculate the MAC is:

```
b4392f7ada7de625dcf0e375ba926f7630323549cb99047761385388d942e8de
```

The resulting ‘tag’ based on the earlier example data is:

```
8d244815695c2467a59c2c2bbabf611c489a46fd7e758b3e3a6a00e33a5ed470
```

The resulting structure and calculcated MAC can be found in the response example in D.4.1.2.

D.5.4 JWS

The header and signature with the response from D.4.2.1.2 as payload are:

JWT header:

```
{
  "alg": "ES256",
  "typ": "JWT",
  "x5c": [

"MIICDzCCAbWgAwIBAgITDzSOSBpQWt+y8Z/ffgCCwpy5STAKBggqhkjOPQQDAjBdMQswCQYDVQQGEwJVUzELMAk
GA1UECBMCQ08xEzARBgNVBAcTCkNlbnRlbm5pYWwxLDAqBgNVBAMTI1V0b3BpYSBEZXBhcnRtZW50IG9mIE1vdG9
yIFZlaGljbGVzMB4XDTE5MTAxMTA2MDAwMFoXDTE5MTExMDA3MDAwMFowSzETMBEGA1UEBxMKQ2VudGVubmlhbDEL
MAkGA1UECBMCQ08xCzAJBgNVBAYTAlVTMRowGAYDVQQDExFVdG9waWEgSldUIFNpZ25lcjBZMBMGByqGSM49AgEGC
CqGSM49AwEHA0IABIm3ob+EOWNMiXrgGa+ewTVDxR9uTDksilcafpVQI6MP05ux6y3yZU6MmNvhuufvmt1HoGyDR/
CkvMP3ehERO/ijZjBkMA4GA1UdDwEB/wQEAwIHgDAdBgNVHQ4EFgQUCvHtitaf1uK8l8qJgPN4o9kLXNMwHwYDVR0j
BBgwFoAUqEtwixMUQg9De/SaJTjE70ebqrMwEgYDVR0lBAswCQYHKIGMXQUBAzAKBggqhkjOPQQDAgNIADBFAiAOVK
J8j/HS3E4nd3g+EotIfxZw3MpfG5KeeujJUHtpsQIhAOww3lERUE4scEvmkURtXtcFcq0fCWU1HM1qqLBEQNwJ"
  ]
}
```

JWT signature:

```
Ay6YOq1A6ECFehbwPidtlO-aHm2l5A3_nhK6ogM8j8rYglvNwXALTs5G8shEprAUUHDEcLelBBxN08gbyPcfQQ
```

Full signed JWT:

```
eyJhbGciOiJFUzI1NiIsInR5cCI6IkpXVCIsIng1YyI6WyJNSUlDRHpDQ0FiV2dBd0lCQWdJVER6U09TQnBRV3QreT
haXC9mZmdDQ3dweTVTVEFLQmdncWhrak9QUVFEQWpCZE1Rc3dDUVlEVlFRR0V3SlZVekVMTUFrR0ExVUVDQk1DUTA
4eEV6QVJCZ05WQkFjVENrTmxiblJsYm01cFlXd3hMREFxQmdOVkJBTVRJMVYwYjNCcFlTQkVaWEJoY25SdFpXNTBJ
RzltSUUxdmRHOXlJRlpsYUdsamJHVnpNQjRYRFRFNU1UQXhNVEEyTURBd01Gb1hEVEU1TVRFeE1EQTNNREF3TUZvd
1N6RVRNQkVHQTFVRUJ4TUtRMlZ1ZEdWdWJtbGhiREVMTUFrR0ExVUVDQk1DUTA4eEN6QUpCZ05WQkFZVEFsVlRNUm
93R0FZRFZRUURFeEZWZEc5d2FXRWdTbGRVSUZOcFoyNWxjakJaTUJNR0J5cUdTTTQ5QWdFR0NDcUdTTTQ5QXdFSEE
wSUFCSW0zb2IrRU9XTk1pWHJnR2ErZXdUVkR4Ujl1VERrc2lsY2FmcFZRSTZNUDA1dXg2eTN5Wlu2TW1Odmh1dWZ2b
XQxSG9HeURSXC9Da3ZNUDNlaEVST1wvaWpaakJrTUE0R0ExVWREd0VCXC93UUVBd0lIZ0RBZEJnTlZIUTRFRmdRVUN
2SHRpdGFmMXVLOGw4cUpnUE40bzlrTFhOTXdId1lEVlIwakJCZ3dGb0FVcUV0d2l4TVVRZzlEZVwvU2FKVGpFNzBlY
nFyTXdFZ1lEVlIwbEJBc3dDUVlIS0lHTVhRVUJBekFLQmdncWhrak9QUVFEQWdOSUFEQkZBaUFPVktOGpcL0hTM0U
0bmQzZytFb3RJZnhadzNNcGZHNUtlZXVqSlVIdHBzUUloQU93dzNsRVJVRTRzY0V2bWtVUnRYdGNGY3EwZkNXVTFIT
TFxcUxCRVFOd0oiXX0.eyJ2ZXJzaW9uIjoiMS4wIiwiZG9jVHlwZSI6Imlzby5vcmcuMTgwMTMuNS4xLm1ETCIsIm5
hbWVTcGFjZXMiOnsib3JnLmlzby4xODAxMy41LjEiOnsiZmFtaWx5X25hbWUiOiJUVVJORVIiLCJnaXZlbl9uYW1lI
```

joiU1VTQU4iLCJiaXJ0aF9kYXRlIjoiMTk5OC0wOC0yOCIsImlzc3VlX2RhdGUiOiIyMDE4LTAxLTE1VDEwOjAwOjA
wLjAwMDAwMDAtMDc6MDAiLCJleHBpcnlfZGF0ZSI6IjIwMjItMDgtMjdUMTI6MDA6MDAuMDAwMDAwMC0wNjowMCIsI
mlzc3VpbmdfY291bnRyeSI6IlVTIiwiaXNzdWluZ19hdXRob3JpdHkiOiJDTyIsImRvY3VtZW50X251bWJlciI6IjU
0MjQyNjgxNCIsImRyaXZpbmdfcHJpdmlsZWdlcyI6W3siY29kZXMiOlt7ImNvZGUiOiJEIn1dLCJ2ZWhpY2xlX2Nhd
GVnb3J5X2NvZGUiOiJEIiwiaXNzdWVfZGF0ZSI6IjIwMTktMDEtMDEiLCJleHBpcnlfZGF0ZSI6IjIwMjctMDEtMDE
ifSx7ImNvZGVzIjpbeyJjb2RlIjoiQyJ9XSwidmVoaWNsZV9jYXRlZ29yeV9jb2RlIjoiQyIsImlzc3VlX2RhdGUiO
iIyMDE5LTAxLTAxIiwiZXhwaXJ5X2RhdGUiOiIyMDE3LTAxLTAxIn1dLCJ1bl9kaXN0aW5ndWlzaGluZ19zaWduIjo
iVVNBIiwicG9ydHJhaXQiOiJcLzlqXC80QUFRU2taSlJnQUJBUUVBa0FDUUFBRFwvMndCREFCTU5EaEVPREJNUkR4R
VZGQk1YSFRBZkhSb2FIVG9xTENNd1JUMUpSMFE5UTBGTVZtMWRURkZvVWtGRFg0SmdhSEYxZTN4N1NseUdrSVYzajI
xNGUzYlwvMndCREFSUVZGUjBaSFRnZkh6aDJUME5QZG5aMmRuWjJkbloyZG5aMmRuWjJkbloyZG5aMmRuWjJkbloyZ
G5aMmRuWjJkbloyZG5aMmRuWjJkbloyZG5aMmRuYlwvd0FBUkNBQVlBR1FEQVNJQUFoRUJBeEVCXC84UUFHd0FBQXd
FQUF3RUFBQUFBQUFBQUFBQUFBQVVHQkFFQ0F3ZlwveEFBeUVBQUJBd01EQWdVQ0F3a0FBQUFBQUFBQkFnTUVBQVVSQ
mhJaEV6RVVGVkZoY1NKQkI0R2hGalZDVW5PUnNzSHhcLzhRQUZRRUJBUUFBQUFBQUFBQUFBQUFBQUFBSFwveEF
BYUVRRUJBUUFEQVFBQUFBQUFBQUFBQUFBQUFVRVJJVEZoXC85b0FEQU1CQUFJUkF4RUFQd0NsdTk0aTJpTXB4OWFTd
kgwTkFcL1VzK3dcLzNYbnArOCtkd2x5T2gwTnJoUnQzN3M4QTV6Z2V0SzlSNmZqTGJ1TjBkVXRidlN5aFBaS1NBQm4
zN1VmaFwvKzVYXC9BT3VmOFUwaFhlWnE4SW5PUkxmYjNweTJpUW9vT08zZkdBZVBldDFpMUJIdlRibXhDbVhXdVZvV
WM0SHFEVWxia3pKMVwvbXU2ZGNFVUVFcUxwQkJCUHBnOVwvd0JQV3ZYVFMwdE0zbU10Q1wvSDlGWks5MlJ4a0VmT1R
UQyttcjJ0VWwxMFFiYzlLWmE1VzZGWUFIcndEeDg0cDNaN3ZIdkVQeEVmY25hZHEwcTdwTlRlaHVuNVBjTjJPXC93Q
lh4dFwvN1hob1poVXFEZFk1VVVvZFFsRzdHY0VoUXpRTjd6ckNMYlgwc3gyMHpGXC94N1hNQlB0bkJ5YWNYRzRNVzJ
DdVZKSkNFanNPU1Q5Z0tnZFZXZU5abHcyWTI0bFNWRmExSEpVY2l2b1Q2bzZZNDhXV2cyZUQxY1lcL1dtR3BuOXR5
a0lkZHRMNklxemhMdTd2OGNZUDk2cVl6NkpVZHQ5bzViY1NGSlBzYWk5WVJwYW9xSkRMekNyUWdwNmJUSkF4eGpQQ
XgrcDcweWExVkFnV3FBcFVkOUtIV3lFSWJBVnQybmJqSklwZzM2aXZvc2JEVG5RNjZuRkZJVHYyNHdPXC9ZMGxqYT
g4UkphWjh1MjlSWVRucjV4azRcL2xybStzbzFLeEFreDVrZU1qbmFpU29KVVNWQWRobjBySGMzcnJwbTV4MUt1VHM
xdDNrb1huQndlUmdrNCtSU2U5bFhsRmNBNUdhS0p5ejNLSjQrM3Z4ZFwvVDZxQ25kak9QeXJKcCt6ZVNRbHgrdjE5
emhYdTJiY2NBWXhrK2xGRkZMSk9qaytNWEp0MXdlZ2xlZHdTTTlcL3NDQ09QYXQxaTA1R3N3Y1VsYW5ubkJ0VXRRe
Hg2QVVVVUM1XC9SU2VzNllOeGVpTXU4TGFDU1FSNmR4eDg1cDNaclJIczBUb1I5eXNuY3RhdTZqUlJRWWRRNmI4OG
VaYzhWME9rQ01kUGRuUDVpbVZ4dHpGeWhLaXlRU2hYM0hkSjlSUlJUNEowYUlVVUpZY3V6Nm9xVlpETzNnZkhPTTl
cL3RWUERpdFFvcmNkaE8xdHNZQW9vb0YxOTBcL0d2YUVGeFNtbmtjSmNUeng2RWZjVmhpYVBTbWEzSnVNOTZlcHZH
MUt4Z2NkZ2NrNUh0UlJTQ2xvb29vUFwvMlE9PSJ9fSwiaWF0IjoxNTcxMDg1Mjk3LCJleHAiOjE1NzEwODU0MTd9.
Ay6YOq1A6ECFehbwPidtlO-aHm2l5A3_nhK6ogM8j8rYglvNwXALTs5G8shEprAUUHDEcLelBBxN08gbyPcfQQ

Annex E
(informative)

Privacy and Security Recommendations

E.1 Introduction

This annex is informative guidance for issuing authorities to design and implement their mDL solutions for privacy and security. It also contains background information on privacy, and a section on how the existing standard for data transfer has already addressed privacy concerns.

Regulations such as Europe's GDPR regionalize privacy protections and set consequences for violating them. Every participant in the ecosystem should therefore use these recommendations and implement Privacy By Design for their components.

E.2 Achieving Privacy for the mDL Holder

E.2.1 Privacy goals

Individual privacy and security of personally identifiable information (PII) in the mobile, electronic age must be ensured and is a shared responsibility of all involved parties. No technical standard for data interchange can dictate how all privacy measures are achieved. Privacy is achieved by the end-to-end solution, and with the participation of all participants in an ecosystem. Each actor in the mDL ecosystem should fulfill their role in a responsible manner that best protects PII.

E.2.2 Privacy Definitions, Goals, and Principles

This document has been developed with the following Privacy By Design goals and allows both mDL Holder and issuing authority concerns listed in this Annex to be achieved in multiple ways.

- Data minimization and anonymization wherever possible
- Be proactive to prevent data breach
- Privacy should be the default setting
- Embed privacy in your design, flows, and architecture
- Privacy does not need to be traded off for full functionality
- Protect the full lifecycle of the identity end-to-end
- Keep all operations visible and transparent to the mDL Holder
- Design for user-centricity and user-control of their identity

To achieve these Privacy By Design goals, ISO/IEC 29100:2011 defines the following principles for privacy protection.

1. Consent and choice
2. Purpose legitimacy and specification
3. Collection limitation
4. Data minimization

5. Use, retention and disclosure limitation
6. Accuracy and quality
7. Openness, transparency and notice
8. Individual participation and access
9. Accountability
10. Information security
11. Privacy compliance

These principles put names on the protections that consumers (Data Subject) expect from Data Controllers and their Data Processors. These have been codified into law (e.g. GDPR) or regional principles (e.g. US DHS Fair Information Practice Principles).

- **Consent and Choice** – The Data Subject must consent to the processing of their personal data (see definitions in the section on mDL Holder Consent below).
- **Purpose Specification** – the Data Subject should be fully aware of the purpose for which their personal data is being collected, processed, and potentially stored.
- **Collection Limitation** – The Data Controller and Data Processors should only collect the data necessary for their purpose and should only collect data consistent with these principles.
- **Data Minimization** – Processing of data should be minimized to that specifically necessary for the purpose specified.
- **Use, Retention, and Disclosure Limitation** – Data Processors should not use personal data of the Data Subject except for the purposes specified and consistent with these other principles. Personal Data should only be retained for the period necessary to provide the service.
- **Data Accuracy and Quality** – High accuracy of data being processed and held is in the best interest of the Data Subject and processors should take measures to ensure accuracy.
- **Openness and Transparency** – What data and how data is being processed should be well-known to the Data Subject, including obtaining consent, and posting and updating clear notices.
- **Individual Participation** – the Data Subject should be involved in the collection, consent, processing, and storage management of their personal data.
- **Information Security (of Data and Data Subject)** - Personal data should be protected by security safeguards against such risks as loss or unauthorized access, destruction, use, modification or disclosure
- **Privacy Compliance, Accountability and Auditing** – The Data Controller and Data Processors must be accountable for all aspects of the processing of Personal Data and provide audit logs and auditability to the Data Subject.

E.2.3 Privacy Protections Embedded in the Protocol Design

The privacy principles above were taken into consideration during protocol design, and a set of protections implemented at the protocol level in order to establish a privacy baseline. Table 27 details protections embedded in the protocol. The remainder of this annex lists protections that should be considered and implemented as part of solutions.

Table 27 — – Privacy principles and protections

Principle	Protection
4. Data Minimization	Fields (data groups) have been separated into individual data elements to support privacy-preserving attribute queries. mDL Readers request data elements individually and should ask for only those necessary for their use case.
3. Collection Limitation 4. Data Minimization	The Request-response model permits disclosing additional data if the initial response did not fulfill the requirements of the use case. This lessens the need to over-ask for all data in the first pass.
1. Consent and Choice	The data transfer model of this document provides the ability for mDL solutions to implement either pre-consent or transaction-time consent as defined below.
4. Data minimization	Optional mDL Reader authentication may be used to restrict access to specific data elements of high sensitivity. This can prevent leakage, for example, of biometric data.
3. Collection Limitation	Replay attacks and follow-up requests are not permitted once the connection is broken in either online or offline models. Use of "intent to retain" provides a framework for limitation of collection of data by an mDL Reader.
10. Information security	mDL data is signed by the issuing authority using digital certificates for protection of integrity and authenticity. mDL Readers use trusted certificates to validate the integrity and authenticity of mDL data.
8. Individual participation	Sequence of "Device Engagement, secure connection, Request, Response, Repeat" permits pre-consent and transaction-time consent models at mDL Holder discretion.
10. Security of Data Subject	Lower level transport protocols should not leak unique mDL Holder or device identifiers. For Wi-Fi aware and BLE: this document encourages random MAC addresses for mDL on every data path and encourages rotation of MAC address for service discovery. For NFC: no unique stable identifiers should be available at the protocol level.
10. Information security of Data and Data Subject	Ephemeral keys are used in channel security. mDL Data Authentication can prevent data tampering. Random numbers are used in Issuer Data Authentication signatures.
4. Data minimization	Privacy-preserving signatures: Cryptographic signatures do not leak additional unique identifiers. That is, the protocol does not make mDL Holders identifiable if they cannot be identified by the transmitted data attributes themselves.
10. Security of Data Subject (Unlinkability)	Multiple mDL transaction signatures and the MSO of a single mDL Holder are distinct and not linkable. Other than repeated release of persistent mDL data elements, linkability could be created through the mDL authentication keys. Frequent update of mDL authentication keys reduces the linkability (see recomendation below).
6. Data Accuracy and Quality	The interface between the issuing authority and the mDL allows the issuing authority to update the mDL to contain up-to-date mDL data. Online retrieval allows the issuing authority to return up-to-date mDL data.

E.2.4 Responsibilities of Roles and Participants

E.2.4.1 Participants in the mDL Ecosystem

Each responsible participant in the mDL Ecosystem should make technology choices in line with privacy principles and the recommendations in this Annex. All participants should post their privacy policy

in easily discoverable public locations and provide mechanisms for contact and redress of privacy problems encountered by individuals.

- Issuing authorities - entities that officially issue Driving Licences... and now sign mDLs
- Technology Providers - provide systems and Apps for Issuing Authorities to issue mDLs
- mDL Verifiers - operate mDL Readers that obtain trustworthy data from mDLs
- Master List Providers - publish lists of trustworthy IACA certificates for verifying data
- mDL Holders - the individual who controls the mDL containing their identity data

E.2.4.2 Issuing Authorities

Above all, issuing authorities are responsible for selecting or developing technology that meets the privacy regulations of their jurisdictions and regions, and they have a responsibility to protect PII with the rigor with which they currently protect physical card issuances.

Issuing authorities are responsible for the accurate provisioning of mDL to the proper mDL Holder to achieve their target Identity Assurance Level as defined within their regional trust frameworks (e.g. NIST 800-63 IAL-3). Issuing Authorities are also responsible for security controls of the mDL in order to resist cloning and tampering including those binding the mDL to the mobile device, and to provide for the synchronization of relevant data updates to the mDL.

E.2.4.3 Technology Providers

A technology provider should apply Privacy By Design principles for the applicable role under which they will provide software - issuing authority, mDL Verifier, or Individual/Apps.

Technology Providers should work with issuing authorities to ensure that mDLs provide adequate information about consent, including purpose, and clear indication of what data is needed for how long.

Similarly, providers of mDL Verifier technology should work with mDL Verifiers to ensure only the data required for each use case is requested during each transaction.

E.2.4.4 mDL Verifiers (also known as Relying Parties)

Since the interchange of data with a Mobile Driving Licence is straightforward and easy to perform, it is recommended that mDL Verifiers never store PII, but rather request data from the mDL Holder whenever it needs to process data to perform a transaction.

mDL Verifiers should request the minimum data required for their use case. With each request for data, mDL Verifiers should provide adequate informed consent, including the purpose and scope of data requested, to the mDL Holder and never use blanket terms and conditions in lieu of informed consent.

mDL Verifiers must follow the applicable laws for data processing.

E.2.4.5 Master List Providers

Master Lists are signed lists of certificates from actual, real-world verified Issuing Authorities. These Lists can be maintained by issuing authorities themselves, associations of issuing authorities, by vendors of mDL Readers, or by others. The Master List Provider endeavors to keep the list of certificates current and extensive for their field of operation.

mDL Verifiers should use mechanisms to assemble IACA certificates from known trusted Issuing Authorities. They can do it using solutions such as bilateral and/or regional agreements, in effect acting as their own Master List Provider or they may obtain Master Lists of certificates from reliable Master List Providers.

E.2.4.6 mDL Holders

Individuals should pay attention to each consent they may be asked to grant, and not to accept blanket terms and conditions or consent to share. Consent should not be granted by the mDL Holder unless they are satisfied with the purpose for sharing specific data and know the mDL Verifier with whom they are sharing.

E.3 Privacy Recommendations

The remainder of this Annex contains guidance to assist Issuing Authorities and all participants in the mDL Ecosystem so that they can release solutions that protect the privacy of their citizens and achieve the above privacy principles.

E.4 Transparency of Data Storage & Use

mDL Holders should have transparency into all data that is held in the mDL. The mDL Holder should always be given the ability to consent to the sharing of that data and be informed of the onward storage of that data using the "intent to retain" flag. Proper informed consent as per section below can help ensure transparency over the use of personal data to the mDL Holder.

E.5 Protection of Keys and mDL Data

Offline retrieval requires that mDLs be able to dynamically authenticate (sign or MAC) the data they return to an mDL Reader in possession of the mDL Verifier. This mDL Authentication key should be protected from unauthorized usage.

The key should be protected with industry state-of-the-art protection methods. Examples are:

- A Trusted Execution Environment.
- A hardware Secure Element with appropriate certifications.
- Other secure hardware solution or combination of the aforementioned.
- Use of biometrics or strong on-device authentication technology

Controls and technology for fraud detection and secure mDL Reader configuration should be implemented.

In some other cases a backend could verify additional information such as digital signature, revocation and tokenization status. Such approach overall results in protecting the mDL Holder by ensuring that the right mDL Holder is presenting the data while not relying entirely on the mDL Holder device and mDL App security.

Many mDL data elements are PII and should also be stored securely. Encryption at rest is recommended, and if available secure hardware should be used to perform the encryption, with keys not available outside of the secure hardware, or else the data elements should be stored in the secure hardware.

Implementations may further protect the mDL authentication key and data with a biometric or PIN.

E.6 PKI and Trust Framework

The integrity and authenticity of an mDL is protected through the use of cryptographic mechanisms and digital certificates managed by a Public Key Infrastructure (PKI) under control of the respective issuing authority (IA). At the core of this PKI is the Issuing Authority Certification Authority (IACA), from which a specific set of certificates are generated and used to protect mDL data and transactions. The IACA (one per IA) is the root of trust for all mDL Verifiers (or relying parties in general) to securely validate the integrity and authenticity of mDLs issued by the respective IA. It is therefore of utmost

importance that IAs put strict security controls on the deployment, administration and operation of their IACA. A compromise of the IACA (e.g. disclosure of the IACA private key, fraudulent issuance of document signer certificates, etc.) can be exploited for forgery of mDLs and ultimately undermines the trust on the IA's mDLs.

Dissemination of IACA certificates is also critical to a robust and secure ecosystem. Trusting a fake IACA opens room for forgery of mDLs, leading to possible attacks of impersonation, skimming, eavesdropping, amongst others. mDL Verifiers should accept and trust an IACA root certificate through their own inspection or through a trusted Master List Provider.

At domestic level, dissemination of the IACA root certificate is expected to be distributed by the IA through secured and well known channels to the mDL Verifiers (e.g. downloadable from a webpage on the TLS protected website of the IA).

However, for cross border and interoperable verification of mDLs, a more scalable and dynamic mechanism is necessary. In the absence of a worldwide organization that could operate a globally trusted directory of IACA root certificates, it is envisioned that some organizations (public or private) will collect the list of IACAs and make available that compilation to mDL Verifiers, under the terms and conditions at their own discretion. Such organizations are referred to in this standard as Master List Providers and the list of IACAs as Master List (ML).

Given the criticality of the information contained in the Master List, a set of minimum security requirements are specified under Annex D. The requirements include provisions for the initial validation and due diligence procedures of IAs. The requirements also contain provisions for ML Providers to secure communication with IAs for the respective IACAs, data management and security controls over the ML and distribution of the ML to mDL Verifiers.

mDL Verifiers are recommended to integrate one or more MLs into their validation processes for security and interoperability, thus accepting mDLs for the broadest range of IAs. When selecting ML Providers, mDL Verifiers should assess the trustworthiness of the ML Provider amongst other criteria.

mDL Verifiers should load the trusted certificates from the selected Master List into the certificate store of each mDL Reader and ensure adequate protections of these certificate stores. mDL Readers engaged in online model transactions should confirm the TLS certificates of the IA against those trusted within their certificate store. All mDL Readers should validate the signatures on data elements against those trusted within their certificate store.

E.7 Auditability

E.7.1 Holder auditability

The mDL Holder should be able to view an audit trail of their interactions with mDL Readers. To accomplish this, mDLs should, at the mDL Holder's choice, log information about each transaction in a form that can be reviewed by the mDL Holder. The mDL should not make the transaction log available to anyone other than the mDL Holder, except at the mDL Holder's direction. The transaction log should include:

- The identity of the mDL Verifier / mDL Reader;
- Which data elements were requested by the mDL Reader;
- Which data element requests were rejected by the mDL, due to mDL Holder consent or other policy controls; and
- The date, time and location of the transaction.

To ensure that the audit log is not detrimental to mDL Holder privacy, the mDL Holder should have the capability to disable the audit log and delete logged transactions and mDL Holder Authentication should be required to view it. The transaction log should be stored in secure mobile storage.

E.7.2 mDL Verifier auditability

mDL Verifiers should provide the mDL Holder the ability to review data collected from their mDL transactions and to revoke consent to process or hold the mDL Holder's data.

The mDL Verifier should implement controls, which may include:

- Register mDL Readers individually and specify the names of the mDL Verifier and each mDL Reader operator if a common certificate used;
- Utilize Open ID Connect Client Registration in advance for connecting online mDL Readers to common issuing authorities, and Dynamic registration with the same mDL Reader information for less seldomly encountered IAs.
- Avoid logging mDL Holder PII, and log only ephemeral user identifiers if the identity of the mDL Holder is required by law to be logged
- Log only the data required by law

E.8 Anonymity and Unlinkability

E.8.1 Applicability

Anonymity and Unlinkability privacy recommendations apply equally to online and offline data transmission and regardless of the architecture of an mDL solution.

Within the scope of this Annex, we define anonymity according to ISO/IEC 29100 as a “characteristic of information that does not permit a personally identifiable information principal to be identified directly or indirectly“. We note that this explicitly includes required mitigations against de-anonymization techniques, which are defined as “any process in which anonymous data is cross-referenced with other sources of data to re-identify the anonymous data source” in ISO/IEC 20889.

Transactions between mDL Holders and mDL Verifiers should be anonymous and unlinkable under at least the following threat models:

- Tracking of identities by (potentially colluding) mDL Verifiers: While mDL Verifiers may be able to identify an mDL Holder based on transmitted attributes in one transaction, a different transaction with the same mDL Holder with a non-overlapping set of attributes should not be linkable to the previous transaction(s). Transactions without identifying attributes should remain anonymous to any Verifier.
- Linking pseudonyms or transactions across non-connected mDL Verifiers: Interactions of an mDL Holder with different (domains of) mDL Verifiers should not be linkable through cross-referencing the respective transaction logs.
- Tracking of identities by (global) passive adversaries: Passive observation (eavesdropping) of local or remote network communication should not lead to de-anonymization or linking of transactions between mDL Holders and Verifiers or the relationship between mDL Holders and issuing authorities.

De-identification alone is not sufficient for anonymization, and unlinkability requires anonymization of metadata elements. Note that revocation techniques may compromise privacy guarantees retroactively. Systems that implement explicit revocation techniques beyond limiting document lifetimes should therefore provide backward unlinkability in addition to unlinkability.

The following subsections provide an overview of technical and organizational mitigations to help provide anonymity and unlinkability.

E.8.2 Data Minimization for Metadata

It is important to note that anonymity or pseudonymity (e.g through de-identification) on data attributes can be ineffective if metadata items are associated with multiple transactions. Common metadata elements that could lead to privacy compromise are hardware or network addresses, long-term public key material, or session tokens. Additionally, statistical traffic analysis can lead to re-identification attacks and therefore compromise anonymity through linking of transactions (even if the content of these transactions is confidential through end-to-end encryption in upper layers).

Therefore, data minimization techniques should be applied to both data and metadata. The following subsections describe approaches to minimizing exposure of long-term key material. Minimization of address elements depends on the respective transport (both in device engagement and data transfer phases):

- NFC: Should not use static lower layer addresses.
- QR code: Does not use lower layer addresses.
- Bluetooth LE: LE Privacy should be enabled including private reconnection addresses as of Bluetooth 4.2 to randomize lower layer addresses.
- Wi-Fi Aware: Wi-Fi Aware device's NAN Management Interface Address (NMI) used for discovery and NAN Data Interface Address(es) (NDIs) should be randomly generated every time a Wi-Fi Aware interface is enabled. In addition, Wi-Fi Aware interface unique identifiers such as NMI MAC addresses, UUIDs, and TCP port numbers should be changed at regular intervals, e.g. every 15 minutes, if possible, as recommended by "Wi-Fi Alliance: Neighbor Awareness Networking Specification v3.0".
- Internet: IP addresses are generally visible to passive observers and may include long-term stable lower layer identifiers (such as MAC addresses in IPv6). For data minimization reasons, clients should enable address randomization where possible (including IPv6 address privacy). Explicit anonymization techniques such as tunneling through Tor are an option, but will reduce performance and reliability.

E.8.3 Unlinkability and Forward Secrecy through Ephemeral Identifiers

Ephemeral keys from mDL and mDL Reader are used to establish an authenticated, encrypted secure channel without exposing any long-lived public keys to passive observers. Active man-in-the-middle attackers could see mDL and mDL Reader authentication public keys. All ephemeral keys should be destroyed by both mDL and mDL Reader after use to ensure forward secrecy against an eavesdropper who records the encrypted session and later compromises one or both.

E.8.4 Rotation of Public Keys

The possibility of linkability through the mDL Authentication keys should be taken into consideration by IAs when deciding policy for the frequency of mDL updates. Rotation of public keys during mDL updates will implement some resistance to tracking of mDL transactions by mDL Verifiers. mDL updates can happen more frequently than mDL data is updated.

It is recommended that the mDL Authentication keys are changed as often as possible so they do not become a unique identifier that can be used to link mDL transactions even when the exchanged data does not uniquely identify the mDL Holder. The IA should implement solutions where the mDL authentication key is changed frequently. Triggers for authentication key rotation could be changes in the mDL data, number of days the key is active, or number of times the key is used.

mDLs may store a set of mDL authentication keys (together with a set of random numbers for all data elements and an MSO for each key) and choose an unused or random key for each transaction, or rotate keys for each transaction. In the event that all available mDL authentication keys have been used, the mDL may re-use a key, but re-use should be minimized and obtaining mDL Holder consent should be considered.

E.8.5 Token Time to Live

When exchanging an online token from mDL to mDL Reader during the device engagement or data transfer phases, the online token should be very short-lived and used only once. These two qualities will ensure against replay attacks and provide better mDL Holder control over mDL Verifiers asking for mDL Holder consent. Online tokens can be generated on the mobile device itself, or by the issuing authority service where it does not require a secure element on mobile. This will ensure scalability by supporting all kind of mobile devices without requiring particular hardware (e.g. SE).

Online tokens should not include access rights to mDL Holder data - called an Access Token. If an Access Token is used in a particular implementation of “pre-consent”, then that Access Token should not be exchanged during the device engagement phase, but rather during secure data transfer phase. Access Tokens should be time-limited and use-limited to prevent against replay attack.

E.9 Tracking mDL Usage

Issuing authorities and Technology Providers should not track mDL Holders or the usage of any mDL. Auditability should be for each individual mDL Holder and not permit issuing authorities or Technology Providers to track transactions, aggregate audit trails across mDL Holders, manage mDL Verifier to mDL Holder relationships, prevent (censor) mDL Holder to mDL Verifier relationships, or otherwise facilitate usage of identity information in an mDL that is not strictly in the interest of the mDL Holder.

Offline Transmission Mechanism. Transaction information must not be logged or uploaded in any form that permits issuing authorities or Technology Providers to track usage of the mDL or the habits of mDL Holders.

Online Token & Request Mechanism. The online call from an mDL Verifier to exchange a token for identity information about the mDL Holder should not be utilized as an entry point to track usage or mDL Holder habits.

Technology Providers should implement ephemeral session keys, OpenID Connect pairwise identifiers and key rotation described in the “Unlinkability” section so that mDL Verifiers cannot collude to track an mDL Holder’s usage of their mDL across use cases.

E.10 Collection Limitation

mDL Verifiers should not request all data elements, but should stick to requesting solely those data elements necessary for the transaction at hand.

The mDL should not respond with more data than was requested by the mDL Verifier in order to fulfill the transaction. This document permits field-by-field retrieval of data elements. Consent should always be granted by the mDL Holder with full knowledge of the requester, the requested data and purpose of granting consent.

E.11 Data Minimization

E.11.1 Issuing Authority Data Minimization

Considering the use cases from Annex B of alcohol purchase and car rental, issuing authorities should provision age verification statements to support mDL Verifiers across the largest geographical scope that their mDL Holders are expected to present their mDL. By provisioning nine (9) age verification statements of age greater than 10, each of 15 through 21, and 25, mDL Verifiers should be able to confirm age around the globe without revealing Date of Birth.

E.11.2 mDL Verifier Data Minimization

mDL Verifiers should not request all data elements, but should stick to requesting solely those data elements necessary for the transaction at hand. A wine store, for instance, requires that the purchaser

have government-issued ID that proves they are over-21 and must compare the person to the portrait in order to verify identity. The mDL Reader in this store can ask for age confirmation and portrait, and because the validation of signature on the data demonstrates an government-issued ID card, does not need to request additional data. They should not request the date of birth in order to calculate the over-21 value they require.

In no cases should mDL Verifiers retain or log information about the transaction without mDL Holder consent unless required for compliance with local legal regulations. For instance, the wine store should log what is necessary to display compliance with regulations that require an employee to verify the purchaser is over 21 and their identity matches their photos, but should never log names, identifying numbers, and portraits of patrons.

E.12 mDL Holder (User) Authentication

Unlocking and use of the mDL should only be permitted with appropriate mDL Holder authentication proving that the user is the intended mDL Holder. This ensures that others cannot collect mDL data without the mDL Holder.

User authentication may be required to unlock the device and the mDL. With this, it becomes more difficult for someone other than the mDL Holder to access and unlock the mDL data or to present and use the mDL. However, many mobile device users enroll device-level authentication for shared users of the device that may not be the mDL Holder (e.g. enrolling a child fingerprint to unlock a device in order to play games). mDL Holder authentication should tie the present user to the mDL Holder to whom the mDL was originally issued.

This document provides a method for identity verification performed by the mDL Verifier by comparing the portrait received by the mDL Reader to the person presenting the mDL. If this is not sufficient to meet the risk requirements of the transaction, the mDL Reader may implement biometric comparison of the person presenting the mDL to the portrait.

Access to data elements and the authentication key should be guarded by mDL Holder authentication. It is recommended that the authentication is implemented in secure hardware as well.

E.13 mDL Holder Consent

E.13.1 Informed Consent

No mDL data should be shared with any other party without informed consent. Informed Consent dictates that the mDL Holder be given sufficient informed just-in-time notice about the data being requested, the entity requesting the data, and the purpose for the request. Informed Consent also requires consent gathering — that the mDL Holder should be able to actively confirm or deny, given the informational notice, that their data is about to be shared with the mDL Verifier. Consent Gathering can happen at two distinct times in the flow of a transaction.

In no case should a blanket authorization to share data with any mDL Verifier or specific mDL Verifiers be obtained from the mDL Holder as part of Terms & Conditions, via a well explained pre-consent mechanism, or without a transaction-time informed consent process.

E.13.2 Operating System Medical IDs

On mobile operating systems, there is functionality for the user to provide a “Medical ID”, including name and emergency contact. The user can opt in to provide access to the “Medical ID” by emergency responders from an unlocked phone. Issuing authorities should not provide access to mDL Data without mDL Holder consent by any registered or unregistered terminal but can instead recommend users utilize the “Medical ID” feature.

E.13.3 Device Engagement

The action of tapping or allowing a QR code to be scanned is proximal consent to connect the mDL to the mDL Reader (offline retrieval) or for the mDL Reader to connect to the issuing authority (online retrieval).

E.13.4 Data Transfer

The mDL Reader requests data and the mDL should obtain consent from the mDL Holder to release the requested data elements and to validate the Issuing Authority signature on those data elements. This should be performed in one of two modes - Pre-consent or Transaction-time.

E.13.5 Pre-Consent

Pre-consent enables mDL Holders to configure specific mDL Verifiers with whom they have a trust relationship to share data. Those mDL Verifiers may be permitted access to a repeat set of mDL data without Transaction-time Consent. In all cases that pre-consent is long-lived, the mDL Holder should clearly select this configuration and be aware that pre-consent is in place.

E.13.6 Transaction-time Consent

Transaction-time consent is just-in-time informed notice and consent gathering during the processing of the request and before the response is provided to the mDL Reader.

E.14 Provisioning Accuracy & Reliability

E.14.1 mDL Provisioning

Achieving the privacy principle of Accuracy, Data Quality, and Integrity of the mDL across the mDL Ecosystem depends on accurate provisioning and reliability of mDL use. Nobody wants their identity documents in the hands of the wrong person, or to have someone else able to use their documents. Individual privacy and the trust of the ecosystem depend on accuracy.

In order to form trust an mDL should be provisioned accurately to the proper application on the mobile device of the intended mDL Holder, and it should remain under the control of that intended mDL Holder. Issuing authorities should select methods of provisioning an mDL to the mDL Holder's mobile devices through accurate and secure methods that resist spoofing and impersonation.

Note: Interfaces, protocols and services for provisioning the mDL are not within the scope of this document. These topics are addressed in the ongoing work on ISO/IEC 23220-3.

The Issuing Authority should ensure that the mDL be issued:

- According to Identity Assurance requirements of their jurisdiction (e.g. according to NIST 800-63A for Issuing Authorities in the United States).
- To the intended secure application that meets privacy and PII protection requirements
- To the proper and intended Holder of the driving licence document

and that the mDL:

- Remains in control of that mDL Holder throughout the lifetime of the credential
- Protects the credential and the mDL data
- Can only be used by the mDL Holder or with the consent of the mDL Holder
- Meets the mDL Verifier's needs for Authentication Assurance particular to their region

E.14.2 Provisioning to the Right Application

In the issuance process, there should be some means for the Issuing Authority to ensure the application can be identified as genuine and valid. Various proprietary techniques could apply to identify an application as a genuine. Various code obfuscation and tamper detection techniques are available off-the-shelf to facilitate hacking detection.

E.14.3 Provisioning to the Right mDL Holder

Many methods in use today for obtaining internet credentials – for example, possession of an email address, SMS of one-time passcode, and self-asserted enrollment – do not provide sufficient assurance on their own and will not result in trusted credentials or trust across the mDL ecosystem.

There are various techniques to ensure the intendedmDL Holder receives the mDL such as:

- In-Person Provisioning
- Remote Identity Proofing
- Trusted referees or monitored remote Proofing
- User authentication from the App and mDL provisioning once authenticated
- Using multiple alternative personal channels (mail, SMS, email) to communicate a provisioning "trigger" that is combined to achieve identity assurance

E.14.4 Continued Use by the Right User

The mDL should protect against unauthorized use of mDL by other than the mDL Holder. This should include using authentication factors that tie the user to the mDL data and be wary of device authentication factors that support device-sharing (e.g. registering your children's fingerprints for them to play games on your device). The mDL should not solely rely on device security.

The mDL should authenticate the mDL Holder when launching the mDL to prevent against leaking mDL data to unauthorized users. At transaction-time, identity verification will be performed by the mDL Verifier.

E.15 Data Accuracy and Freshness

Issuers sign the mDL data and therefore are declaring the accuracy of mDL data at the time of signing.

Issuers should establish policy for how often mDL data is to be refreshed. This policy is reflected in the mobile security object (MSO). mDL solutions should endeavor to refresh mDLs at least as often as policy states.

Verifiers should implement business decisions related to that freshness if mDL data is out of date.

E.16 Reader Registration and Authentication

Reader registration is an optional feature so that mDL solutions do not require registration of readers to one central Issuing Authority in order to allow mDL transactions. Consolidating registrations as such would create a tracking vector as discussed above. Implementing Reader registration however allows various use cases when knowledge of the mDL Reader is important, such as access to restricted data elements, providing informed consent to the user, and protecting against man-in-middle attacks that could compromise the confidentiality of mDL data.

It is worth noting that the standard ensures that access to mandatory mDL data elements as listed in Table 3 must not be rejected for unregistered mDL Readers.

If the terminal is registered to allow its authentication, the Issuing Authority should put in place policies and a registration process that guarantees the identity of the holder of the reader, and ensures the protection of the authentication secret and keys within the Reader.

When the Reader is authentified, concern should be taken to make sure the element of its authentication does not become unique identifier allowing to track the reader itself. The mechanisms for rotation of public key can be applied to the reader Authentication keys, as it is to the mDL keys, to achieve this.

While the holder of the mDL is likely to have a right to uniquely identify who initiated a data reading transaction from his mDL for later complaints, the terminal can be protected in order to avoid tracking from third party observing the exchange, or from the mDL if it can initiate the transaction without active involvement of the reader holder. A registry of the temporary Authentication keys issued to readers, managed by the Issuing Authority, can be the mechanism for this.

E.17 mDL Reader Connections to Issuing Authority Base URL

E.17.1 Rogue OP and Rogue Verifier Protection

In the online 'oidc' model, the Base URL of the Issuing Authority Open ID Provider (OP) is provided to Readers during Device Engagement through a QR code or NFC tap. Readers should be protected from connecting to rogue OPs and mDL Holder data should not be leaked to unqualified participants (rogue Verifiers).

Key to the protection of participants in mDL transactions are Reader Registration and the selection of accurate Trust Lists of Signer Certificates.

E.17.2 mDL Reader Protection

Verifiers should ensure they only trust Signer Certificates and Root certificates of well-known Issuing Authorities. They may use for this trusted IDL Master Lists provided by reputable providers that vet the content of their lists through an adequate policy, according to the requirements described in Annex D of the standard.

The TLS certificates used in the TLS connection from the Reader to the OP should be validated against the trusted Issuing Authority's IACA Certificate. The reader will verify that the TLS certificate is signed by either the root certificate of the Issuing Authorities it trusts or the Issuing Authority's CA for web services.

Bibliography

[1] ISO/IEC 7816-8:2016, *Identification cards -- Integrated circuit cards -- Part 8: Commands and mechanisms for security operations*

[2] ISO/IEC 16022:2006, *Information technology -- Automatic identification and data capture techniques -- Data Matrix bar code symbology specification*

[3] AAMVA Mobile Driver's License Functional Needs White Paper version 0.7

[4] OpenID Connect Back-Channel Logout *M. Jones et. al., Defines a logout mechanism that uses direct back-channel communication between the OP and RPs being logged out, January* 2017

[5] ISO/IEC 18000-3:2010, *Information technology -- Radio frequency identification for item management -- Part 3: Parameters for air interface communications at 13,56 MHz*

[6] ISO/IEC 18092:2013, *Information technology -- Telecommunications and information exchange between systems -- Near Field Communication -- Interface and Protocol (NFCIP-1)*

[7] ETSI EN 319-411-1, *Electronic Signatures and Infrastructures (ESI); Policy and security requirements for Trust Service Providers issuing certificates; Part 1: General requirements, Version 1.2.0, August 2017*

[8] FPKIPA *X.509 Certificate Policy For The U.S. Federal PKI Common Policy Framework, Version 1.24, May* 2015

[9] ETSI TR 103 370 V1.1.1*"Practical introductory guide to Technical Standards for Privacy, January 2019*

[10] Privacy by Design" The 7 Foundational Principles", by Ann Cavoukian, Ph.D., Information & Privacy Commissioner Ontario, Canada

[11] ISO/IEC 29100:2011, *Information technology — Security techniques — Privacy framework*

[12] RFC 4287, *M. Nottingham et al., The Atom Syndication Format, December 2005*

[13] RFC 8252, *W. Denniss et al., OAuth 2.0 for Native Apps, October 2019*

[14] ISO/IEC 8859-1:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1*